

ROB312 - TD2

BRAMBILLA Davide Luigi

5 Décembre 2019

1 Partie 1 - Filtre de Kalman

Le filtre de Kalman est une méthode utilisée pour estimer les états d'un système dynamique à partir d'une série de mesures incomplètes ou bruitées. Le filtre de Kalman est utilisé dans le cas d'un système linéaire.

Le code

```
clear % effacer toutes les variables
close all % fermer toutes les fenetres
clc % effacer la ligne de commande
rng(123456) % imposer la graine de gnration de nombres pseudo-alatoire pour la rptabilit

% Paramtres initiaux et de simulation
P_hat = diag([10, 10, 10, 3, 3, 1].^2); % matrice de covariance initiale
X_hat = [0, 0, 100, 5, 5, 0]'; % estim initial
d = size(X_hat,1); % dimension de l'tat
X_reel = X_hat + sqrtm(P_hat)*randn(d,1); % tat vrai (inconnu du filtre)
dt = 0.1; % pas de temps
F = [1 0 0 dt 0 0; 0 1 0 0 dt 0; 0 0 1 0 0 dt; 0 0 0 1 0 0; 0 0 0 0 1 0; 0 0 0 0 0 1]; % matrice
    de dynamique ( complter)
Q = diag([0.2,0.2,0.2,0.01,0.01,0.001]); % matrice de covariance de bruit de dynamique
H = [1 0 0 0 0 0; 0 1 0 0 0 0; 0 0 1 0 0 0]; % matrice d'observation ( complter)
R = diag([10,10,10]); % matrice de covariance du bruit de mesure
dm = size(R,1); % dimension du vecteur de mesures

% Boucle de simulation
T = 10; % dure (s)
for tk = 2:(T/dt)
    % simulation de l'tat vrai
    t = dt*(tk-1); % temps courant
    X_reel = F * X_reel + sqrtm(Q) * randn(d,1); % propagation de l'tat rel ( complter)

    % prediction ( complter: variables X_hat et P_hat)
    X_hat = F * X_hat;
    P_hat = F * P_hat * F' + Q;

    % gnration de la mesure relle bruite ( complter)
    Y = H * X_reel + sqrtm(R) * randn(dm,1);

    % validit de la mesure relle
    if t > 3 && t < 7
        is_measurementValid = false;
    else
        is_measurementValid = true;
    end

    % correction ( complter: variables K, P_hat, inno, X_hat)
    if is_measurementValid
        K = P_hat * H' * inv(R + H * P_hat * H');
        inno = (Y - H*X_hat);
        X_hat = X_hat + K * inno;
        P_hat = (eye(d) - K*H)*P_hat;
    end
end
```

Q1 - Les différents paramètres du filtre

Au début de notre algorithme il est possible de voir la définition des paramètres initiaux et lesquels qui nous allons utiliser et modifier pendant l'exécution du programme. On trouve:

- X_r qui représente les états effectifs du système qui ne sont pas connus et sur lesquels on devra faire l'estimation.
- \hat{X} qui indique l'estimation des états. Le but du filtre est, en fait, de faire rapprocher les valeurs des états observés aux valeurs des états réels du système.

- **Y** qui représente les états mesurables (de sortie) du système.
- **H** qui représente la matrice d'observation du système: laquelle qui nous permet d'obtenir les états mesurables Y du système à partir des états internes X_T .
- \hat{P} qui représente la matrice de covariance de l'erreur. Étant donnée que nous n'avons pas une connaissance sur l'état qu'on doit observer nous allons introduire une distribution gaussienne de probabilité où P représentera la racine carrée de l'écart type de la **gaussienne** considérée. Elle donne une mesure de la précision de l'état estimé.
- **F** qui indique la matrice de dynamique, la matrice qui permet de estimer l'état successif en fonction de l'état précédent.
- **Q** qui donne la matrice de covariance du bruit dynamique. Ce type de bruit influence les valeurs des états réels du système.
- **R** qui donne la matrice de covariance du bruit de mesure. Ce type de bruit influence directement les états mesurés du système.
- **K** qui donne les coefficients de gain k du filtre pour pouvoir passer de la valeur de l'état estimé à sa valeur successive.

Q2 - Le code et les résultats

Dans cette deuxième section, il nous est demandé de compléter le code afin de pouvoir réaliser le filtre de Kalman.

Nous avons complété les parties de **prédiction** et **correction**. La première partie utilise l'état estimé à l'instant précédent pour produire une estimation de l'état courant. La deuxième partie utilise les estimation de l'état courant pour corriger la prédiction afin d'obtenir une estimation plus précise et, au même temps, calculer les coefficients k .

Nous avons implémenté dans notre code ce deux phases.

La prédiction

Nous avons implémenté dans le code les équations suivantes:

$$\hat{X}_k = F * \hat{X}_{k-1} + B * u_k$$

où nous avons choisi $u_k = 0$.

$$\hat{P}_k = F * \hat{P}_{k-1} * F^T + Q_k$$

La correction

Nous avons implémenté dans le code les équations suivantes:

$$\hat{X}_k = \hat{X}_{k-1} + K * (Y_k - H_k * \hat{x}_{k-1})$$

$$\hat{P}_k = (I_d - K * H) * \hat{P}_k$$

$$K = \hat{P}_k * H_k^T * (R_k + H_k * \hat{P}_k * H_k^T)^{-1}$$

Notre cas

Dans notre cas, les états du système à observer sont 6 et sont les trois positions et les trois vitesses.

En executant le code nous avons obtenu:

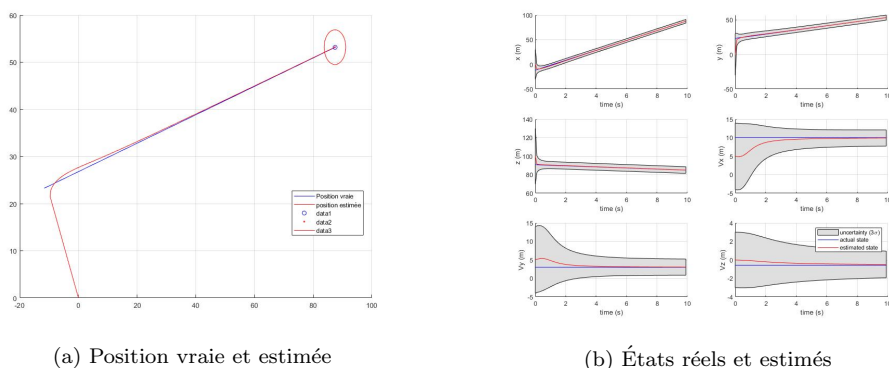


Figure 1: Les résultats obtenus du filtre de Kalman

Dans la figure de gauche nous pouvons observer comment les états estimés permettent de reconstituer la position du système (en rouge) par rapport à la vraie position du système (en bleu).

Dans la figure de droite nous avons le graphique des états estimés par rapport auxquels réels.

Il est possible de voir que, dans ce cas, étant donnée que le système est linéaire, avec le seul filtre de Kalman il est possible d'estimer et reconstituer les états. De plus, nous pouvons remarquer que l'estimation est précise et qu'elle converge rapidement vers l'état réel.

Q3 - Troue de mesures entre 3 seconds et 7 seconds

Dans cette section, il nous est demandé de rendre la partie de correction indisponible pour 4 seconds, entre le second 3 et le second 7.

Ce qu'on observe c'est une croissance d'incertitude entre les états estimés et lesquels réels.

J'ai reporté ci dessous les résultats obtenus dans ce cas.

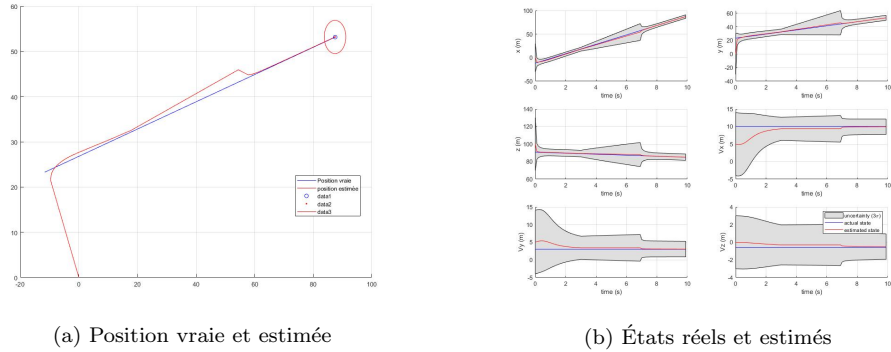


Figure 2: Les résultats obtenus du filtre de Kalman sans la phase de correction entre 3 et 7 seconds

Dans le graphique de gauche, il est possible de voir que le filtre, entre le second 3 et le second 7, maintient toujours ses coefficients constants et ne pourra pas *corriger* son estimation. Dans notre cas, étant donnée que le mouvement ne change pas trop ses caractéristiques l'estimation obtenue n'est pas trop mauvaise. Par contre, si on aurait eu un changement rapide au même temps de la perte d'information sur les mesures l'estimation ne serait pas adéquate.

Le graphique de droite montre significativement la perte d'information directe sur les états du système. Il est possible de voir, par exemple dans les trois positions x , y et z , que même si il n'y a pas une différence trop grande entre l'état estimé et l'état effectif, on peut voir comment l'*incertitude* augmente fortement. Elle diminue en suite quand on recommence à avoir l'information sur les mesures.

Q4 - Variation du bruit dynamique

Dans cette question, nous avons ajouté le bruit sur la dynamique du système. En particulier nous allons l'appliquer à l'état *réel* X_r :

$$X_r = F * X_r + \sqrt{Q} * randn(d, 1)$$

où le dernier terme est un vecteur de termes aléatoires de dimension égal à laquelle de l'état.

En introduisant le bruit dynamique, nous allons avoir une variation inconnue sur les états réels et donc aussi sur l'état de sortie Y . Cela signifie avoir une trajectoire avec plusieurs petits changements par rapport à la trajectoire espérée.

Si la valeur de bruit reste petit par rapport à le reste nous aurons que notre suivi ne sera pas trop mauvaise et nous aurons que le maximum d'erreur arrive à son maximum dans les moments où les mesures ne sont pas disponibles.

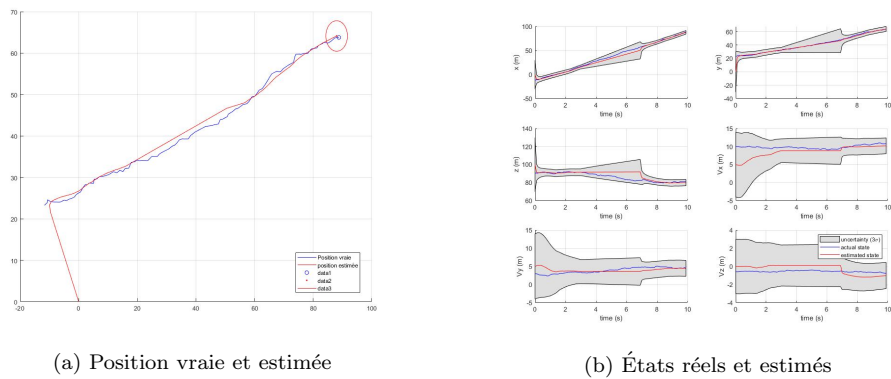
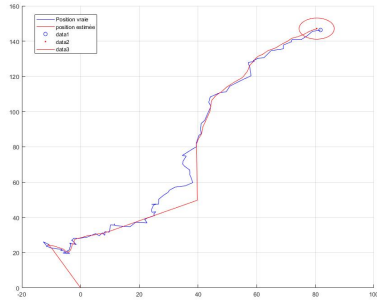
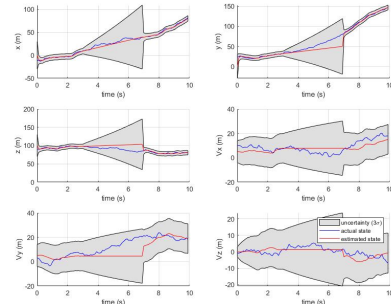


Figure 3: Les résultats obtenus du filtre de Kalman avec bruit dynamique

Si par contre, nous augmentons la valeur du bruit nous aurons que les performances de notre filtre se détériorent.



(a) Position vraie et estimée avec le trou de mesures

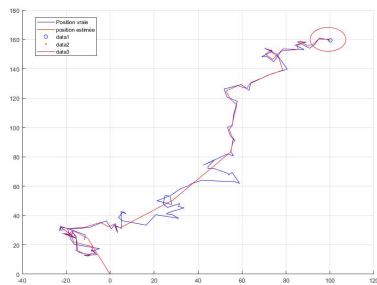


(b) États réels et estimés avec le trou de mesures

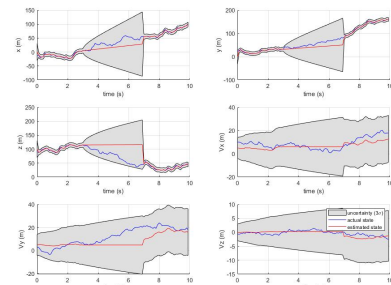
Figure 4: Les résultats obtenus du filtre de Kalman sans la phase de correction entre 3 et 7 seconds et avec un grand bruit dynamique

Il est possible de voir, dans la figure 4 que si nous n'avons pas la phase de *correction*, la différence par rapport à la valeur bruitée devient grande et la valeur d'incertitude sur tous les états est très significative. Si, par contre, la phase de correction est active nous retournons à avoir rapidement une bonne approximation des états et à réduire fortement l'incertitude.

Si nous allons fortement augmenter le bruit dynamique nous allons obtenir:



(a) Position vraie et estimée avec fort bruit dynamique



(b) États réels et estimés avec fort bruit dynamique

Figure 5: Les résultats obtenus du filtre de Kalman avec bruit dynamique multiplié fois 100

Même si la trajectoire est plus bruitée nous allons quand même pouvoir suivre la trajectoire donc ce type de bruit est partialement mitigé par l'effet du filtre.

De plus, si on a la phase de correction disponible, le filtre arrive à bien suivre la trajectoire.

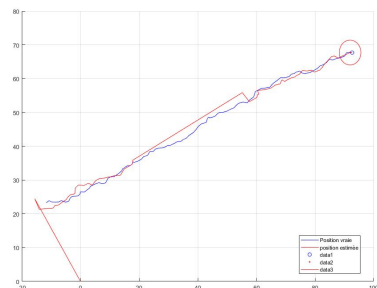
Q5 - Variation du bruit de mesure

Dans cette section, nous avons ajouté le bruit sur les mesures du système. En particulier nous allons l'appliquer à l'état *mesurée* Y :

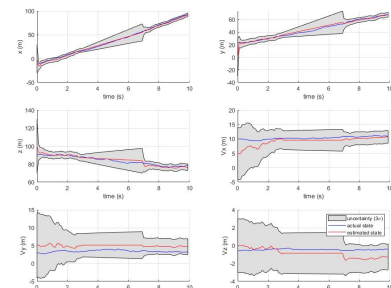
$$Y = H * X_r + \sqrt{R} * randn(dm, 1)$$

où le dernier terme est un vecteur de termes aléatoires de dimension égal à laquelle des mesures.

En introduisant ce bruit de mesure, nous n'aurons pas seulement un bruit sur les états interne du système mais aussi sur les états Y visibles et donc il aura une influence directe plus évidente comme l'on peut voir des résultats:



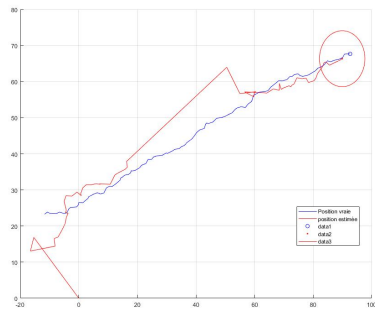
(a) Position vraie et estimée



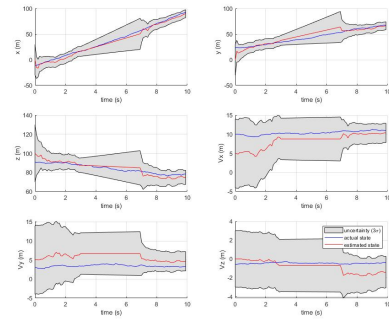
(b) États réels et estimés

Figure 6: Les résultats obtenus du filtre de Kalman avec bruit dynamique et de mesure

En fait, si dans ce cas nous augmentons la valeur du bruit, il est possible de voir que le filtre devient moins performante et que la grandeur du bruit affecte directement sur l'estimation des états de notre système.



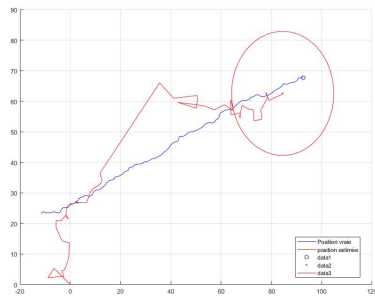
(a) Position vraie et estimée



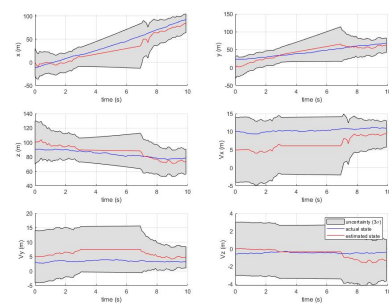
(b) États réels et estimés

Figure 7: Les résultats obtenus du filtre de Kalman sans la phase de correction entre 3 et 7 seconds et avec bruit dynamique et de mesure

Si, dans ce cas aussi, on multiplie la valeur du bruit pour un facteur 100 nous allons obtenir:



(a) Position vraie et estimée



(b) États réels et estimés

Figure 8: Les résultats obtenus du filtre de Kalman avec bruit dynamique et (fort) de mesure

Ce qu'on peut observer c'est que ce type de bruit est plus influente sur les performances du filtre par rapport au bruit dynamique.

On peut conclure que le filtre de Kalman permet de bien estimer les états du système même si il y a du bruit dynamique où du bruit sur les mesures en faisant plus confiance au modèle quand le bruit sur les mesures est grande et le filtre fait plus confiance aux mesures quand le modèle qu'on a du système n'est pas précis on donnera plus d'importance aux mesures.

2 Partie 2 - Filtre de Kalman Étendu

Dans le cas du filtre de Kalman nous voulons estimer les paramètres du système. Pour le faire, nous devons modéliser le système et, dans certains cas, les équations qui permettent de le faire ne sont pas linéairisables. Pour cela le filtre de Kalman n'est plus applicable.

Par contre, il est encore possible estimer les états du système en utilisant le filtre de Kalman étendu. Ce filtre permet de linéariser localement le problème et d'appliquer les équations du filtre de Kalman classique.

Le code

```
clear % effacer toutes les variables
close all % fermer toutes les fenetres
clc % effacer la ligne de commande
rng(123456) % imposer la graine de gnration de nombres pseudo-alatoire pour la rptabilit

% Paramtres du filtre
P_hat = diag([10, 10, 10*pi/180].^2); % matrice de covariance initiale
X_hat = [0, 0, 0]'; % estim initial
d = size(X_hat,1); % dimension de l'tat
Qf = diag([0.1,0.1,0.001]); % matrice de covariance de bruit de dynamique
Rfi = 100*diag([10,1*pi/180].^2); % matrice de covariance du bruit de mesure pour chaque amer

% Paramtres initiaux et de simulation
X_reel = X_hat + sqrt(P_hat)*randn(d,1); % tat vrai (inconnu du filtre)
dt = 0.1; % pas de temps
Q = diag([0.0001,0.0001,0.01]); % matrice de covariance de bruit de dynamique
Ri = diag([10,1*pi/180].^2); % matrice de covariance du bruit de mesure pour chaque amer
amers(1).P = [10;00]; % position de l'amer 1 (point de repre)
amers(2).P = [0;50]; % position de l'amer 2 (point de repre)
% amers(3).P = [0;75]; % position de l'amer 3 (point de repre)
% amers(4).P = [25;50]; % position de l'amer 4 (comportement terrible) (point de repre)
% amers(4).P = [0;125]; % position de l'amer 4 (point de repre)
dm = size(amers,2)*size(Ri,1); % dimension du vecteur de mesures
R = [];
Rf = [];
for i = 1:size(amers,2)
    R = blkdiag(R,Ri);
    Rf = blkdiag(Rf,Rfi);
end

% Rf = 100*Rf;

% Initialisation des variables de stockage des donnees
tk=1;
t_sim(tk) = 0;
K = zeros(d,dm);
K_sim(:, :, tk) = K;
inno = zeros(dm,1);
inno_sim(:, tk) = inno;
P_sim(:, :, tk) = P_hat;
Pdiag_sim(:, :, tk) = diag(P_hat);
X_reel_sim(:, tk) = X_reel;
X_hat_sim(:, tk) = X_hat;
% Boucle de simulation
T = 10; % dure (s)
for tk = 2:(T/dt)
    % commande (dfinit la trajectoire)
    V = 10; % Vitesse
    omega = cos(tk*2*pi/100); % vitesse angulaire

    % simulation de l'tat vrai (attention, inconnu du filtre)
    t = dt*(tk-1); % temps courant
    X_reel = X_reel + [V*dt*cos(X_reel(3)); V*dt*sin(X_reel(3)); dt * omega] + sqrtm(Q) *
        randn(d,1); % propagation de l'tat rel ( complter)
    X_reel(3) = mod(X_reel(3), 2*pi);

    % prediction ( complter : variables X_hat et P_hat)
    F = [1 0 -V*dt*sin(X_hat(3)); 0 1 V*dt*cos(X_hat(3)); 0 0 1]; % matrice de dynamique
    X_hat = X_hat + [V*dt*cos(X_hat(3)); V*dt*sin(X_hat(3)); dt*omega]; % tat estim prdit
    P_hat = F * P_hat * F' + Qf;
    X_hat(3) = mod(X_hat(3), 2*pi); % modulo sur le cap

    % gnration de la mesure relle ( complter)
    Y = [];
    for i = 1:size(amers,2)
        P_amer = amers(i).P;
```

```
Delta = P_amer(1:2) - X_reel(1:2); % attention, Delta rel inconnu du filtre
Yi = [sqrt(Delta(1)^2 + Delta(2)^2); atan2(Delta(2),Delta(1))-X_reel(3)] %+
      sqrt(Ri)*randn(2,1); % mesure sur l'amer i, complter
Yi(2) = mod(Yi(2),2*pi); % modulo sur le cap relatif
Y = [Y;Yi];
end
Y = Y + sqrtm(R)*randn(dm,1);

is_measurementValid = false;
% % validit de la mesure réelle
if (t == 1 || t == 2 || t == 3 || t == 4 || t == 5 || t == 6 || t == 7 || t == 8 || t == 9 ||
    t == 10)
    is_measurementValid = true;
end
% correction ( complter : variables K, P_hat, inno, X_hat)
H = [];
Yi = [];
if is_measurementValid
    for i = 1:size(amers,2)
        P_amer = amers(i).P;
        Delta_hat = P_amer(1:2) - X_hat(1:2);
        Hi = [-(Delta_hat(1)/(sqrt(Delta_hat(1)^2+Delta_hat(2)^2))),
              -(Delta_hat(2)/(sqrt(Delta_hat(1)^2+Delta_hat(2)^2))), 0; ...
              (Delta_hat(2)/(Delta_hat(1)^2+Delta_hat(2)^2)),
              -(Delta_hat(1)/(Delta_hat(1)^2+Delta_hat(2)^2)), -1];
        Yii = [sqrt(Delta_hat(1)^2 + Delta_hat(2)^2);
               atan2(Delta_hat(2),Delta_hat(1))-X_hat(3)];
        Yii(2) = mod(Yii(2),2*pi); % modulo sur le cap relatif
        H = [H;Hi];
        Yi = [Yi;Yii];
    end
    K = P_hat * H' * inv(Rf + H * P_hat * H');
    P_hat = (eye(d)-K*H)*P_hat;
    inno = Y - Yi;
    X_hat = X_hat + K * inno;
end
```

Q1 - Le code et les résultats

Cette algorithm se différencie par rapport au cas précédent pour le fait que la mis à jour des états internes du système n’est pas faite de façon linéaire mais elle utilise des matrices non linéaires. Il est possible de le voir dans la matrice F , dans la mise à jour de la valeur de X_{hat} et dans la mise à jour de la matrice H et dans le calcul de la valeur de Y .

La prédiction

Nous avons implémenté dans le code les équations suivantes:

$$\hat{X}_k = f(\hat{X}_{k-1}, u_k)$$

où maintenant f n’est plus une fonction linéaire mais non linéaire.

$$\hat{P}_k = F * \hat{P}_{k-1} * F^T + Q_k$$

$$F = \frac{\delta f}{\delta x}$$

où la matrice F est la matrice Jacobienne de la fonction par rapport aux états.

La correction

Nous avons implémenté dans le code les équations suivantes:

$$\hat{X}_k = \hat{X}_{k-1} + K * (Y_k - h(\hat{x}_{k-1}, u_k))$$

$$\hat{P}_k = (I_d - K * H) * \hat{P}_k$$

$$K = \hat{P}_k * H_k^T * (R_k + H_k * \hat{P}_k H_k^T)^{-1}$$

$$H = \frac{\delta h}{\delta x}$$

Notre cas

Je vais reporter les résultat ci-dessus:

C’est possible d’observer que, par rapport au cas précédent, le filtre prends plus de temps pour arriver à avoir une bonne estimation. Cela est dû au fait que l’évolution de Y est de type non linéaire et le filtre n’arrive pas toute de suite à suivre la bonne trajectoire.

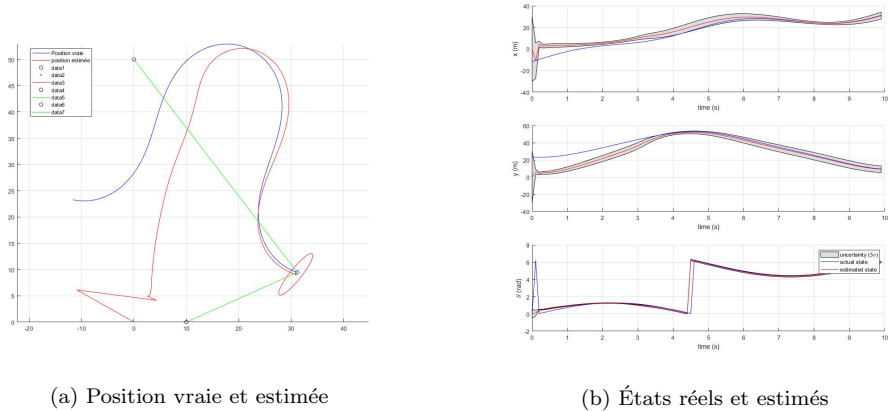


Figure 9: Les résultats obtenus du filtre de Kalman étendu

Q2 - Modification de la fréquence des mesures

Dans cette section j'ai modifié les fréquences des mesures afin d'avoir une correction chaque 1 second. J'ai obtenu un résultat intéressant car il converge plus rapidement vers la vraie position. En fait seulement après deux corrections dues aux mesures l'erreur est petit.

Pour une fréquence moins élevé nous avons eu les résultats suivantes:

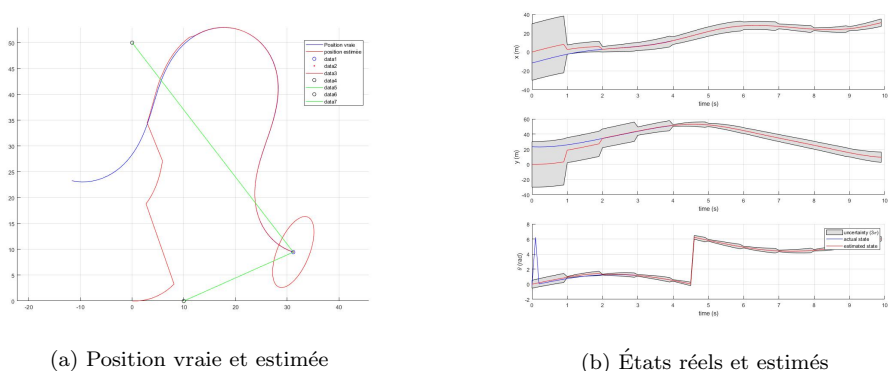


Figure 10: Les résultats obtenus du filtre de Kalman étendu avec une fréquence de correction de 1Hz

Nous pouvons observer que, dans ce cas, chaque second l'incertitude que l'on a sur les états se réduit: ce moment correspond à l'instant de temps quand on a le droit à effectuer l'étape de correction avec les mesures.

Nous avons utilisé *isMeasurementValid* afin d'avoir la disponibilité de mesures chaque 1 second et performer la correction à une fréquence de 1 Hz.

L'amélioration de la performance du filtre est donnée par le fait que nous avons une plus grande incertitude sur les états qui nous fait avoir une action du filtre plus grande sur la correction de la position estimée.

Q3 - Variation du bruit dynamique du filtre

Nous allons, dans ce cas, varier le bruit dynamique du filtre qui est représenté par la matrice Q_f et nous allons reporter ici les résultats. Dans cette question et les suivantes j'ai choisi de reporter au niveau normal la fréquence de la phase de correction.

Nous allons obtenir:

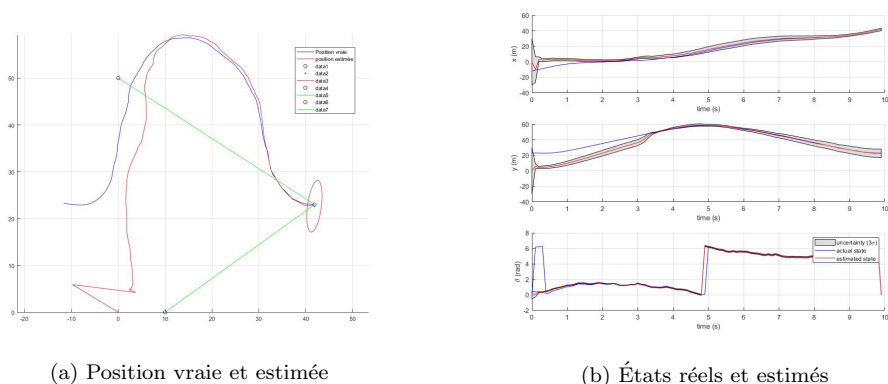


Figure 11: Les résultats obtenus du filtre de Kalman étendu avec bruit dynamique

Il est possible de voir que le bruit affecte notre observation étant donné que il est possible de voir des

variations autour de la trajectoire.

Si, par contre, nous augmentons fortement ce type de bruit d'un facteur 100 nous allons obtenir:

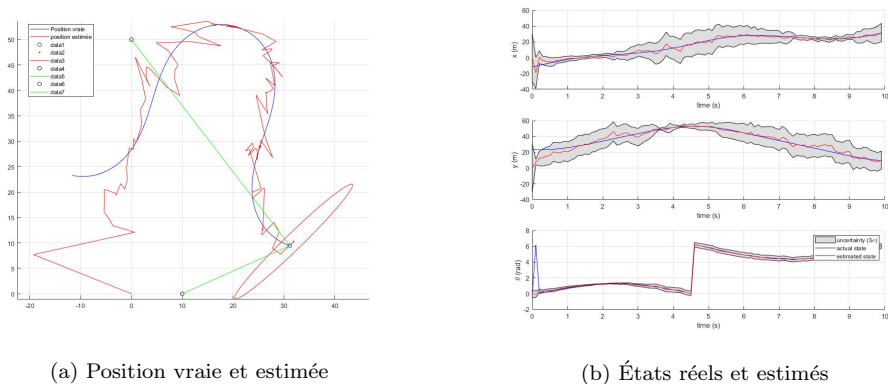


Figure 12: Les résultats obtenus du filtre de Kalman étendu avec un (fort) bruit dynamique

Il est évident que le performances sont moins bonnes car notre signal d'estimation est fortement influencé par le bruit introduit par la matrice Q_f . Dans la figure de gauche, il est possible d'observer que l'incertitude est aussi *bruitée* car elle continue à varier et osciller.

Nous avons aussi essayé avec des valeurs plus grands du bruit et, pour des valeurs trop grands, le filtre n'arrive pas à la convergence.

Q4 - Variation du bruit de mesure du filtre

Nous avons dans ce cas modifié le bruit de mesure du filtre qui est donnée par la matrice R_f . Cette matrice indique la covariance du bruit de mesure du filtre.

Nous avons obtenu:

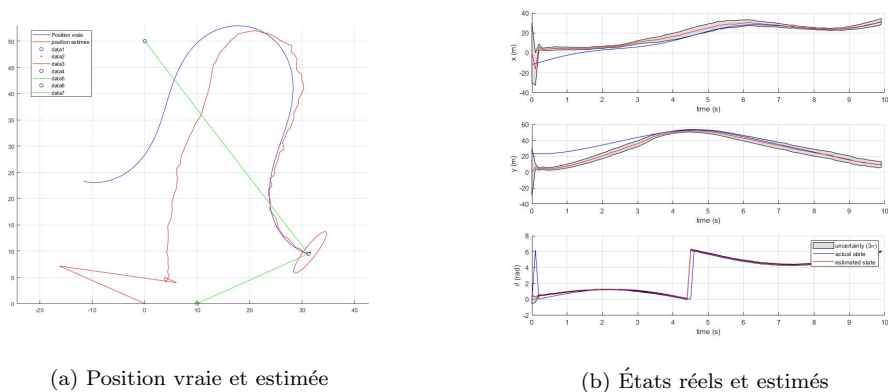


Figure 13: Les résultats obtenus du filtre de Kalman étendu avec bruit de mesure

Il est possible de voir que le bruit affecte notre observation étant donnée que il est possible de voir que les variations sont plus accentuées car le bruit est ajouté directement sur les mesures et son influence est plus forte.

Dans ce cas, nous avons aussi multiplié la matrice R pour une valeur de 100 et nous avons obtenu:

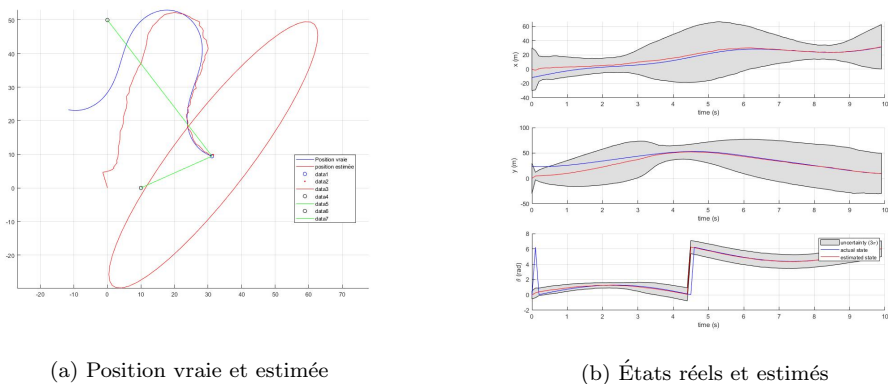


Figure 14: Les résultats obtenus du filtre de Kalman étendu avec (fort) bruit de mesure

Ce qu'il est possible d'observer c'est que en augmentant ce bruit, nous avons un incertitude beaucoup plus grande pour le fait que nous avons que notre phase de correction est affectée par le bruit.

Par contre, nous pouvons observer que sur la convergence vers la bonne trajectoire il n'y a pas trop de variation par rapport au cas avec un bruit de mesure petit: cela est dû au fait que ce bruit augmente la valeur de l'incertitude mais la performance du filtre est maintenue.

Q5 - Ajoute des autres *amers*

Dans cette section nous avons visualisé le changement dans le cas du changement de nombre de *amers*.

En particulier, si on ajoute des *amers* aléatoirement sans regarder où la trajectoire se développe, nous avons que le filtre peut arriver a ne pas converger. Si, par contre, on choisi des amers cohérents avec le problème, ils donneront encore plus d'informations au niveau des mesures sur le système et permettront au filtre d'attendre des performances meilleures. J'ai reporté les deux différents cas dans les deux figures suivantes.

Dans ce premier cas nous avons ajouté deux *amers* dans les positions (0;75) et (25;50) et nous avons obtenu:

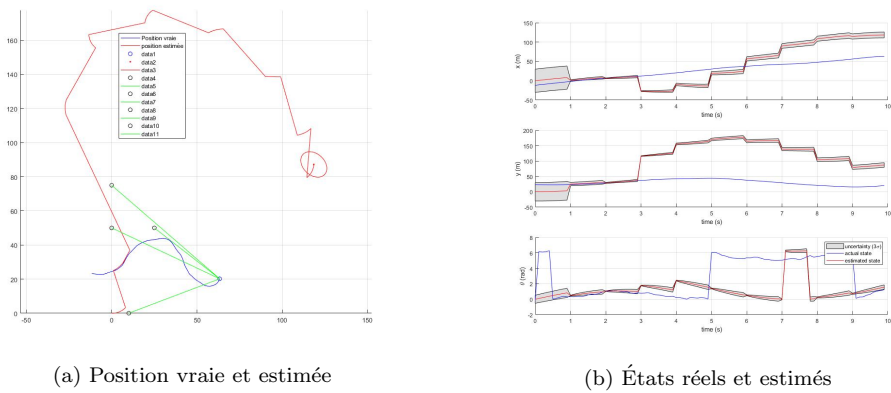


Figure 15: Les résultats obtenus du filtre de Kalman étendu avec une fréquence de correction de 1Hz et bruit de mesure et l'ajoute des amers

Il est possible d'observer que l'*amer* positionné en (25;50) n'aide pas le filtre a converger: c'est pour cela que j'ai changé sa position en (0;125). Les graphiques obtenu sont reporté dans les images suivantes où il est possible de voir que la rapidité de convergence du filtre est améliorée.

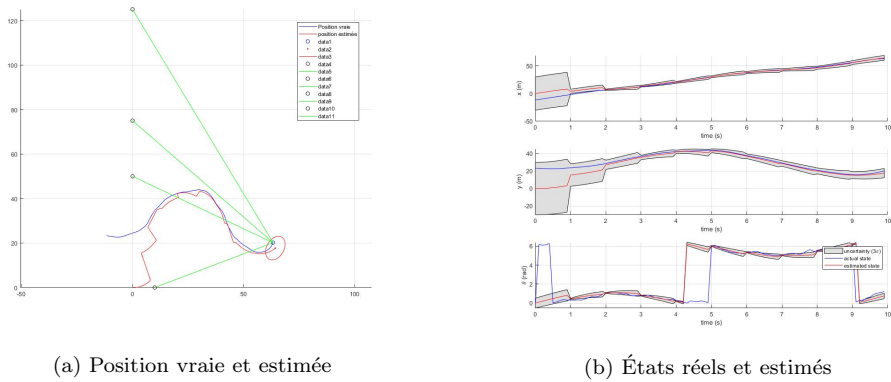


Figure 16: Les résultats obtenus du filtre de Kalman étendu avec une fréquence de correction de 1Hz et bruit de mesure et l'ajoute des amers