

## TD no3 : Filtre de Kalman Estimation de la position d'un robot

L'objectif de ce TD est d'utiliser un filtre de Kalman et de le mettre en œuvre pour estimer la position d'un robot à partir des données fournies par différents capteurs.

### I. Implémentation du filtre de Kalman

Le fichier Python *KalmanFilter.py* définit la classe d'objet *KalmanFilter* pour l'implémentation d'un filtre de Kalman linéaire.

I.1) A partir des formules vues en cours, complétez les méthodes *predict* et *update* implantant respectivement les phases de prédiction et de mise à jour du filtre.

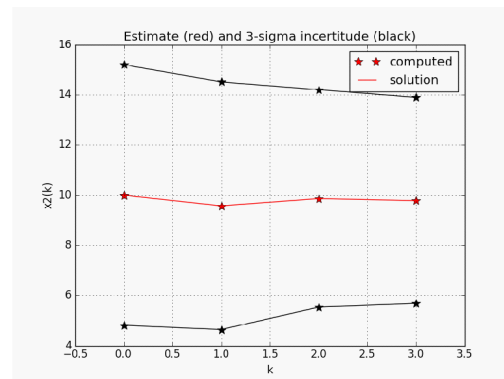
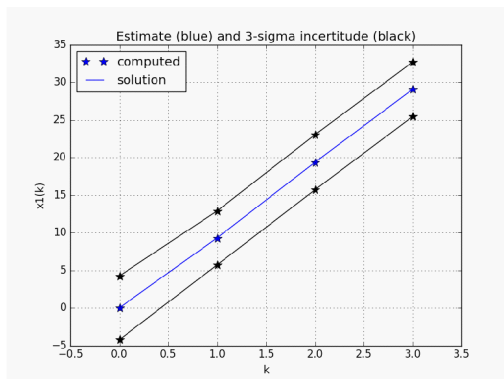
Fonctions Python utiles :

*numpy.dot(A, B)* : calcul du produit matriciel  $A*B$

*A.T* : calcul de la transposée de la matrice  $A$

*numpy.linalg.inv(A)* : calcul de l'inverse d'une matrice  $A$  inversible

I.2) Exécutez le fichier *KalmanFilter.py* pour valider votre implémentation à partir de l'exemple fourni dans la fonction *main*. (Pour vérification : les étoiles doivent être situées sur les courbes tracées si l'implémentation est correcte, comme présenté sur les figures ci-dessous)



### II. Estimation de la position du robot

On cherche à estimer au mieux la position (en 1 dimension) d'un robot à partir d'une mesure fournie par un capteur de vitesse et d'une mesure fournie par un capteur de position.

Le fichier *donnees.py* permet de définir et d'utiliser différents jeux de données :

- des mesures en position et vitesse enregistrées lorsque le robot est immobile à la position de coordonnée zéro (classe *MesuresAuRepos*)
- la position et la vitesse réelle du robot lorsqu'il est en mouvement selon une trajectoire prédéfinie (classe *TrajectoireVraie*). Elles serviront de base de comparaison pour comparer les estimations réalisées.
- les mesures en position et vitesse enregistrées lorsque le robot est en mouvement selon cette trajectoire prédéfinie (classe *MesuresTrajectoire*)

Toutes les données (temps, position, vitesse) sont exprimées en unité SI.

### II.1) Caractérisation des capteurs

A partir du fichier *etudeMesuresRobotImmobile.py*, utilisez le jeu de données correspondant au cas où le robot est immobile à la position de coordonnée zéro pour estimer les caractéristiques (moyenne, écart type et variance) des mesures fournies par le capteur en position et le capteur en vitesse.

Fonctions Python utiles : *numpy.std(x)* : calcul de l'écart type ; *numpy.mean(x)* : calcul de la moyenne

Remplissez le tableau suivant :

	Moyenne	Ecart type	Variance
Capteur de position			
Capteur de vitesse			

### II.2) Estimation de la position à partir des mesure de vitesse uniquement

II.2.1) Complétez dans fichier *EstimationPosition.py* les caractéristiques des capteurs (écarts types des mesures).

II.2.1) Ne connaissant pas la position initiale réelle du robot on supposera que celle-ci est nulle (variable *x0Est*). Complétez le code fourni pour calculer directement par intégration numérique la position du robot à partir de la mesure en vitesse. Que remarque-t-on ?

### II.3) Estimation de la position par fusion des mesures en position et en vitesse à l'aide d'un filtre de Kalman

II.3.1) De quelle(s) grandeur(s) est composé le vecteur d'état que l'on cherche à estimer grâce au filtre de Kalman ? Quelle sera la grandeur utilisée comme entrée du filtre de Kalman ? Quelle sera la grandeur utilisée comme mesure au sein du filtre de Kalman ? Quelle est la représentation d'état associée ? (équation d'état, équation de mesure)

II.3.1) Complétez dans fichier *EstimationPosition.py* la définition du filtre de Kalman (dimensions et matrices de la représentation d'état).

II.3.2) A quoi correspondent le bruit d'état et le bruit de mesure pris en compte par le filtre de Kalman dans le problème que l'on considère ici ? Complétez la définition des matrices de covariance de bruit d'état et de bruit de mesure dans le fichier *EstimationPosition.py*.

II.3.2) Lancez le script dans fichier *EstimationPosition.py* pour exécuter le filtre de Kalman et calculer l'estimée en position. Comparez cette position filtrée avec celle calculée à la question II.2.1.

II.3.3) Etude de l'influence des covariances sur le bruit de mesure et sur le bruit d'état :  
- Multipliez la covariance de bruit d'état par un facteur que vous ferez varier par puissance de 10 de  $10^1$  à  $10^{-4}$ . Que constatez-vous ? Quelle explication peut-on donner ?

- Multipliez la covariance de bruit de mesure par un facteur que vous ferez varier par puissance de 10 de  $10^{-1}$  à  $10^4$ . Que constatez-vous ? Quelle explication peut-on donner ?

II.3.4) On supposera qu'on peut faire confiance aux mesures fournies par le capteur de vitesse et on multipliera par  $10^{-2}$  la covariance sur le bruit d'état. On cherche à étudier l'influence de la covariance sur l'estimée initiale. Pour cela multipliez cette covariance par un facteur que vous ferez varier par puissance de 10 de  $10^0$  à  $10^{-3}$ . Que constatez-vous ? Quelle explication peut-on donner ?

II.3.5) Etude de l'influence de la fréquence des mesures. On supposera que le capteur en position ne délivre pas une mesure à chaque période d'échantillonnage, mais à des intervalles de temps dont la durée est un multiple de celle-ci (avec un facteur multiplicatif  $\alpha$ ). Modifiez la ligne de code présente en début du fichier *EstimationPosition.py*

`mesuresTrajectoire = donnees.MesuresTrajectoire(pasMesures=alpha)`

en testant les valeurs numériques suivantes :  $\alpha = 1, 2, 5$  ou  $10$ . Que constatez-vous ? Quelle explication peut-on donner ?