Filtre de Kalman et Signal

3A Spécialité SYSMER

Année universitaire 2020/2021

Elie Boltenhagen
Sandrine Wolfer

M. JAUFFRET

Table des matières

1 Introduction

Le but de se TP est de comprendre le fonctionnement du filtre de Kalman et de construire un programme sur Matlab permettant d'utiliser ce filtre.

On considère le signal déterministe à temps discret $(Z_k,\ k \in \mathbb{Z})$ défini par :

$$Z_k = a\cos(\omega k + \phi) \tag{1}$$

où l'amplitude a de la phase ϕ sont inconnues. La pulsation est connue.

2 Question 1

Calcul des deux constantes a_1 et a_2 telles que $Z_k = a_1 Z_{k-1} + a_2 Z_{k-2}$

$$\begin{aligned}
& Z_k = a_1 Z_{k-1} + a_2 Z_{k-2} \\
\Leftrightarrow \quad & a\cos(\omega k + \phi) = a_1 a\cos(\omega(k-1) + \phi) + a_2 a\cos(\omega(k-2) + \phi) \\
\Leftrightarrow \quad & \cos(\omega k + \phi) = a_1\cos(\omega(k-1) + \phi) + a_2\cos(\omega(k-2) + \phi) \\
\Leftrightarrow \quad & \cos(\omega k + \phi) = a_1\cos(\omega k + \phi)\cos(\omega) + a_1\sin(\omega k + \phi)\sin(\omega) + a_2\cos(\omega k + \phi)\cos(2\omega) + a_2\sin(\omega k + \phi)\sin(2\omega) \\
\Leftrightarrow \quad & \cos(\omega k + \phi) = \underbrace{(a_1\cos(\omega) + a_2\cos(2\omega))}_{=1}\cos(\omega k + \phi) + \underbrace{(a_1\sin(\omega) + a_2\sin(2\omega))}_{=0}\sin(\omega k + \phi)
\end{aligned}$$

Nous arrivons sur 2 équations et 2 inconnus :

$$\begin{cases} a_1\cos(\omega) + a_2\cos(2\omega) &= 1 \\ a_1\sin(\omega) + a_2\sin(2\omega) &= 0 \end{cases}$$

En conslusion :

$$\boxed{\begin{cases} a_1 &= 2\cos(\omega) \\ a_2 &= -1 \end{cases}} \tag{2}$$

3 Question 2

Ensuite, on en déduit la matrice F carrée telle que

$$\begin{bmatrix} Z_k \\ Z_{k-1} \end{bmatrix} = F \begin{bmatrix} Z_{k-1} \\ Z_{k-2} \end{bmatrix} \tag{3}$$

D'après nos résultat précèdent on trouve :

$$\begin{bmatrix} Z_k \\ Z_{k-1} \end{bmatrix} = \underbrace{\begin{pmatrix} 2\cos(\omega) & -1 \\ 1 & 0 \end{pmatrix}}_{F} \begin{bmatrix} Z_{k-1} \\ Z_{k-2} \end{bmatrix} \tag{4}$$

4 Question 3

On pose $X_k = \begin{bmatrix} Z_k \\ Z_{k-1} \end{bmatrix}$, on détermine la matrice H telle que :

$$z_k = HX_k,\ \forall\, k \geq 0. \tag{5}$$

Donc

$$\boxed{H = \begin{bmatrix} 1 & 0 \end{bmatrix}} \tag{6}$$

5 Question 4

On observe un signal aléatoire $(y_k, k \in \mathbb{Z})$ définit par $y_k = z_k + \varepsilon_k$:

— Équation de mesure : $y_k = H_k x_k + \varepsilon_k$

— Équation d'état : $x_k = F_k X_{k-1}$, ici $G - k = 0$ et $\nu_k = 0$

Il faut faire attention que l'on respecte bien le théormème de Shannon-Nyquist, on prend $\Delta t = 1$ la péridode d'échantillonage. le théorème dit que $f_e > 2f_{max}$, ici $\omega = 2\pi f$ donc $\omega < \pi$ sinon le filtre ne fonctionne pas.

6 Question 5

Pour l'initialisation de $X_{0|0}$ nous choissons un vecteur null :

$$\boxed{X_{0|0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}} \tag{7}$$

7 Question 6

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

Dans un premier temps on initialise le programme :

```
%Initialisation
t=1:300;
Omega=1;

F=[2*cos(Omega) -1 ; 1 0];
Hk=[1 0];

epsilon=2;
psi=0;

z=zeros(1,length(t));
x=zeros(2,length(t));

x(:,1)=[pi/3;0];

a=1;
Zsb(t)=a*cos(Omega*t+psi); % signal sans bruit
z(t)=Zsb(t)+epsilon*randn(1,length(t));   % signal bruité

y(1)=Hk*x(:,1)+epsilon*randn;
```

Figure 1 – code initialisation

Nous avons créer un signal bruité.

Puis on créer la boucle pour le filtre de Kalman :

```
% Boucle filtre de Kalman
for k=2:length(t)
    Xkk=F*x(:,k-1);
    P=F*p*F'
    K=P*Hk'*inv(Hk*P*Hk'+epsilon^2);
    x(:,k)=Xkk+K*(z(k)-Hk*Xkk);
    p=P-K*Hk*P;
end
```

FIGURE 2 – code bloucle de Kalman

Pour finir on affiche nos résultat sur un graphique :

```
%  Affichage des résulats
figure
plot(z,'r')
hold on
plot(x(1,:),'b')
hold on
plot(Zsb,'g')
legend('bruité','prédiction','non bruité')
xlabel('temps')
ylabel('amplitude')
```

FIGURE 3 – code tracer les courbes

8 Interprétation des résultats

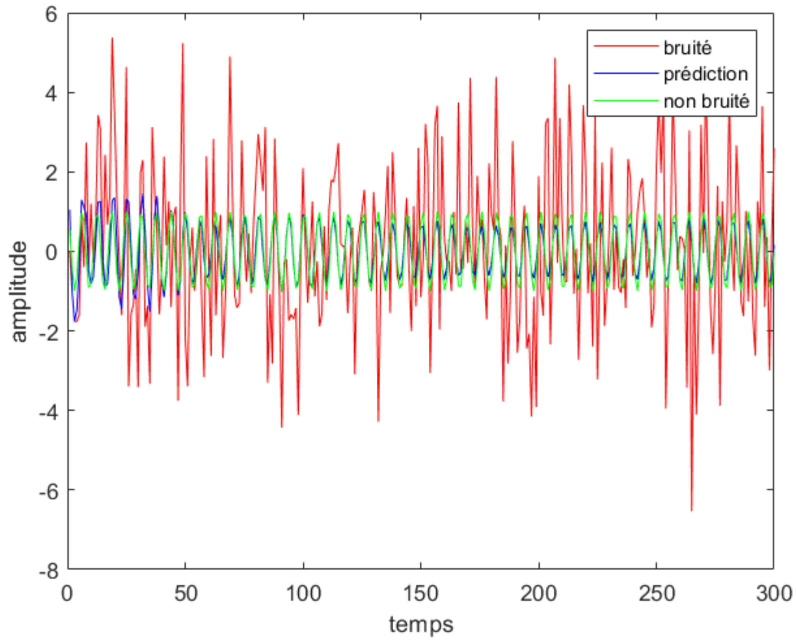

FIGURE 4 – Courbe $T_{final} = 300$

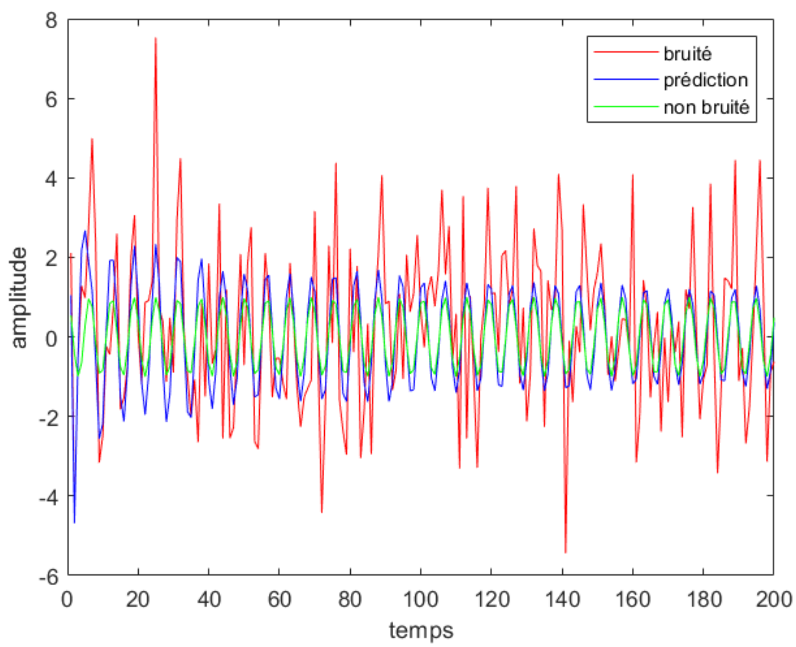

FIGURE 5 – Courbe $T_{final} = 200$

On s'aperçoit que plus le T_{final} est élevé plus la prédiction du signal est juste pour un T faible. On peut voir la courbe bruité en rouge, qui semble aléatoire. Ainsi que notre courbe de prédiction en bleu qui tend à suivre notre signal d'origine en vert.

9 Conclusion

Nous avons modélisé un signal sinusoïdal. Puis nous l'avons bruité de manière aléatoire pour voir si on pouvait le retrouver grâce au filtre de Kalman. Ensuite, nous avons codé sur Matlab le filtre de Kalman pour retrouver notre signal de départ. Nous avons pu voir que le filtre de Kalman nous permettait de retrouver un signal lisible à partir d'un signal complètement bruité.