Filtre de Kalman Etendu

3A Spécialité SYSMER

Année universitaire 2020/2021

Elie Boltenhagen
Sandrine Wolfer

M. JAUFFRET

Table des matières

1 Introduction

On considère deux objets (assimilés à des points mathématiques) dans un plan muni d'un repère orthonormé. L'un est l'observateur, noté O. L'autre, noté T, est appelé « cible ». L'observateur est immobile et positionné sur l'origine du repère. La cible se déplace selon un mouvement rectiligne uniforme. Sa vitesse est notée $v = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}$. Ainsi, sa position à l'instant t est $P(t) = P(0) + tV$. A chaque instant $t_k = (k-1)\Delta t$, l'observateur mesure l'angle de visée dans lequel il détecte la cible, relativement à l'axe (O_y). La mesure d'angle acquise à l'instant t_k et notée Z_k obéit à l'équation de mesure $Z_k = Arctg\left[\frac{x_k}{y_k}\right] + \varepsilon_k$, où (x_k, y_k) sont les coordonnées de $P(t_k)$. Le bruit de mesure ε_k est gaussien, centré, de variance σ^2 . Les bruits de mesures sont indépendants entre eux.

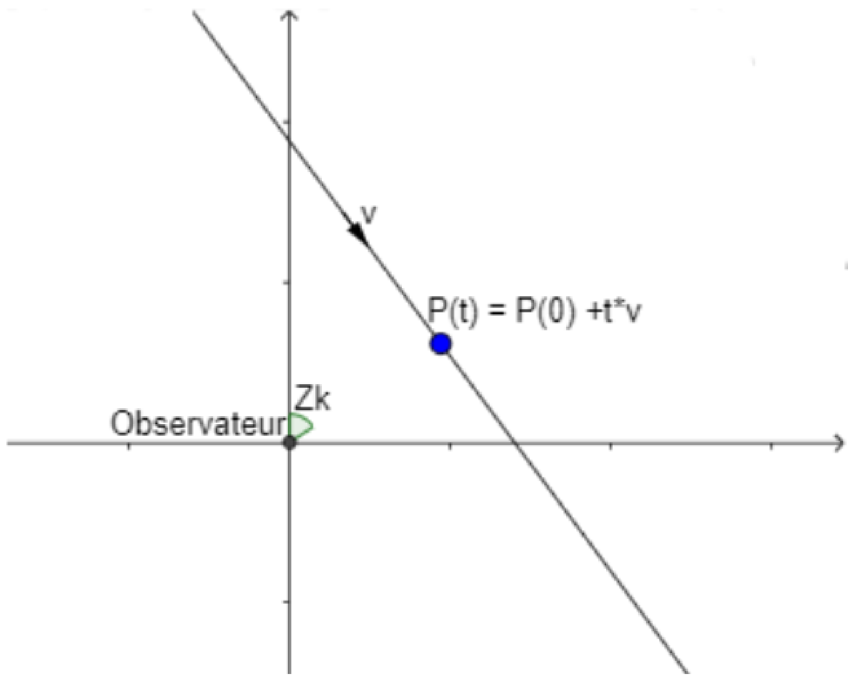

Figure 1 – Schéma du montage

Le but de ce mini projet est d'estimer par le filtre de Kalman étendu (EKF) les positions successives à partir des angles mesurés depuis l'observateur.

Les hypothèses sont les suivantes :

— L'observateur connaît l'écart-type du bruit de mesure, sa propre position et la vitesse de la cible.

— La cible ne se dirige jamais vers l'observateur.

2 Équation de Kalman

$$\underbrace{\begin{pmatrix} x_k \\ y_k \end{pmatrix}}_{\xi_k} = \begin{pmatrix} x_{k-1} + \Delta t \dot{x} \\ y_{k-1} + \Delta t \dot{y} \end{pmatrix}$$

donc :

$$\left.\begin{aligned}\xi_k = \underbrace{\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}}_{F_k}\xi_{k-1} + \underbrace{\begin{pmatrix}\Delta t\dot{x}\\ \Delta t\dot{y}\end{pmatrix}}_{connu}\end{aligned}\right\}\textbf{Équation de commande}$$

$$\left. Z_k = Arctg\begin{pmatrix}x_k\\ y_k\end{pmatrix} + \varepsilon_k \right\}\textbf{Équation de mesure}$$

Ensuite on fait l'initialisation, On choisie :

— $\xi_{0|0} = rg\begin{pmatrix}\sin(Z_1)\\ \cos(Z_1)\end{pmatrix}$, avec rg la distance "intuitive".

— $P_{0|0} = 10^6 I_k$

Dans la boucle de Kalman on fait :

— $\xi_{k|k+1} = \underbrace{\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}}_{F_k}\xi_{k-1|k-1} + \Delta t\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix}$

— $P_{k|k-1} = F_k P_{k-1|k-1} F_k^T$, Nous avons pas de bruit d'état

— Le gain $H_k = \nabla^T \mathcal{H}_k \xi_{k|k-1}$ avec

$$\begin{aligned}\mathcal{H}_k &= Arctg\left(\tfrac{x}{y}\right)\\ \nabla\mathcal{H}_k &= \begin{bmatrix}\frac{\partial}{\partial x}Arctg\left(\frac{x}{y}\right)\\ \frac{\partial}{\partial y}Arctg\left(\frac{x}{y}\right)\end{bmatrix}\\ &= \begin{bmatrix}\frac{1}{1+(\frac{x}{y})^2}\cdot\frac{1}{y}\\ \frac{1}{1+(\frac{x}{y})^2}\cdot(-\frac{x}{y^2})\end{bmatrix}\\ &= \begin{bmatrix}\frac{y}{x^2+y^2}\\ -\frac{x}{x^2+y^2}\end{bmatrix}\end{aligned}$$

3 Code Matlab

Maintenant nous mettons en place un Code Matlhab qui permet de mettre en oeuvre notre filtre de Kalman étendu.

Dans un premier temps on initialise le programme. C'est à dire que nous créons le vecteur vitesse v à partir d'un point d'origine et nous lui attribuons une trajectoire. :

```
%initialisation
t=1:1000;
deltat=1;

xpoint = 5; %m/s
ypoint = -7 ; %m/s
v=[xpoint, ypoint];% vecteur vitesse du projectile

sigma=10*pi/180; % ecart type
epsilon=sigma*randn(1,length(t));


Zsb=zeros(1,length(t));
z=zeros(1,length(t));
Ksi=zeros(2,length(t));
Ksik=zeros(2,length(t));

%P=zeros(2,2,length(t));
Hk=zeros(2,length(t));
%K=zeros(2,length(t));

p=zeros(2,2,length(t));
X=zeros(1,length(t));
Y=zeros(1,length(t));

F=[1 0 ; 0 1];

X(1)=1000;
Y(1)=1000;

Zsb(1)=atan2(X(1),Y(1)); % signal sans bruit
z(1)=Zsb(1)+epsilon(1);  % signal bruité

rg=1400; % distance intuitive (la distance que l'on pense entre nous et l'objet)

Ksik(:,1)=rg*[sin(z(1)) cos(z(1))];

p=10^6*[1 0 ; 0 1];
```

FIGURE 2 – code initialisation

De part la trajectoire que nous avons crées nous pouvons récuperer l'angle a un instant t. Nous notons cette angle Zsb. Ensuite nous bruitons notre angle pour avoir l'angle bruité Z. Avec cette angle bruité et le filtre de Kalman étendu nous pouvons retomber sur la trajectoire que nous avons crée précedement.

```
for i=2:length(t)
    X(i)=X(i-1)+deltat*v(1);
    Y(i)=Y(i-1)+deltat*v(2);
    P=F*p*F';

    Zsb(i)=atan2(X(i),Y(i));
    z(i)=Zsb(i)+epsilon(i);

    Ksi(:,i)=F*Ksik(:,i-1)+deltat*v';


    Hk=1/(Ksi(1,i)^2+Ksi(2,i)^2)*[Ksi(2,i) -Ksi(1,i)];
    K=P*Hk'*inv(Hk*P*Hk'+sigma^2);

    Ksik(:,i)=Ksi(:,i)+K*(z(i)-atan2(Ksi(1,i),Ksi(2,i)));
    p=P-K*Hk*P;
end
```

FIGURE 3 – code boucle de Kalman étendu

Pour finir on affiche nos résultat sur un graphique :

```
%  Affichage des résulats
figure(1)
plot(Ksik(1,:),Ksik(2,:), 'b')
hold on
plot(X,Y, 'g')
legend('prédiction','trajectoire réel')
xlabel('x (en m)')
ylabel('y (en m)')

figure(2)

plot(t,Zsb, 'g','linew',5)
hold on
plot(t,z, 'r')
legend('angle sans bruit','angle bruité')
xlabel('t (en s)')
ylabel('angle (en rad)')
```

FIGURE 4 – code tracer les courbes

4 Interprétation du résultat

Ici nous pouvons voir en vert l'angle non bruité qui vient de la trajctoire donnée à notre objectif. Et en rouge nous pouvons voir l'angle que nous avons bruité pour pouvoir l'utiliser dans le filtre de Kalman étendu.

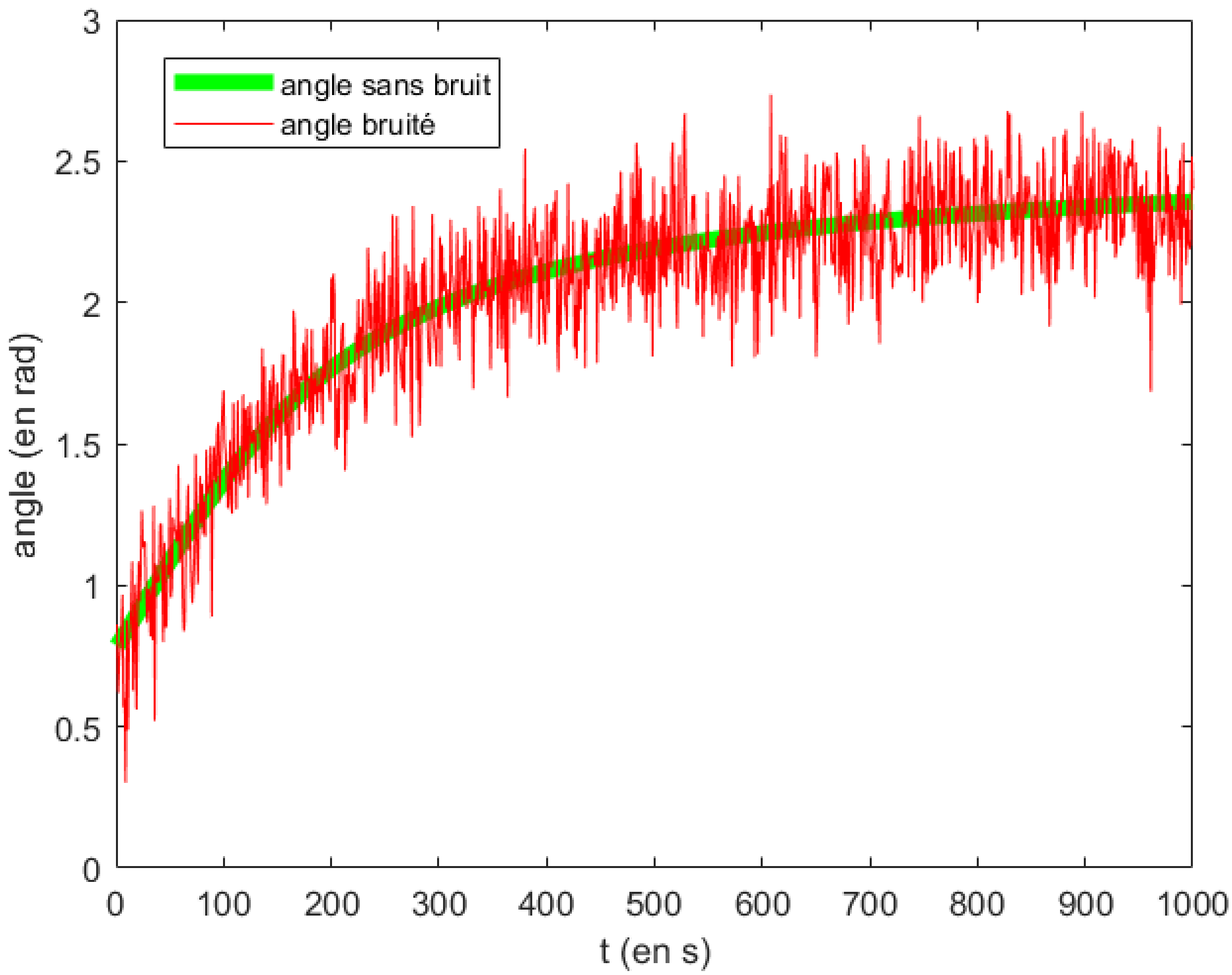

FIGURE 5 – Evolution de l'angle entre observateur et objectif

Nous pouvons voir en vert la trajectoire réeel de notre objectif. Et en bleu nous avons la prédiction de la trajectoir

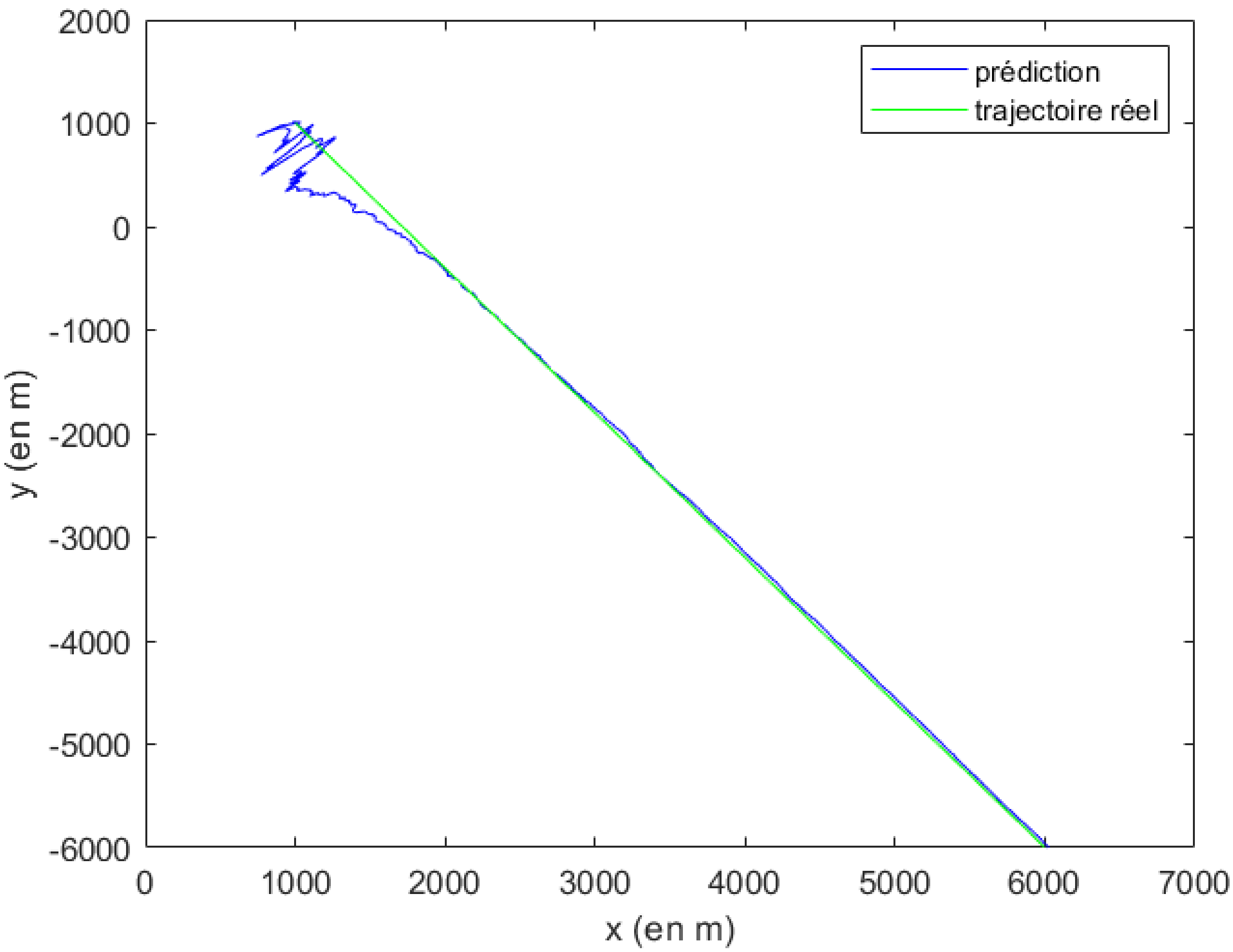

FIGURE 6 – Evolution de l'angle entre observateur et objectif