

## TD2: Localisation par filtre de Kalman étendu

Ce deuxième TD traite de la localisation d'un robot mobile. Le robot simulé est de type unicycle à conduite différentielle. Matlab sera employé pour les simulations. Un simulateur "*simu.m*" est fourni dans le dossier **packTD2.zip** disponible sur la page web du cours.

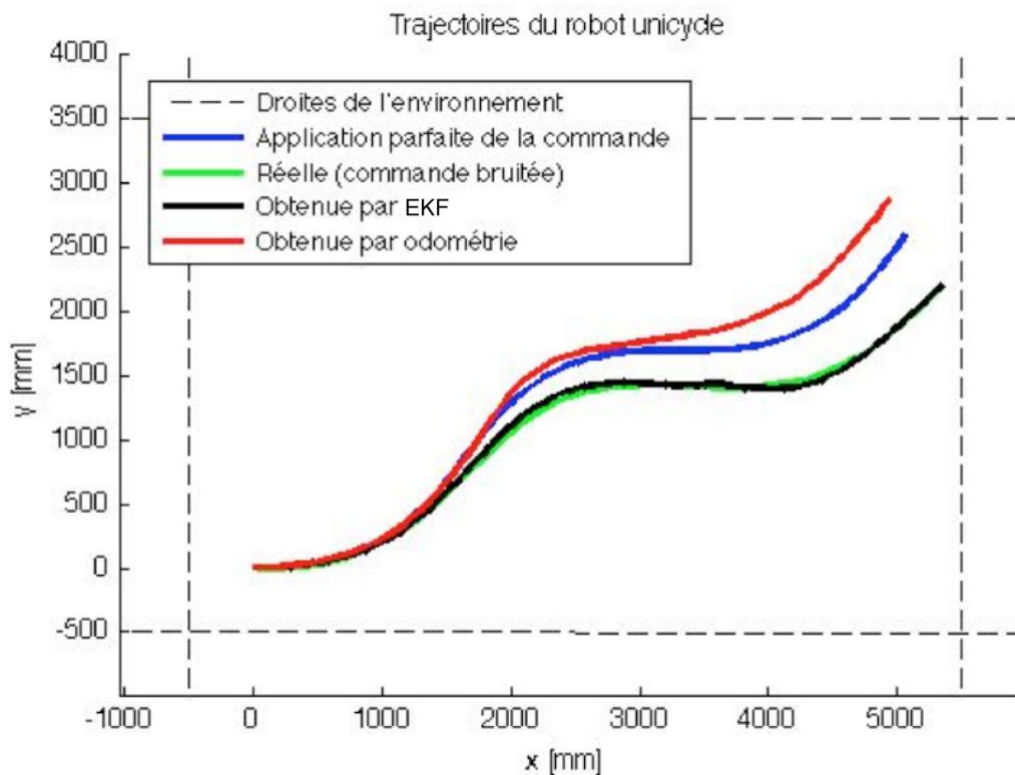


Fig. 1 - Simulation du mouvement d'un robot unicycle dans un environnement continu décrit par quatre droites (traits pointillés).

L'objectif du TD est d'utiliser le *filtre de Kalman étendu*<sup>1</sup> (EKF) pour localiser le robot mobile en mouvement dans un environnement délimité par quatre droites (pointillées dans la Fig. 1 ci-dessus).

La *prédiction* se fait par odométrie et la *correction* se fait grâce à la connaissance et à la perception des droites (les primitives de l'environnement).

Les droites, en représentation polaire ( $r, \alpha$ ), forment la carte M de l'environnement et sont accessibles dans le code Matlab dans les tableaux monodimensionnels *alpha\_droites* et *r\_droites* (voir les fichiers "*initEnviro.m*", "*genereTrajs.m*"). Leurs perceptions sont accessibles à chaque instant de la boucle de simulation dans *alpha\_droite\_obs* et *r\_droites\_obs*, tableaux à deux dimensions (index de temps et paramètre de droite perçue).

<sup>1</sup> On utilise le filtre de Kalman étendu car notre modèle dynamique (c'est-à-dire, le champ vectoriel  $f(\mathbf{x}_{t-1}, \mathbf{u}_t)$ ) est *non linéaire*. On se sert d'une *version linéarisée* de cette fonction (cf. les matrices jacobiennes  $\mathbf{F}_x$ ,  $\mathbf{F}_u$ ) dans le filtre de Kalman étendu.

### Exercice 1:

Créez une fonction appelée *prediction* qui prend en entrée l'état (c'est-à-dire, la pose du robot) estimé à l'instant précédent, la matrice d'incertitude de l'état à l'instant précédent, la distance parcourue par la roue droite et gauche, la longueur de l'essieu et les gains d'incertitude sur les incréments des roues. La fonction renvoie l'état et sa matrice d'incertitude prédits.

### Exercice 2:

Créez les quatre fonctions suivantes, correspondant à la phase de *correction* du filtre de Kalman étendu:

1. Une fonction *correction* qui prend en entrée l'état et l'incertitude prédits, entre autres, et renvoie l'état et l'incertitude estimés. Cette fonction réalise l'observation (lecture des tableaux *alpha\_droite\_obs* et *r\_droites\_obs*). Elle appelle les trois fonctions suivantes:
2. Une fonction *predictionDeMesure* qui prend en entrée le modèle de l'environnement et le transforme pour qu'il soit exprimé dans le repère  $\{R\}$  du robot mobile prédit.
3. Une fonction *appariement* qui met en correspondance les primitives observées et prédites par calcul d'innovation et de distance de Mahalanobis.
4. Une fonction *estimation* qui applique la dernière étape de la mise à jour de l'état par mesure.

Affichez la trajectoire (positions  $x$  et  $y$  estimées) et les ellipses d'incertitude sur la pose du robot, en réutilisant ce qui a été vu au TD 1 (cf. l'Exercice 3).