TP2

Filtres de Wiener

1 Introduction

Le but de ce TP est d'apprendre à

- Simuler des signaux correspondant à un système donné (décrit physiquement ou caractérisé par sa fonction de transfert)
- Calculer et implémenter un filtre de Wiener qui estime un paramètre.

Pour cela, on commence par décrire et résoudre un problème-type. Cette partie est très commentée dans l'énoncé. Le but est de reproduire le système décrit en comprenant ce que vous faites, afin de pouvoir le refaire ultérieurement.

Dans tout le TP, les calculs mathématiques expliquée en cours et faits en TD sont donnés.

La partie essentielle du TP consiste à étudier deux autres cas de calcul et d'implémentation de filtre de Wiener. Ces deux cas sont plus simples à implémenter que le premier cas.

Essayer de répondre aux questions avec un peu d'inspiration. En particulier, chercher à faire des commentaires plus pertinents que "on voit que ça marche", même si c'est parfois difficile.

2 Déconvolution d'un canal de transmission numérique

Ce problème, que l'on reproduit ci-dessous, a été étudié en TD (exercice 3) :

> On considère un processus de la forme $X_n = b_n + \alpha b_{n-1} + W_n$ où W_n est un processus gaussien de moyenne 0 et de variance σ^2 tel que W_n et W_j sont indépendants si $n \neq j$. On suppose que b_n est une variable aléatoire uniforme à valeurs dans $\{-1, 1\}$, indépendante de $(W_n)_{n\in\mathbb{Z}}$ et que de même, b_n et b_j sont indépendants si $n \neq j$ ($\mathbb{P}(b_n = 1) = \mathbb{P}(b_n = -1) = \frac{1}{2}$). Construire un filtre qui estime b_{n-d}.

Application numérique : $\alpha = 0,5$, $\sigma = 0,2$, on estimera le bit en entrée avec un retard $d = 4$ et on cherchera un filtre de Wiener d'ordre $p = 5$.

On va commencer par écrire un bloc qui génère des bits valant -1 ou 1 avec probabilité $\frac{1}{2}$. Pour cela, on crée un schéma simulink de la Figure 1. Dans le bloc Uniform Random Number, on a fixé la valeur Minimum à 0, le valeur Maximum à 2, et la valeur Sample time à 1. que l'on transforme en sous-système, Figure 2, en sélectionnant les blocs en jaune puis par clique droit, option create subsystem.

On accède à nouveau aux blocs internes en double-cliquant, Figure 3.

Question 1 *Expliquer comment fonctionne ce générateur de nombre aléatoires.*

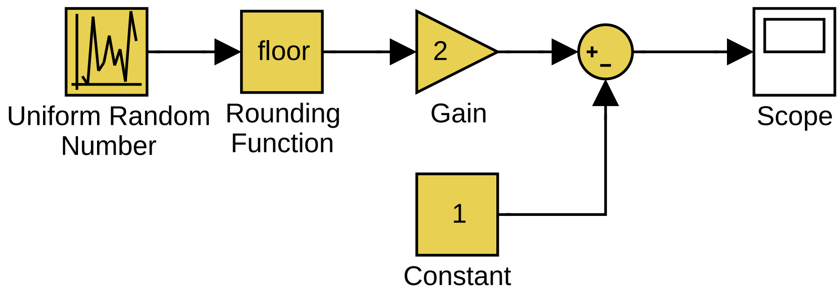


FIGURE 1 – Générateur aléatoire de bits (-1 et 1)

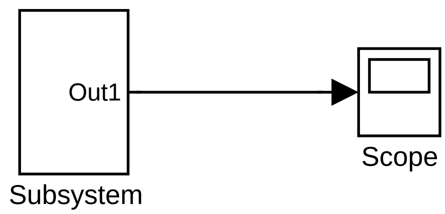


FIGURE 2 – Générateur aléatoire après création d'un sous-système

On incorpore ensuite ce sous-système (Bit generator, en jaune) dans un schéma qui réalise le filtre de Wiener :

Dans ce schéma, on a défini "alpha" et "sigma" comme étant des constantes (Figure 5, par exemple).

Ces dernières devront être fixées dans Matlab. Sinon, vous aurez des erreurs de type :

```
Error evaluating parameter 'Gain' in 'yourfile/alpha': Error using ==> alpha
```

Il en est de même du retard dans le bloc "Integer Delay" et du polynôme $a\,(z)$ dans le bloc "Discrete Filter", qui est par exemple défini par :

Question 2 *Pourquoi a-t-on placé un bloc "Integer delay" ?*

Question 3 *Expliquez ce que fait le bloc "Discrete Filter". En particulier, comment faut-il interpréter les variables "Numerator" et "Denominator" ?*

Il faut donc fixer "alpha", "a" et les autres. pour cela, on a écrit un script, sauvegardé dans le fichier "mk_wiener.m" et disponible sur le site du TP :

```
alpha=0.5;
sigma=0.2;
p=5;
d=4;
d0=(1+alpha^2+sigma^2)*ones(p,1);
d1=alpha*ones(p-1,1);
```

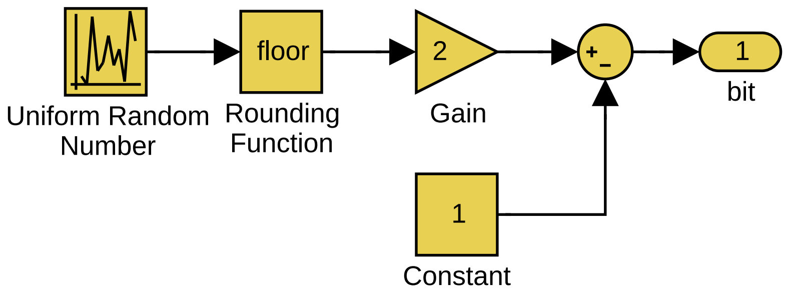


FIGURE 3 – Sous-système "générateur de bits"

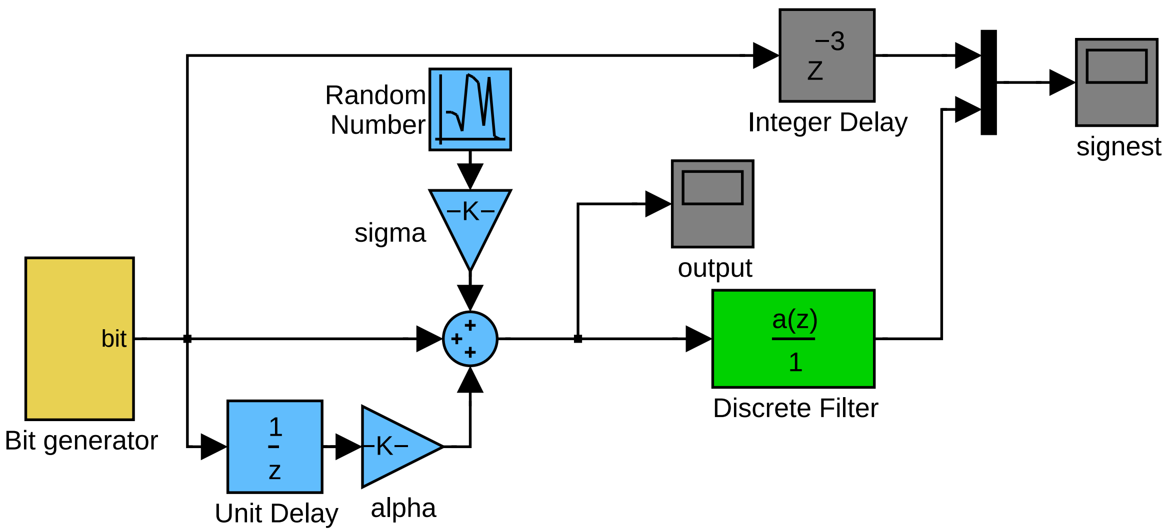


FIGURE 4 – Modèle Simulink final avec le générateur et le filtre de Wiener. Noter la présence du bloc retard.

```
G=diag(d0)+diag(d1,1)+diag(d1,-1);
g=zeros(p,1);
if d-1<=p,g(d-1)=alpha;end,
if d<=p,g(d)=1;end,
a=(G\g)';
```

Les quatre premières lignes de ce fichier sont directement issues de l'énoncé. Ensuite, on défini **d0**, **d1**, puis surtout **G** comme étant une matrice tridiagonale (faire help diag) qui est la matrice du système de Wiener-Hopf intervenant dans la solution du problème : c'est la matrice de covariance

$$G = \begin{pmatrix} \Gamma_X(0) & \Gamma_X(1) & \cdots & \Gamma_X(p) \\ \Gamma_X(1) & \Gamma_X(0) & & \Gamma_X(p-1) \\ \vdots & & \ddots & \\ \Gamma_X(p) & \Gamma_X(p-1) & & \Gamma_X(0) \end{pmatrix}$$

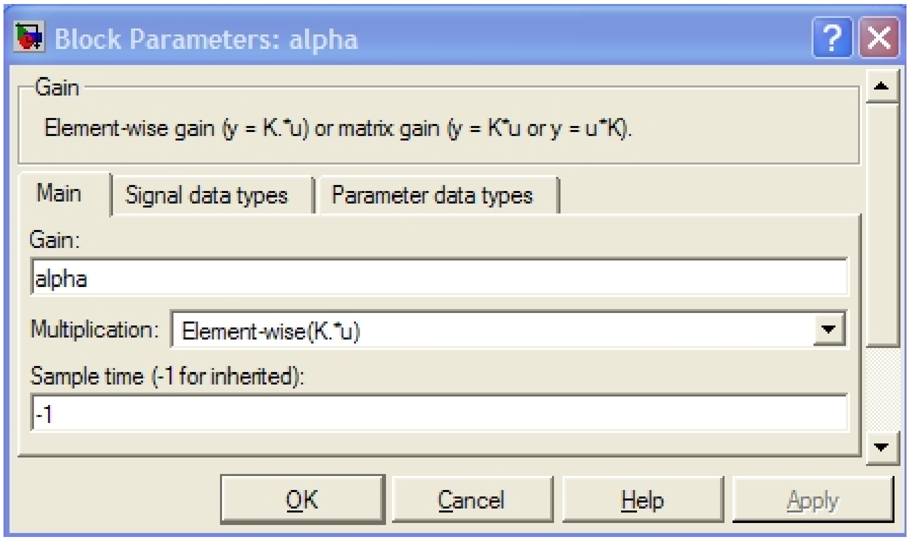


FIGURE 5 – Bloc gain "alpha"

où dans ce cas, on a établit en TD que

$$\Gamma_X(k) = \begin{cases} 1+\alpha^2+\sigma^2 \text{ si } k=0 \\ \alpha \text{ si } k=\pm 1 \\ 0 \text{ sinon} \end{cases}$$

alors que **g** est le vecteur

$$g = \begin{pmatrix} \Gamma_{Xb}(0) \\ \Gamma_{Xb}(1) \\ \vdots \\ \Gamma_{Xb}(p) \end{pmatrix}$$

où

$$\Gamma_{Xb}(k) = \begin{cases} \alpha \text{ si } k=d-1 \\ 1 \text{ si } k=d \\ 0 \text{ sinon} \end{cases}$$

puis la dernière ligne du fichier est la résolution numérique des équations de Yule-Walker.

Question 4 *Consultez* ***G***, ***g*** *et* ***a*** *dans Matlab. Quelles sont leurs valeurs ? Quel est le filtre de Wiener obtenu ?*

Après avoir ainsi initialisé les variables, on peut lancer la simulation, et on obtient le résultat suivant, où l'on voit que l'estimation (en rouge) de la suites de bits (en vert) sera parfaite, quand on aura arrondi à ± 1 (100% de bonnes estimations en régime permanent !).

Question 5 *Quel bloc peut-on mettre pour que les valeurs prédites soient toujours -1 ou +1, en fonction de la valeur estimée par le filtre de Wiener ? Implémenter et vérifier que les courbes (bit réel et son estimation) sont presque superposées. Que se passe-t-il au début de la simulation ?*

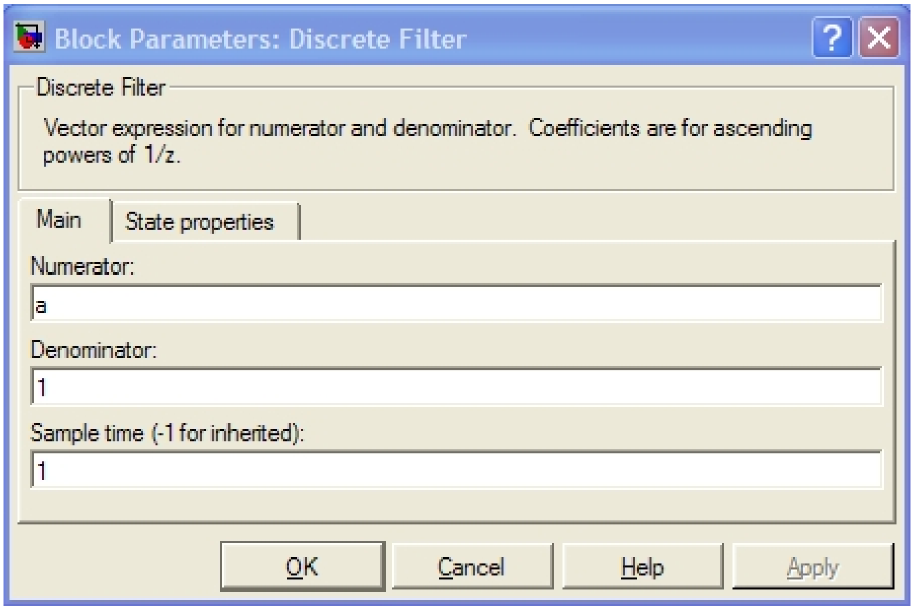


FIGURE 6 – Le bloc "Discrete filter", au coeur de ce cours de traitement numérique du signal

Question 6 *Modifiez la longueur du filtre de Wiener (p). Quelle serait le réglage optimal du filtre, en prenant en compte la performance et le retard d'estimation ("time lag"), qui est proportionnel à la longueur du filtre, c'est à dire à p.*

3 Estimation glissante d'une moyenne

Reprenons le premier exercice de la fiche de TD 1 :

> On considère un processus de la forme $X_n = \theta + W_n$ où W_n est un processus gaussien de moyenne 0 et de variance τ^2 tel que W_n et W_j sont indépendants si $n \neq j$. On suppose que θ suit une loi $\mathcal{N}(0, \sigma^2)$ indépendante de $(W_n)_{n \in \mathbb{Z}}$.
> Caractériser le filtre de Wiener permettant d'estimer θ à partir de X_n, X_{n-1}, ..., X_{n-p+1}. On demande de calculer les coefficients *explicitement*.

Application numérique. $\sigma = \tau = 1$.

Le paragraphe 3.2 donne la solution théorique à cette exercice, qui permettra d'en déduire le filtre de Wiener.

3.1 Réalisation d'une simulation du système

Réaliser un schéma simulink qui permet de simuler ce problème. On notera que θ est un paramètre aléatoire qui ne dépend pas du temps. Outre un boc "sum", le bloc principal à utliser est le bloc "Random number" avec des paramètres correctement choisis,en particulier le paramètre "Sample time", adapté pour θ et.pour W_n.

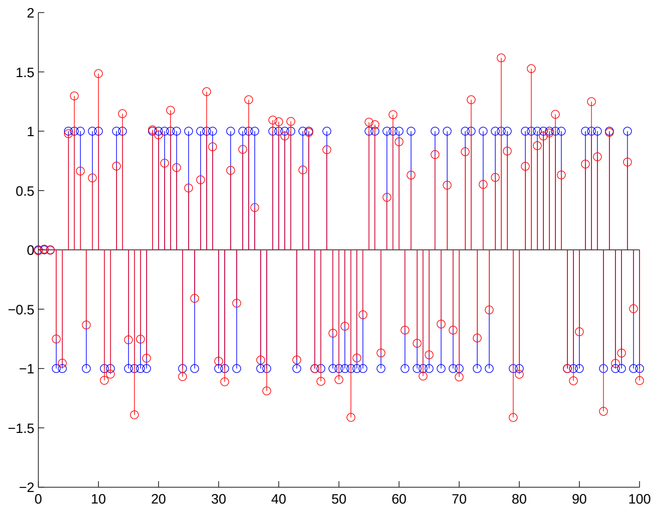


FIGURE 7 – Les valeurs réelles en bleu et leur estimation brute en rouge : 100% de bonnes estimations après un court délai initial

3.2 Synthèse théorique du filtre de Wiener

Reprenons ici le calcul fait en TD. On a d'ailleurs procédé comme indiqué dans le cours :

$$\begin{aligned}
\Gamma_X(0) &= E[X_n X_n] \\
&= E[(\theta + W_n)(\theta + W_n)] \\
&= E\left[\theta^2\right] + 2\underbrace{E[\theta W_n]}_{=0} + E\left[W_n^2\right] = \sigma^2 + 1
\end{aligned}$$

puis

$$\begin{aligned}
\Gamma_X(1) &= E[X_n X_{n-1}] \\
&= E[(\theta + W_n)(\theta + W_{n-1})] \\
&= E\left[\theta^2\right] + \underbrace{E[\theta W_n] + E[\theta W_{n-1}] + E[W_n W_{n-1}]}_{=0} = \sigma^2
\end{aligned}$$

et ainsi de suite, $\Gamma_X(k) = \sigma^2$, $k \geq 1$. Ensuite, on calcule les covariances entre θ et les X_{n-j} :

$$\begin{aligned}
\Gamma_{\theta X}(j) &= E[\theta X_{n-j}] \\
&= E[\theta(\theta + W_{n-j})] \\
&= E\left[\theta^2\right] + \underbrace{E[\theta W_{n-j}]}_{=0} = \sigma^2
\end{aligned}$$

Finallement, on obtient le système matriciel

$$\begin{pmatrix} \tau^2 + \sigma^2 & \tau^2 & \cdots & \tau^2 \\ \tau^2 & \tau^2 + \sigma^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \tau^2 \\ \tau^2 & \cdots & \tau^2 & \tau^2 + \sigma^2 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{p-1} \end{pmatrix} = \begin{pmatrix} \tau^2 \\ \tau^2 \\ \vdots \\ \tau^2 \end{pmatrix}$$

soit encore

$$\begin{aligned}
\left(\tau^2+\sigma^2\right)a_0+\tau^2 a_1+\cdots+\tau^2 a_{p-1} &= \tau^2\\
\tau^2 a_0+\left(\tau^2+\sigma^2\right)a_1+\cdots+\tau^2 a_{p-1} &= \tau^2\\
&\vdots\\
\tau^2 a_0+\tau^2 a_1+\cdots+\left(\tau^2+\sigma^2\right)a_{p-1} &= \tau^2
\end{aligned}$$

On fait la somme des p équations terme à terme ce qui donne

$$\left(p\tau^2+\sigma^2\right)\sum_{i=0}^{p-1} a_i = p\tau^2$$

et par symétrie, pour tout i de 0 à $p-1$

$$a_i = \frac{\tau^2}{p\tau^2+\sigma^2}$$

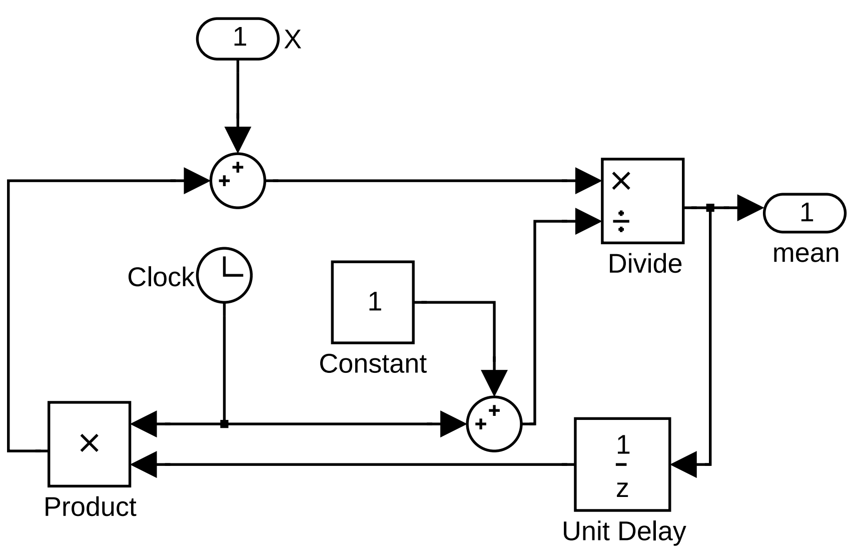
et donc le filtre de Wiener s'écrit

$$\widehat{\theta} = \frac{1}{p+\frac{\sigma^2}{\tau^2}}\sum_{i=0}^{p-1} X_{n-i}$$

L'information *a priori* est présente dans le terme $\frac{\sigma^2}{\tau^2}$ puisque l'estimateur sans connaissance a priori serait naturellement $\widehat{\theta} = \frac{1}{p}\sum_{i=0}^{p-1} X_{n-i}$.

3.3 Simulation d'un estimateur glissant de moyenne

Compléter le simulateur écrit dans la section 3.1 par l'estimateur calculé ici, à l'aide du bloc "Discrete filter" dont les paramètres sont ici beaucoup plus simples que dans le premier exercice (on pourrait d'ailleurs envisager l'utilisation d'autres blocs). Tester et discuter de la performance de ce système.



3.4 Kesako

Question 7 ***(difficile)*** *Si on fait entrer X_n dans ce système, qu'en sort-il ?*

4 Extraction d'un signal inconnu dans un canal bruité

4.1 Le problème et sa solution

Il s'agit du second exercice de la fiche de TD 1 :

> On considère un processus de la forme $X_n = S_n + W_n$ où W_n est un processus gaussien de moyenne 0 et de variance μ^2 tel que W_n et W_j sont indépendants si $n \neq j$. On suppose que S_n suit une loi $\mathcal{N}(0, \sigma^2)$ indépendante de $(W_n)_{n\in\mathbb{Z}}$ et que de même, S_n et S_j sont indépendants si $n \neq j$.

De même que dans le problème précédent, on reproduit succintement la solution théorique :

$$\Gamma_X(0) = E[X_n X_n] = E\left[S_n^2\right] + 2\underbrace{E[S_n W_n]}_{=0} + E\left[W_n^2\right] = \sigma^2 + 1$$

puis

$$\Gamma_X(1) = E[X_n X_{n-1}] = E[S_n S_{n-1}] + E[S_{n-1} W_n] + E[S_n W_{n-1}] + E[W_n W_{n-1}] = 0$$

et ainsi de suite, $\Gamma_X(k) = 0$, $k \geq 1$. Ensuite, on calcule les covariances entre θ et les X_{n-j} :

$$\Gamma_{SX}(j) = E[S_n X_{n-j}] = E[S_n S_{n-j}] + \underbrace{E[S_n W_{n-j}]}_{=0} = \sigma^2 \text{ si } j = 0 \text{ et } 0 \text{ sinon}$$

Finallement, on obtient un système matriciel qui se réduit à

$$\begin{aligned}
\left(\tau^2 + \sigma^2\right) a_0 &= \sigma^2 \\
\left(\tau^2 + \sigma^2\right) a_1 &= 0 \\
&\vdots \\
\left(\tau^2 + \sigma^2\right) a_{p-1} &= 0
\end{aligned}$$

ce qui donne $a_0 = \frac{\sigma^2}{\tau^2+\sigma^2}$ et $a_j = 0$ pour $j \geq 1$ et donc le filtre de Wiener s'écrit $\hat{S}_n = \frac{\sigma^2}{\tau^2+\sigma^2} X_x$

4.2 Simulation

Implémenter en Simulink ce problème et la solution obtenue. Commentez les résultats.

Question 8 *Etes vous surpris par les résultats : sont ils meilleurs ou moins bons que ce que vous pensiez ?*