

Cartographie et filtrage de Kalman

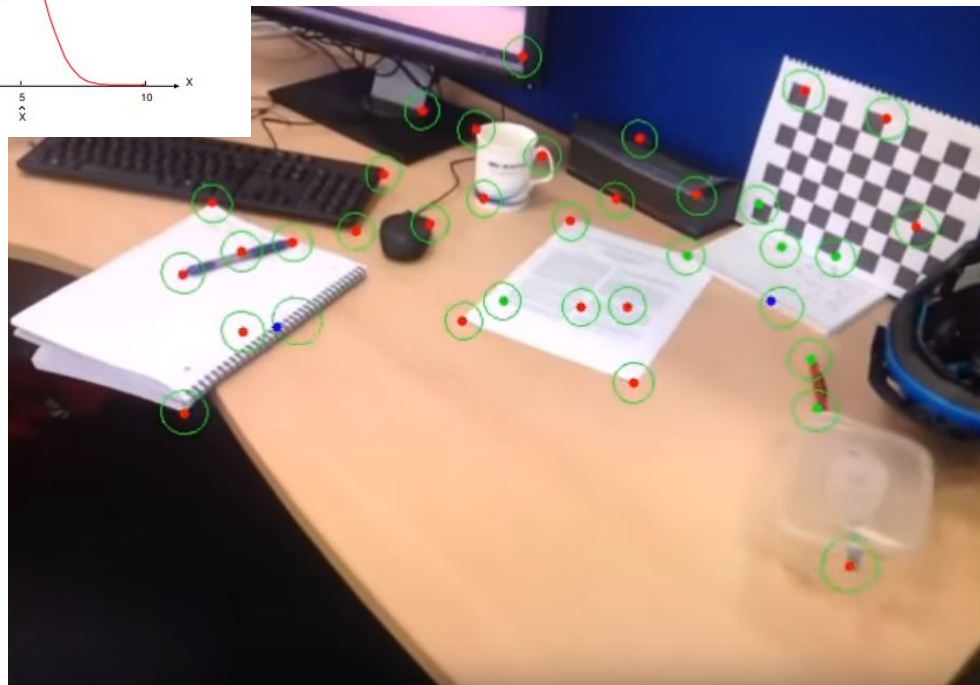
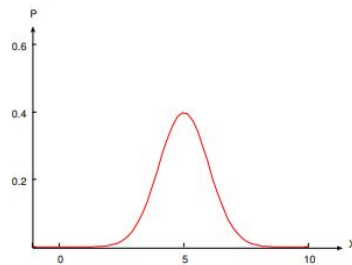
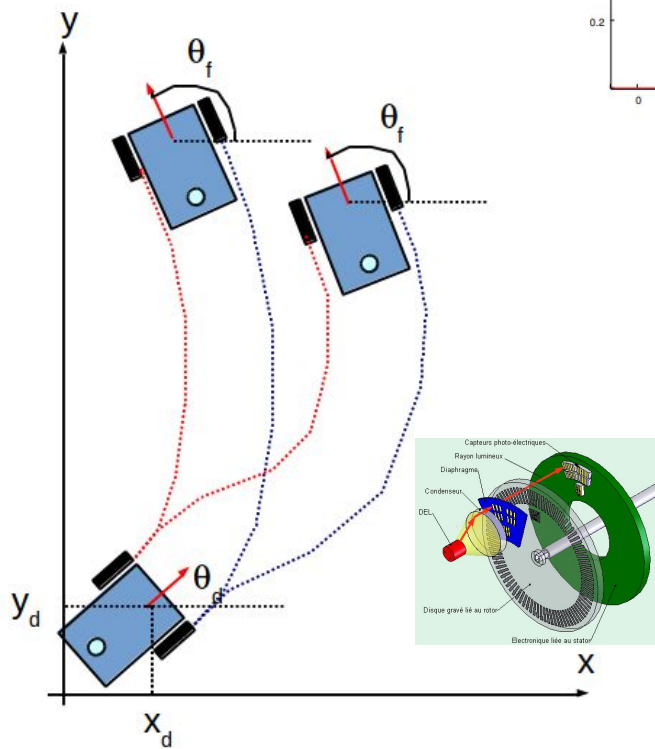
Marin STAMM
Pierre POTEL



Où suis-je? Où vais-je?

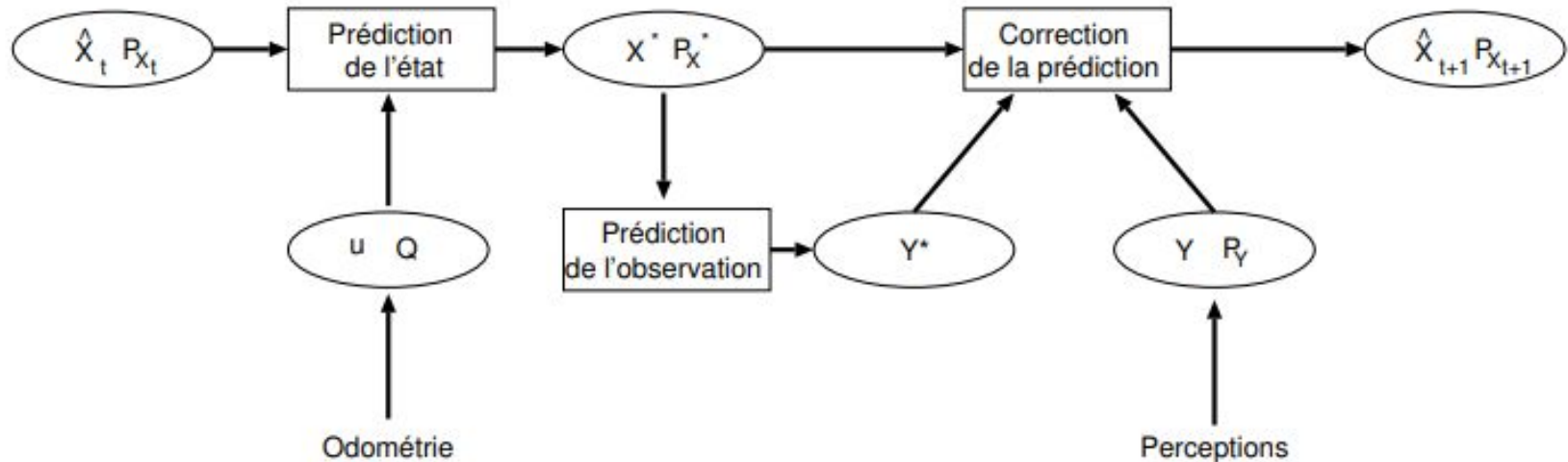


Mesures



Filtre de Kalman

X : vecteur position du robot
 P_x : covariance de la position
 u : odométrie
 Q : covariance de l'odométrie
 Y : vecteur observation
 P_y : covariance de l'observation



Simulation 1 :

**Amers tous détectés et dans le
même ordre**

—

Code

- Ouvrir le fichier
- Lire la première perception dans le fichier
- Initialiser les variables du filtre
- Afficher la carte (pause)
- Tant que le fichier n'est pas vide :
 - Lire une odométrie
 - Effectuer la prédiction de l'état
 - Afficher la carte (pause)
 - Lire une perception
 - Effectuer la prédiction de l'observation
 - Effectuer la correction
 - Afficher la carte (pause)

```
function []=main_observSimpl(fichier);

close all;
addpath(genpath('.'));
clc;
rng(123456);
lambda=0.7;

fileID=fopen(fichier,'r');
red=textscan(fileID,'percep :');
data=textscan(fileID,'%f');
Y=cell2mat(data);

[X,P,A,B,H]=init(Y,lambda);

trajectoirecorr=[X(1,1);X(2,1)];
trajectoirepred=[X(1,1);X(2,1)];
affichage(X,P,trajectoirecorr,trajectoirepred);

pause(1);

while ~feof(fileID)

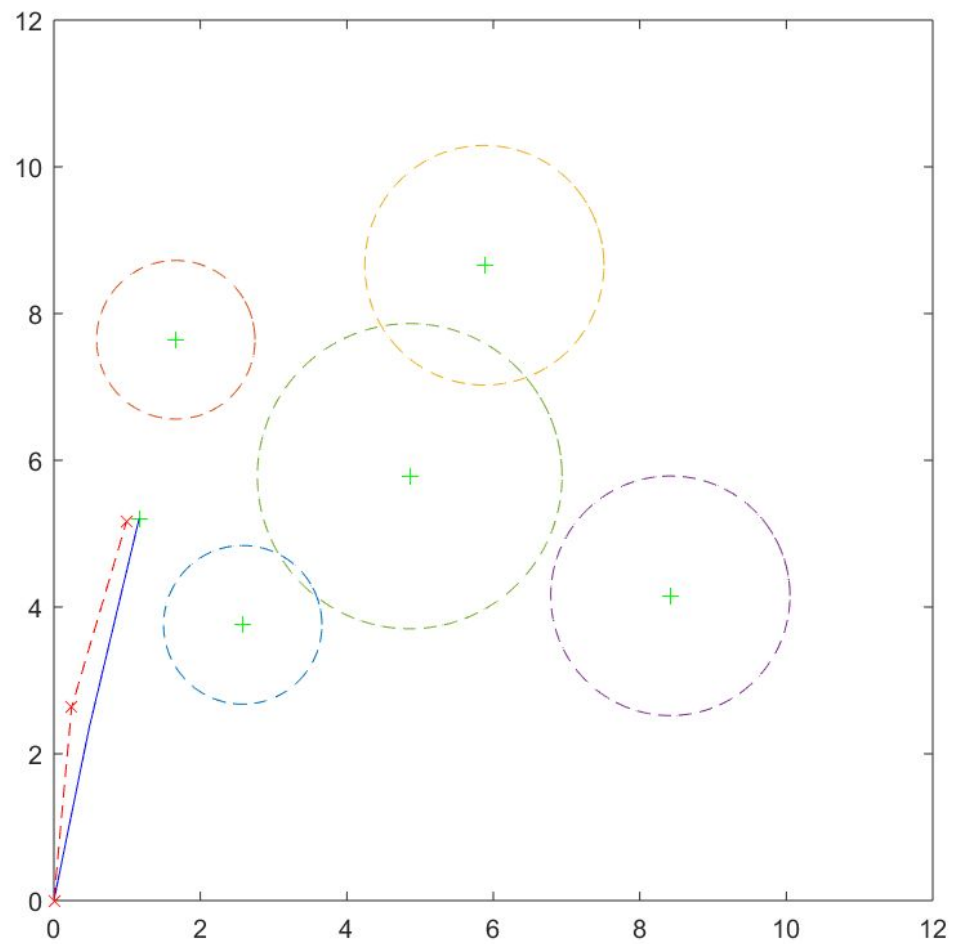
    red=textscan(fileID,'odom :');
    odom=textscan(fileID,'%f');
    u=cell2mat(odom);
    Q=covodo(u,1);

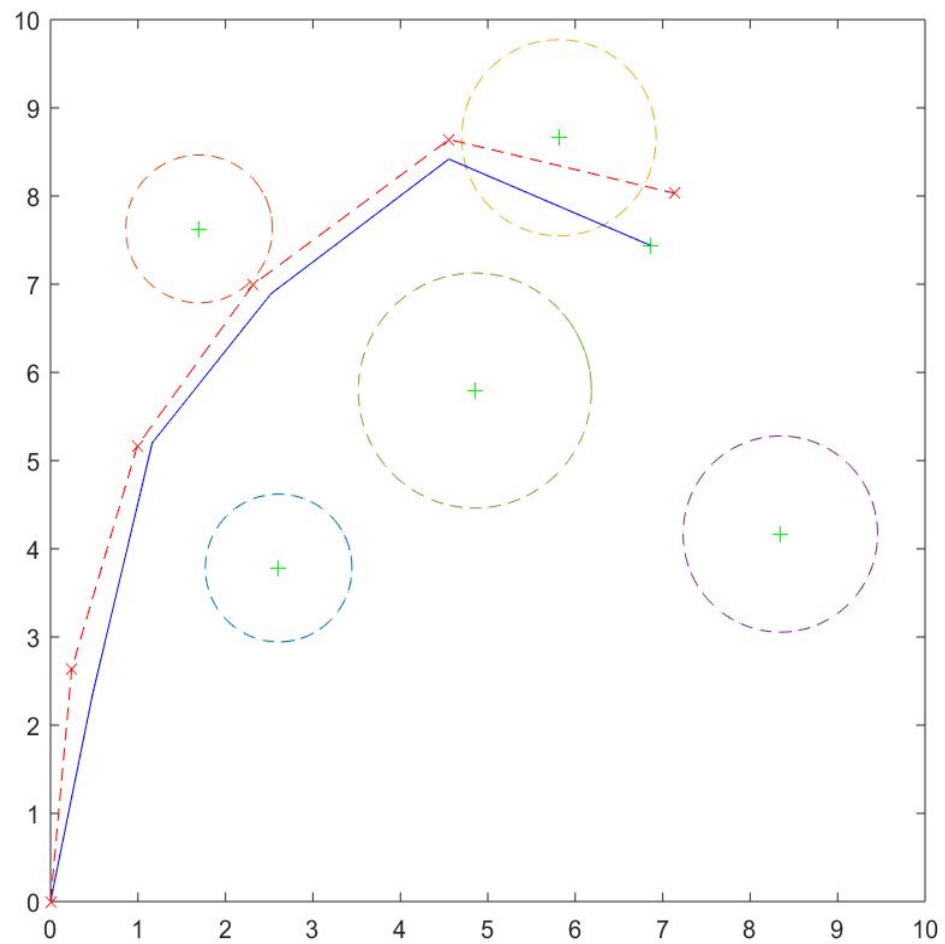
    [Xstar Pstar]=statepred(X,u,B,P,Q);
    trajectoirepred=[trajectoirepred;trajectoirepred(size(trajectoirepred,1)-1:end) + u];

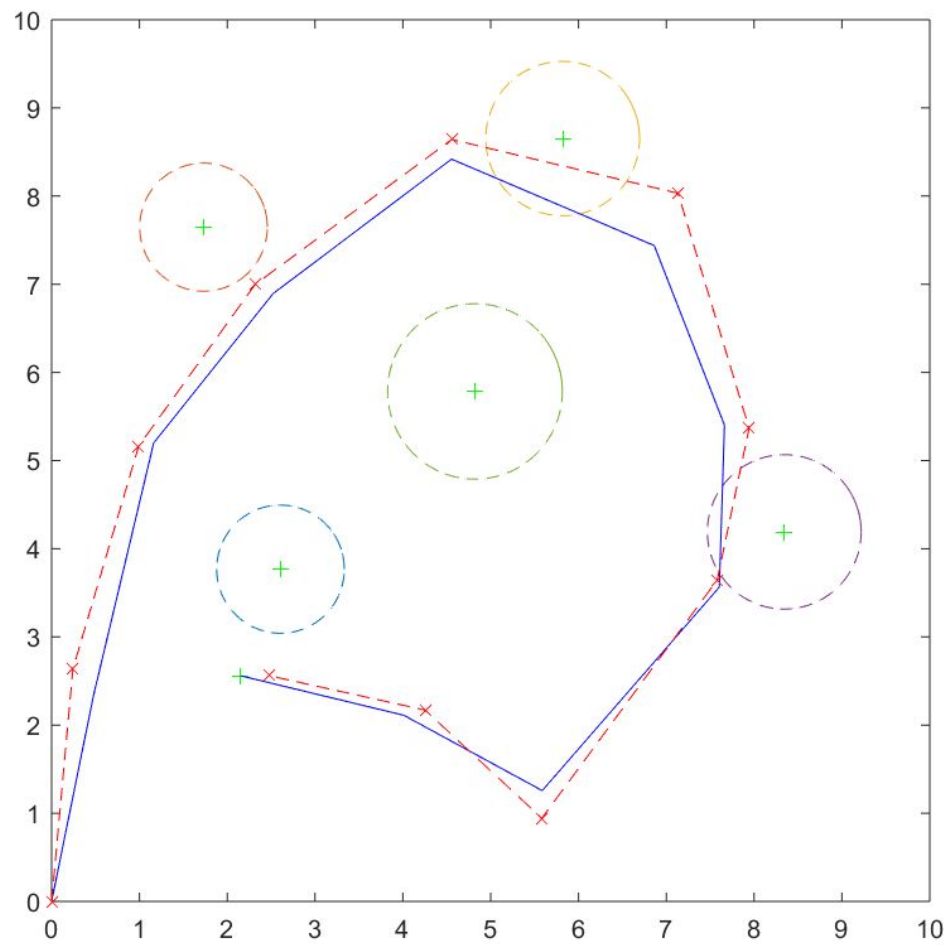
    affichage(Xstar,Pstar,trajectoirecorr,trajectoirepred);
    pause(0.2);
    red=textscan(fileID,'percep :');
    data=textscan(fileID,'%f');
    Y=cell2mat(data);
    Ystar = obspred(H,Xstar);

    Py = covobs(X,0.7);
    [P X]=corstate (Xstar,Pstar,Ystar,Y,H,Py);
    trajectoirecorr=[trajectoirecorr;X(1,1);X(2,1)];
    affichage(X,P,trajectoirecorr,trajectoirepred);

    pause(0.2);
end
```





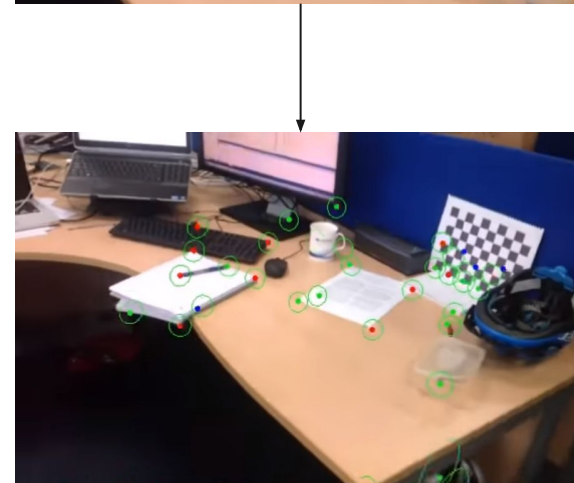
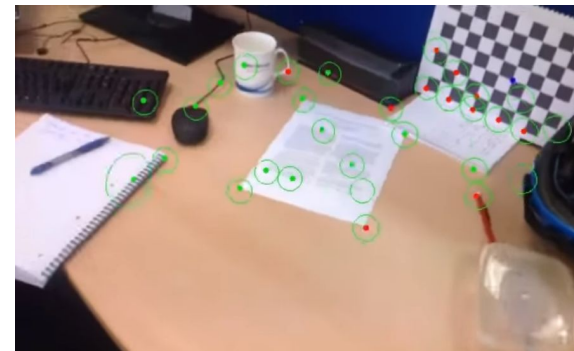
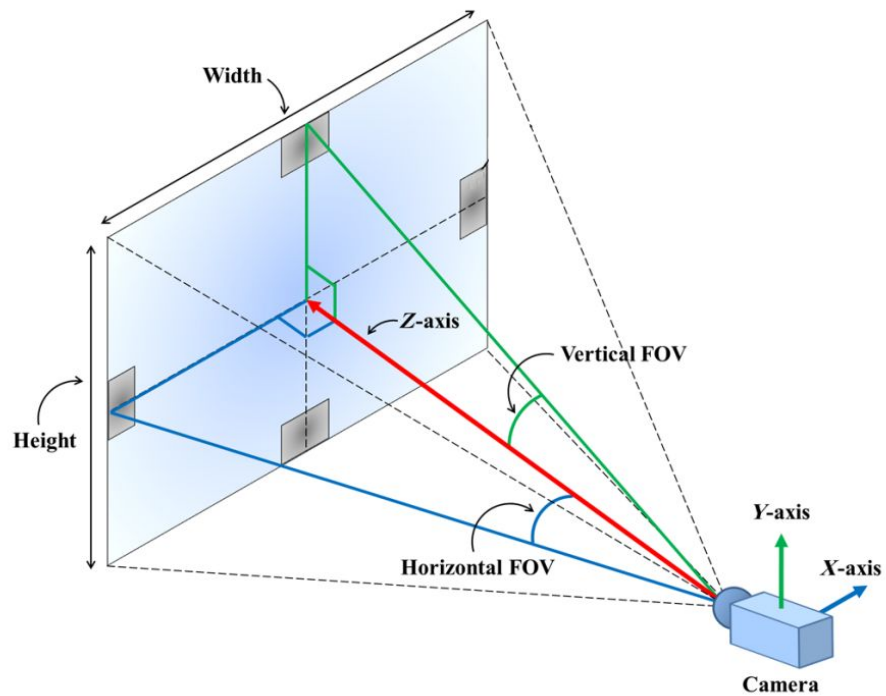


Simulation 2 :

**Amers détectés en partie sans
connaître leur ordre**

—

Plus réaliste...



Code V2

- Ouvrir le fichier
- Lire la première perception dans le fichier
- Initialiser les variables du filtre
- Afficher la carte (pause)
- Tant que le fichier n'est pas vide :
 - Lire une odométrie
 - Effectuer la prédiction de l'état
 - Afficher la carte (pause)
 - Lire une perception
 - Déterminer Y_{known} , Y_{new} et construire la matrice H
 - Effectuer la prédiction de l'observation Y_{known}^* à partir de X^* et H
 - Effectuer la correction en fonction de Y_{known}^* et Y_{known}
 - Ajouter les nouveaux amers et mettre à jour les matrices du filtre en fonction de Y_{new}
 - Afficher la carte (pause)

$$X = \begin{bmatrix} 1 & 1 & 4 & 4 & 5 & 1 \end{bmatrix}^T$$

$$Y = \begin{bmatrix} 3 & 0 & 4 & 4 & 0 & 4 \end{bmatrix}^T$$

$$Y_{known} = \begin{bmatrix} 3 & 0 & 4 & 4 \end{bmatrix}^T Y_{new} = \begin{bmatrix} 0 & 4 \end{bmatrix}^T$$

```
function []=main_observPartiel(fichier)

close all;
addpath(genpath('.'));
clc;
rng(123456);
seuil=2;
lambda=0.1;

fileID=fopen(fichier,'r');
textscan(fileID,'percep :');
data=textscan(fileID,'%f');
Y=cell2mat(data);

[X,P,A,B,H]=init(Y,lambda);
trajectoirecorr=[0;0;X(1,1);X(2,1)];
trajectoirepred=[0;0;X(1,1);X(2,1)];
affichage(X,P,trajectoirecorr,trajectoirepred);

pause(1);

while ~feof(fileID)

    textscan(fileID,'odom :');
    odom=textscan(fileID,'%f');
    u=cell2mat(odom);
    Q=covodo(u,0.7);

    [Xstar,Pstar]=statepred(X,u,B,P,Q);
    trajectoirepred=[trajectoirepred;trajectoirepred(size(trajectoirepred,1)-1:end) + u];
    affichage(Xstar,Pstar,trajectoirecorr,trajectoirepred);
    pause(0.1);

    textscan(fileID,'percep :');
    data=textscan(fileID,'%f');
    Y=cell2mat(data);

    for i = 4:size(Y,1)/2
        Y(2*i-1) = Y(2*i-1)+trajectoirecorr(size(trajectoirecorr,1)-1);
        Y(2*i) = Y(2*i)+trajectoirecorr(size(trajectoirecorr,1));
    end

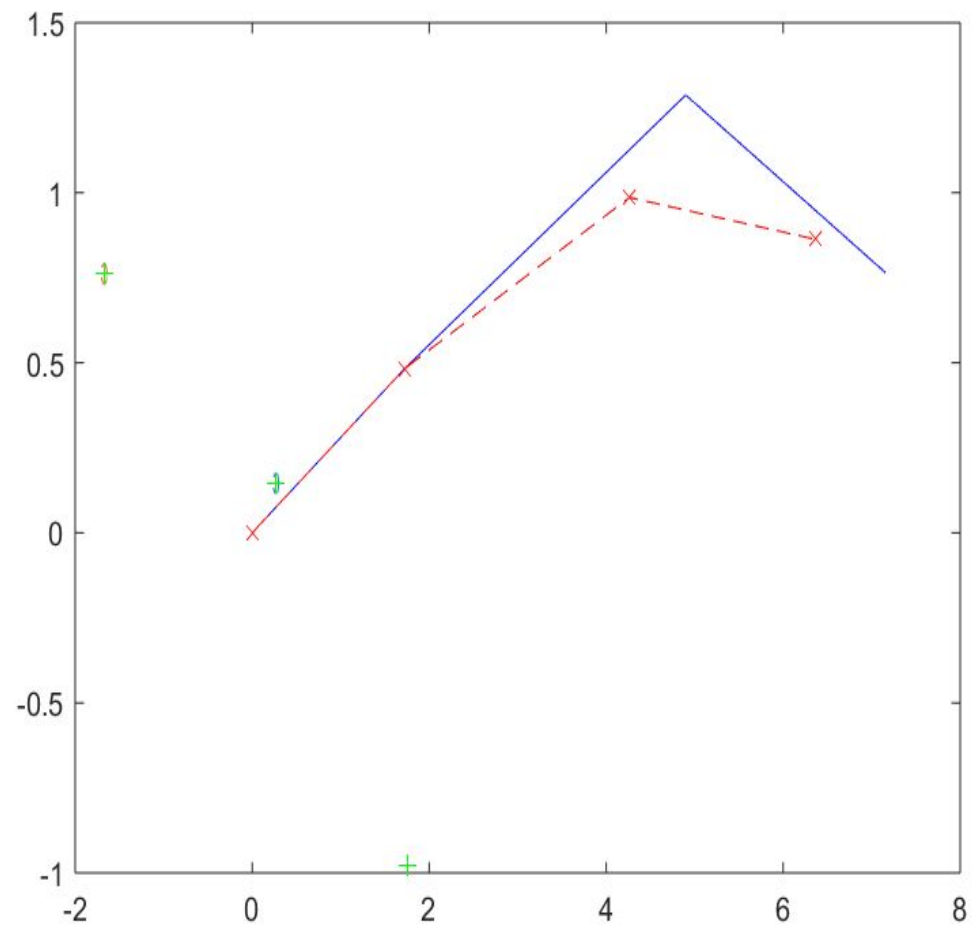
    [H,Yknown,Ynew] = nouveauH(Xstar,Y,seuil);

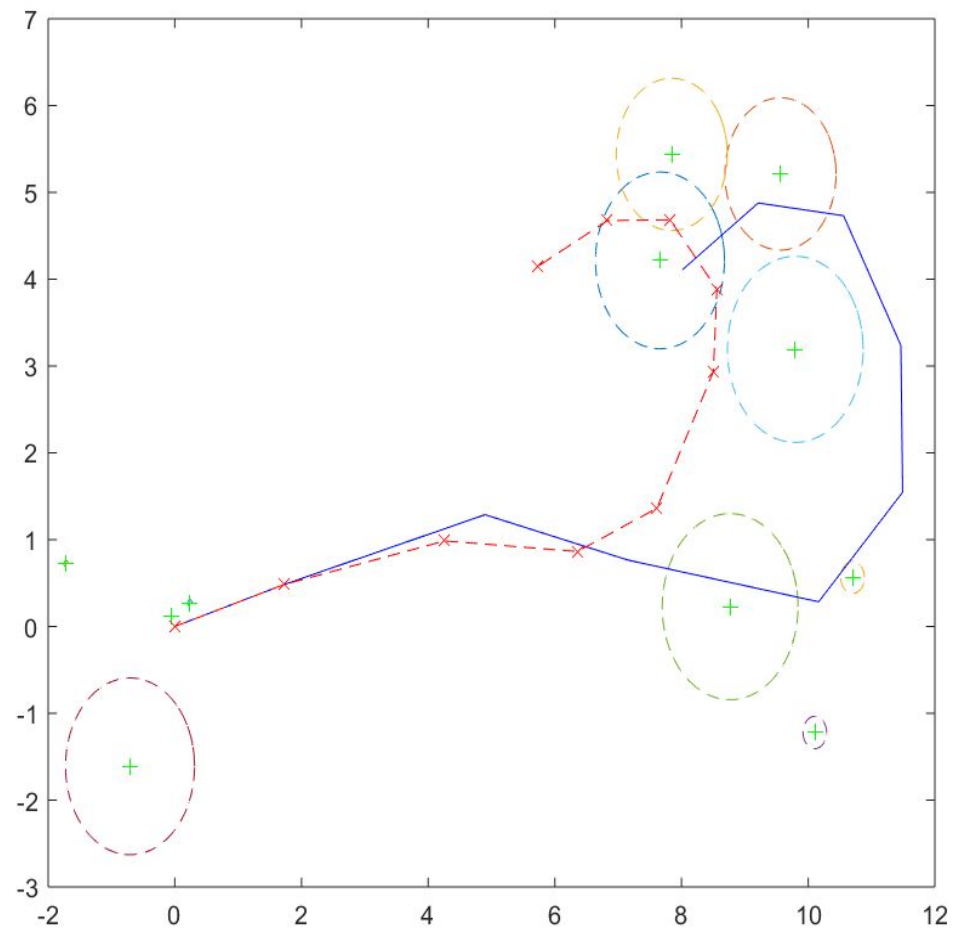
    Yknownstar = obspred(H,Xstar);
    Py = covobs([X(1:2);Yknown],0.7);
    [P,X]=corstate(Xstar,Pstar,Yknownstar,Yknown,H,Py);
    trajectoirecorr=[trajectoirecorr;trajectoirecorr(size(trajectoirecorr,1)-1:end) + Xstar(1:2)];

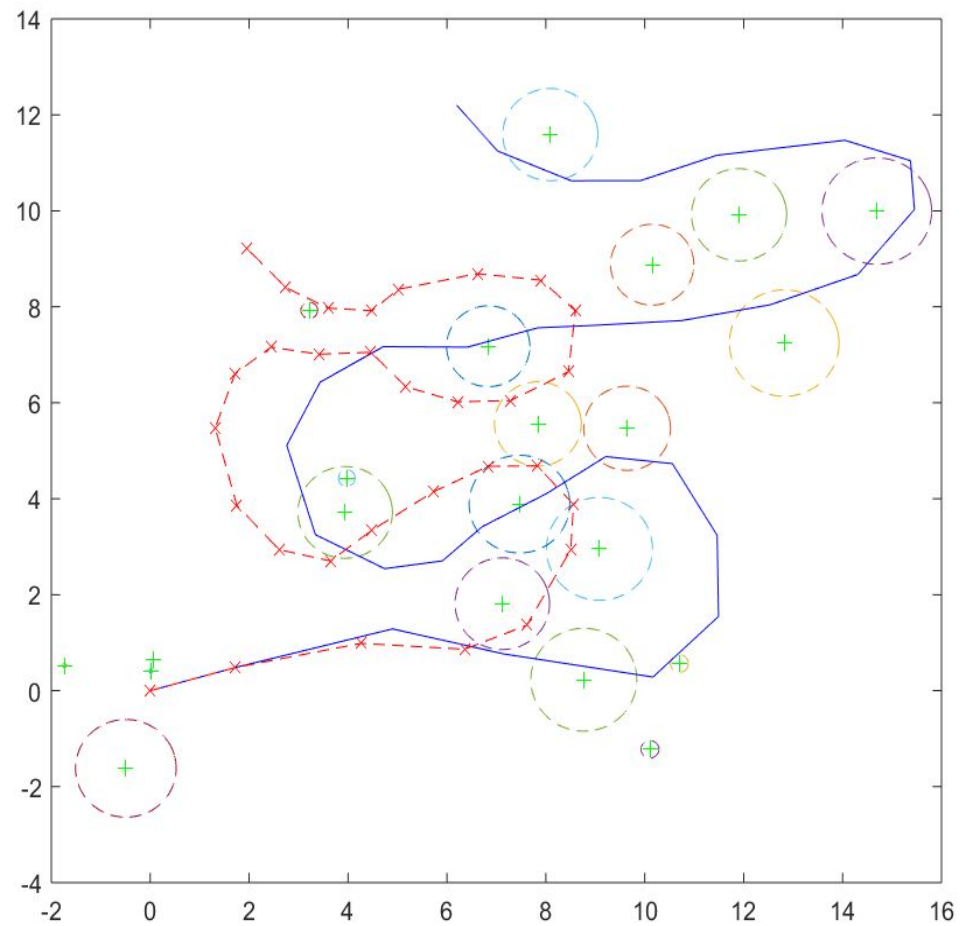
    [X,P,A,B]=nouveauPX(X,P,A,B,Ynew,lambda);
    affichage(X,P,trajectoirecorr,trajectoirepred);

    pause(0.1);
end

fclose(fileID);
```







Conclusion

