

Traitement des Signaux Aléatoires

Détection Quadratique

4 ETI – CPE Lyon

Travaux Pratiques TSA

2020-2021

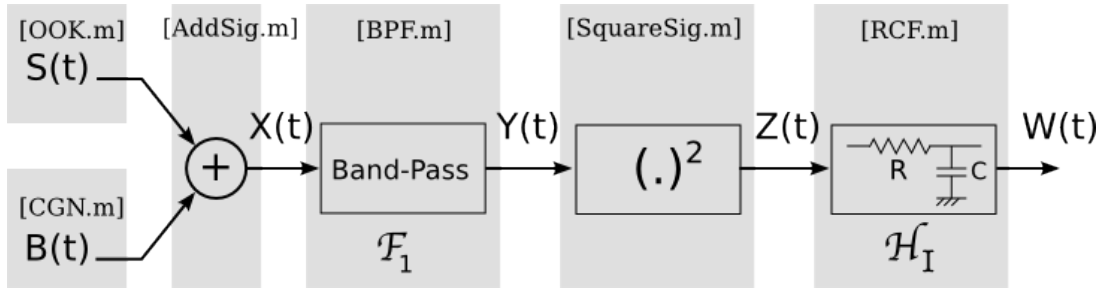
Noms, Prénoms : BRUYERE Axel CHARRAT Philippe

Groupe : D

Date : 20/10/2020

Contexte et Objectif

On souhaite étudier expérimentalement la chaîne de détection quadratique suivante :



On souhaite détecter la présence ou non d'un signal aléatoire $S(t)$ dans un mélange signal + bruit. Le signal $X(t)$ reçu est égal à :

$$X(t) = S(t) + B(t)$$

Le signal $S(t)$ est un signal sinusoïdal de fréquence ν_0 , d'amplitude A_0 , à phase équipartie sur $[0, 2\pi[$ et modulé par un signal binaire $M(t) = 0$ ou 1 :

$$S(t) = M(t) \cdot A_0 \cos(2\pi\nu_0 t + \phi)$$

Ce signal est bruité lors de la transmission par un bruit $B(t)$ gaussien, centré, stationnaire d'ordre 2 et de largeur de bande B centrée sur ν_0 (on supposera que le bruit est blanc sur le support fréquentiel du filtre \mathcal{F}_1).

$$X(t) = \begin{cases} B(t) & \text{si } M(t) = 0; \\ A_0 \cos(2\pi\nu_0 t + \phi) + B(t) & \text{si } M(t) = 1 \end{cases}$$

L'objectif de la chaîne de détection quadratique est de détecter dans l'observation reçue $X(t)$, la présence ($\mathbf{M}(\mathbf{t}) = \mathbf{1}$) ou l'absence ($\mathbf{M}(\mathbf{t}) = \mathbf{0}$) du signal utile $S(t)$.

Etudier soigneusement le TD corrigé qui vous a été remis et qui détaille le calcul des rapports signal sur bruit (SNR) aux différents étages de la chaîne de détection. Répondre aux questions de préparation qui suivent.

Manipulation

Dans l'ensemble du TP :

- tous les signaux sont échantillonnés à la fréquence $F_s = 500$ Hz.
- la bande passante du bruit $B(t)$ est fixée à $B = 160$ Hz.
- la fréquence du signal sinusoïdal $S(t)$ est fixée à $\nu_0 = 100$ Hz
- l'ordre du filtre passe-bande \mathcal{F}_1 (butterworth) est fixé à $\text{ordre} = 6$

Vous veillerez à mettre sur vos Figures des légendes et des labels explicites et informatifs.

1 Etude du bruit seul

Dans cette partie, $M(t) = 0$, $\forall t$, de sorte que le signal est toujours absent.

1.1 Synthèse du bruit $B(t)$

On considère un bruit $B(t)$ centré, de puissance moyenne $\overline{P_B} = 5 V^2$.

Avec les paramètres déterminés en préparation, reproduire dans le cadre ci-dessous, le code Matlab permettant :

- de générer une réalisation du bruit $B(t)$ sur une durée $T = 100$ s
Afficher la sortie de `CGN.m` dans la Figure 1 (**veillez à ajouter des légendes pertinentes**)
- de mesurer sur la trace de bruit ainsi obtenu les paramètres demandés à la Table 1.

Pour réaliser la séquence de bruit, nous avons utilisé la fonction `CGN` (disponible sur CPe-Campus). Cette fonction génère une structure en fonction 4 paramètres, un entier pour l'écart-type, un entier pour la fréquence échantillonnage, un entier pour la bande passante du bruit et un pour la durée du signal.

```
% -----  
% Nom : partie1.m  
% But : Étude du bruit  
% -----  
% --- initialisaitons des variables  
tmax = 100;  
fs = 500;  
v0 = 100;  
b = 160;  
delta_v = 16;  
% --- Création du bruit  
Xp = struct('sigma',sqrt(5),'Fs',fs,'B',b,'T',tmax) ;  
[X,Xp] = CGN(Xp);  
% --- Calcule des valeurs particulières  
moyennedeB = mean(X.data);  
variancedeB = (std(X.data))^2;  
P_B_X = trapz(X.time,(X.data).^2)/tmax;  
Gamma_X = P_B_X/(2*b);
```

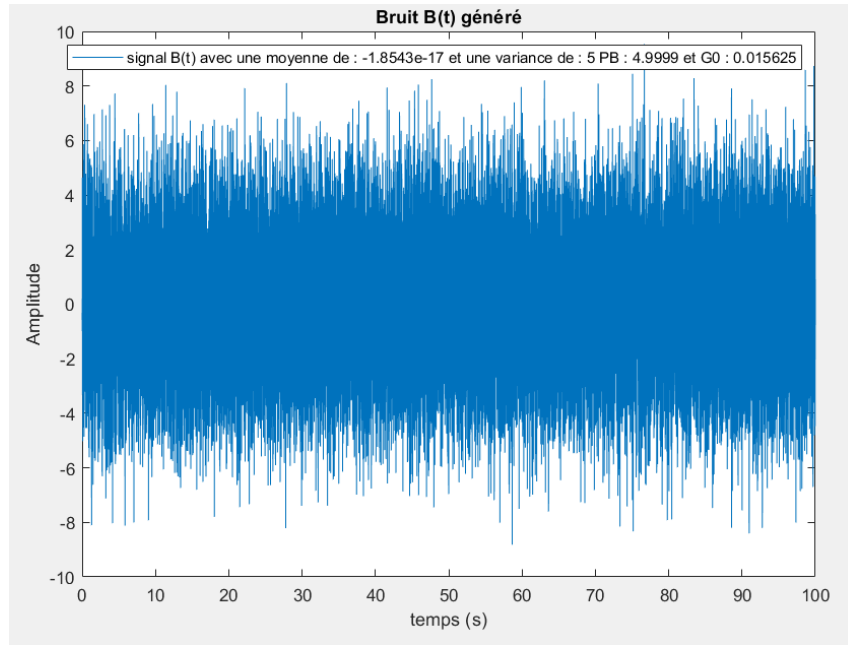


FIGURE 1 – Réalisation du bruit avec la légende contenant la moyenne, l'écart-type, Γ_0 empirique.

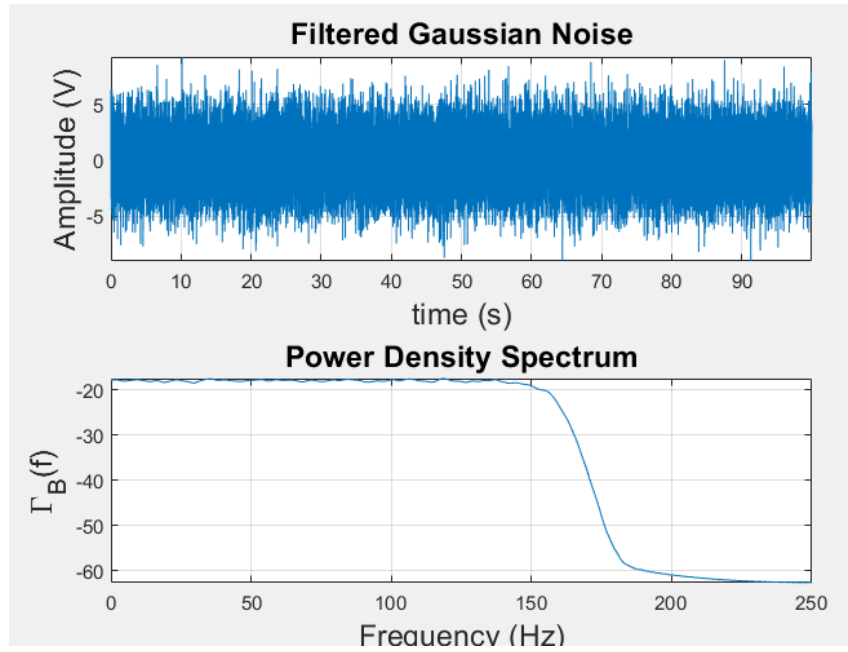


FIGURE 2 – Réalisation et densité spectrale de puissance moyenne du bruit : $B = 160 \text{ Hz}$, $P_B = 5 \text{ V}^2$, $\sigma_B = \sqrt{5}$, $\mu_B = 0$.

Moyenne $B(t)$	$-1.8543 * 10^{-17}$
Variance $B(t)$	5

TABLE 1 – Mesures de la moyenne et de la variance de $B(t)$.

A partir de la Figure 2 et en expliquant la démarche suivie, retrouver (approximativement) la valeur de Γ_0 . Comparer à la valeur théorique de la préparation.

Dans la préparation, nous avons établi que le Γ_0 théorique vaut -18dB. Pour retrouver la valeur Γ_0 empirique, nous avons appliqué deux méthodes qui sont :

- Méthode graphique : le Γ_0 correspond à la valeurs moyenne dans l'intervalle 0 à B (160 Hz). Sur la figure 2, nous observons que nous somme à un Γ_0 d'environ -18dB.
- Méthode de calcul empirique : nous pouvons calculer la Densité Spectrale de Puissance Moyenne (DSPM) à l'aide de la méthode des trapèzes et obtenir Γ_0 en divisant la DSPM par 2 fois la Bande passante. On obtient 0,01526, si l'on applique : $\Gamma_0 = 20 * \log(resultat)$ soit environ -18.063 dB.

Nous pouvons en déduire que les valeurs empiriques sont cohérentes avec les valeurs théorique.

1.2 Etude du filtre passe-bande \mathcal{F}_1

On filtre le bruit $B(t)$ par un filtre passe-bande, de bande passante $\Delta\nu$ centrée sur la fréquence F_0 .

1.2.1

On choisit $\Delta\nu = 16 \text{ Hz}$ et la valeur de F_0 identifiée dans la préparation.

Reproduire dans le cadre ci-dessous, le code permettant de :

- synthétiser le filtre \mathcal{F}_1 correspondant
- filtrer le bruit $B(t)$ par le filtre \mathcal{F}_1 (afficher avec des légendes pertinentes, la sortie du `BPF.m` dans la Figure 4)
- de mesurer sur la trace en sortie du filtre \mathcal{F}_1 les valeurs des paramètres demandés au Tableau 2 :

Pour réaliser le filtre, nous avons utilisé la fonction *BPF* (disponible sur CPe-Campus). Cette fonction génère une structure du signal filtré en fonction de 2 structures, la première qui est le bruit généré partie 1. La seconde partie est une structure contenant entier pour la fréquence d'échantillonnage, un entier pour la fréquence ν_0 , un entier pour le $\Delta\nu$ et un entier pour l'ordre du filtre.

```
% --- initialisations des variables
v0 = 100;
delta_v = 16;
% --- Création et filtrages s
Fp = struct('Fs',fs,'F0',v0,'Dnu',delta_v,'order',6,'class','BP filter') ;
Y = BPF(X,Fp) ;
% --- Calculs des variables utiles
moyennedeBF = mean(Y.data);
variancedeBF = (std(Y.data))^2;
P_B_Y = trapz(Y.time,(Y.data).^2)/tmax;
Gamma_Y = P_B_Y/(2*b);
```

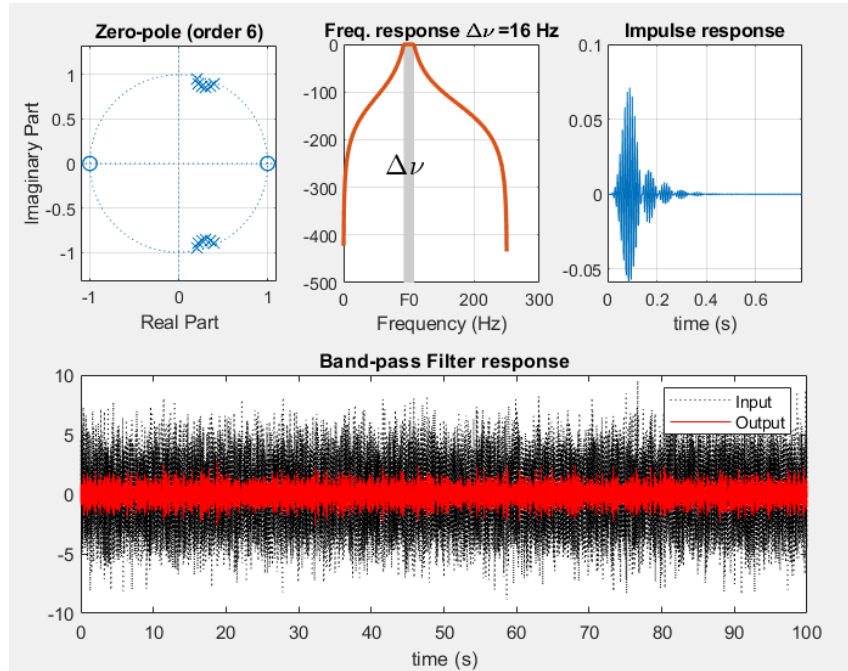


FIGURE 3 – Bruit $Y(t)$ filtré passe-bande pour $\Delta\nu = 16 \text{ Hz}$

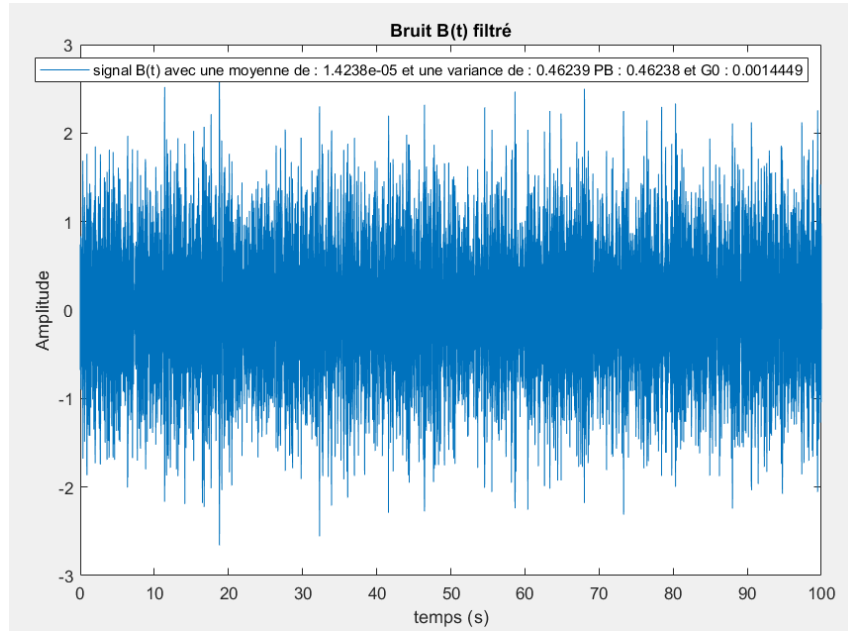


FIGURE 4 – Bruit $Y(t)$ filtré passe-bande pour $\Delta\nu = 16$ Hz

Moyenne $Y_B(t)$	0
Variance $Y_B(t)$	0.46239

TABLE 2 – Mesures de la moyenne et de la variance de $Y_B(t)$.

1.2.2

Estimer la valeur de Γ_0 . Comparer les mesures (\overline{P}_{Y_B} et Γ_0) aux valeurs théoriques obtenues en préparation. Comment peut on expliquer les éventuelles différence ?

Dans la préparation, nous avons établi que le PB théorique vaut 0.5, or la valeur PB empirique est 0.4623 donc on obtient une variation d'environ 0.04. Cette variation peut être expliquer par l'utilisation de réalisation aléatoire empirique qui entraîne une variation sur les paramètres. Le Γ_0 empirique est de 0.001449 or la valeur Γ_0 théorique vaut 0.015 donc le filtre entraîne aussi une variation sur la DSPM.

1.2.3

En pratique, qu'est ce qui limite le choix d'une bande passante $\Delta\nu$ trop étroite ?

A l'aide de la relation entre temps et fréquence, si nous prenons une bande passante très étroites alors la réponse impulsionnelle sera lente. Ainsi, il faut trouver un compromis en fonction du cahier des charges.

1.3 Elévation au carré et Filtrage RC passe-bas

1.3.1

Comme précédemment, on choisit $\Delta\nu = 16Hz$. En faisant varier le produit $\Delta\nu \times RC$ dans une boucle (du type `for ...end`), donner dans le cadre ci-dessous, le code qui :

- génère le signal $Z_B(t) = Y_B^2(t)$
- calcule la valeur de la constante RC correspondant au produit $\Delta\nu \times RC$ choisi
- filtre le signal $Z_B(t)$ par le filtre \mathcal{H}_I de constante de temps RC
- mesure sur la sortie $W_B(t)$ les paramètres demandés dans la Table 3

```
% --- initialisations des variables
RC = 0;
tab_RC_dV = [2 20 100];
tab_Wb = [];
tab_RCP = [];
% --- Passage du signal dans le Quadratureur
Z_b = SquareSig(Y);
% --- Boucles de traitements des 3 cas
for i=1:3
    figure(3+i)
    % --- Calcul de la valeur RC en fonction du produit
    RC_dV = tab_RC_dV(i);
    RC = RC_dV/delta_v;
    % --- initialisations de la structure en fonction
    RCFp = struct('Fs','fs','RC',RC);
    [Wb,RCFp] = RCF(Z_b,RCFp);
    tab_Wb = [tab_Wb; Wb];
    tab_RCP = [tab_RCP; RCFp];
end
```

Remplir le tableau de mesures de la Table 3 (ignorez dans un premier temps les mesures demandées *après correction*).

$\Delta\nu \times RC$	2	20	100
RC	1/8	5/4	6.25
moyenne $W_B(t)$	0.46185	0.4558	0.43242
variance $W_B(t)$	0.4528	0.0059138	0.0060582
Kurtosis $W_B(t)$	5.3958	6.6982	14.0553
moyenne $W_B(t)$ (après correction)	0.4628	0.4605	0.45574
variance $W_B(t)$ (après correction)	0.4628	0.0046886	0.0039945
Kurtosis $W_B(t)$ (après correction)	5.4117	2.9536	2.36328

TABLE 3 – Sortie Filtre RC - Cas du bruit seul.

1.3.2

Le processus $Z_B(t)$ (**signal en sortie du quadrateur**) est-il gaussien ? Pourquoi ?

Nous étudions un processus aléatoire qui est un bruit Gaussien centré généré à partir de la fonction *rand*. Ce processus est filtré par un filtre linéaire et passé dans le Quadrateur, or aucun des deux éléments altère la nature du signal. Donc on observe un processus Gaussien.

1.3.3

Pour les 2 valeurs extrêmes de $\Delta\nu \times RC$ proposées dans la Table 3, afficher dans la Figure ci-dessous, les sorties de RCF.m.

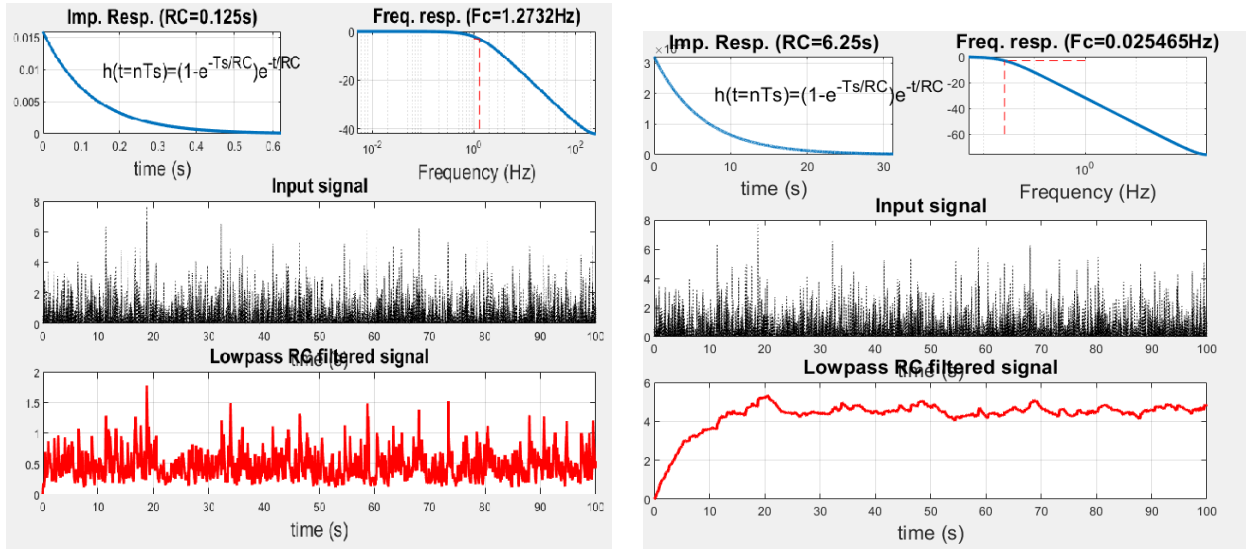


FIGURE 5 – Sortie $W_B(t)$ du filtre passe-bas pour $\Delta\nu = 16\text{ Hz}$ – (a) $RC = 1/8$. ($\Delta\nu \times RC = 2$) (b) $RC = 6.25$ ($\Delta\nu \times RC = 100$)

1.3.4

Comparer pour chaque valeur de la constante RC , la valeur moyenne mesurée à la valeur théorique déterminée dans la préparation. Qu'est ce qui peut expliquer ces différences ? Comment corriger cet effet ?

Nous possédons un régime transitoire sur le filtre RC qui impacte les premières valeurs et qui biaises les indicateurs tel que la moyenne ou le Kurtosis. Ce régime dure environ 5 τ , donc on doit attendre le régime permanent avant de mesurer les indicateurs. Cela se traduit en MATLAB par ignorer les n valeurs pour n correspondant 5 τ des vecteurs *data* et *time* de la structure *Z*.

1.3.5

Donner dans l'encadré ci-dessous les 2 lignes de code qui implémentent cette solution.

```
% --- Le premier indice > 5*tau
itrue = Wb.time >= 5*RCp.RC;
% --- Calcul des divers indicateurs
moyenne = mean(Wb.data(itrue));
variance = (std(Wb.data(itrue)))^2;
kurtosi = kurtosis(Wb.data(itrue));
```

Appliquer cette correction et porter les nouvelles mesures dans la Table 3 (partie *avec correction*).

1.3.6

Lorsque le Kurtosis est proche de 3, que peut on dire de la statistique du processus $W_B(t)$?

Quel théorème important ce résultat illustre-t-il ?

Pour quelles(s) valeur(s) de RC a-t-on une *intégration forte* ? Comparer les variances de $W_B(t)$ mesurées pour les deux valeurs extrêmes de RC .

Ces résultats illustrent le théorème Centrale Limite. Dans le cas d'un processus Gaussien, la valeur du Kurtosis est à 3. Pour étudier la valeur du Kurtosis rigoureusement, il faudrait réaliser un grand nombre fois le processus aléatoire et étudier la moyenne des valeurs obtenues car la valeur Kurtosis d'un processus aléatoire varie à chaque réalisation .

L'intégration forte est l'approximation que les éléments portes et triangles autour de $\delta = 0$ sont très large et constant devant le module carrée de $H(v)$. Pour assurer cette hypothèse il faut que $1/RC \ll \Delta_V$.

Dans notre cas, une simple étude d'une valeur suffit à déduire que les processus sont Gaussiens pour les valeurs 5/4 et 6.25 de RC , nous pouvons en déduire que plus le résultat $\Delta_v * RC$ est important et plus le processus sera proche d'un processus Gaussien.

Dans la suite du TP, il faudra systématiquement appliquer cette correction aux mesures effectuées en sortie du filtre RC.

2 Mélange Signal + Bruit

On étudie à présent le signal $W(t)$ en sortie du filtre RC passe-bas, lorsque le mélange $X(t) = S(t) + B(t)$ est reçu en entrée du détecteur.

2.1 Sortie du filtre passe-bande \mathcal{F}_1

2.1.1

En utilisant les paramètres déterminés en préparation, générer une réalisation du signal $S(t)$ sur la même durée $T = 100$ s et la même fréquence d'échantillonnage $F_s = 500$ Hz.

Reporter le code correspondant ci-dessous.

```
% Nom : partie2.m
% But : Étude d'un signal mélangé avec du bruit et passé dans une chaîne de traitement
%Initialisation des variables
nu0 = 100;%fréquence du signal sinusoïdal
AO = 1;%amplitude du signal sinusoïdal
b = 160;%bande passante
T = 100;%temps d'observation
Fs = 500;%Fréquence d'echantillonnage
Dnu = 16;%bande passante centrée
DnuRC = 20 ;%Choix du paramètre Dnu * RC
RC = DnuRC/Dnu;%Paramètre RC
Sp = struct('Fs',Fs,'A',AO,'Fc',nu0,'FM',0,'Phi',0,'T',T,'W',[]);
[S,Sp,M] = OOK(Sp); %Signal sinusoïdal
Xp = struct('sigma',sqrt(5),'Fs',Fs,'B',b,'T',T) ;
[B,Xp] = CGN(Xp);%Bruit
X = AddSig(B,S);%Add du bruit

Fp = struct('Fs',500,'F0',nu0,'Dnu',Dnu,'order',6,'class','BP filter') ;
[Y,Fp] = BPF(X,Fp);%Filtre passe-bande
[Ybruit,Fpbruit] = BPF(B,Fp);%Filtre passe-bande

Z = SquareSig(Y);%Elevation au carré
Zbruit = SquareSig(Ybruit);%Elevation au carré
RCFp = struct('Fs',Fs,'RC',RC);
[W,RCFp] = RCF(Z,RCFp);%Filtrage passe-bas
[Wbruit,RCFp] = RCF(Zbruit,RCFp);%Filtrage passe-bas
```

2.1.2

Vérifier que le filtre passe-bande, s'il est accordé sur la fréquence ν_0 n'altère pas le signal $S(t)$, en mesurant en sortie de \mathcal{F}_1 (dans le cas où $S(t)$ se présente seul en entrée) les paramètres demandés à la Table 4. En reprenant les mesures effectuées au paragraphe 1.2, déterminer le rapport signal sur bruit η_{E_1} en sortie du sortie du filtre \mathcal{F}_1 ainsi que le gain η_{E_1}/η_E introduit par \mathcal{F}_1 .

Fréquence $Y_S(t)$	100
Amplitude $Y_S(t)$	0.951
Puissance $Y_S(t)$	0.49999
Puissance $Y_B(t)$ (recopie Table 2)	0.4766
SNR η_{E_1}	10.490
Gain η_{E_1}/η_E	10

TABLE 4 – Mesures des SNR et gains en sortie de \mathcal{F}_1 .

2.1.3

Comparer aux valeurs théoriques.

Nous avons 0.5 comme valeur théorique de puissance de notre signal, ainsi qu'un gain de 10.

Nous trouvons ici une puissance de 0.4999 (0.4766 en sortie de filtre donc signal très peu altéré) ainsi qu'un gain de 10. Les valeurs empiriques sont donc concordantes avec les valeurs théoriques.

2.2 Sortie du filtre RC passe-bas

2.2.1

Dans les mêmes conditions expérimentales ($\overline{P_B} = 5 \text{ V}^2$, $\Delta\nu = 16\text{Hz}$, $\eta_E = -10 \text{ dB}$), effectuer les différentes mesures demandées dans le tableau 5.

	$\Delta\nu \times RC$	2	20	100
	RC	0.125	1.25	6.25
T	S_S	0.5	0.5	0.5
H	$B_S = Std\{W_{S+B}\}$	0.43	0.137	0.061
É	SNR η_S	1.15	3.65	8.17
O	Gain g_1	1.11	3.65	8.17
.	Gain g	11.55	36.52	81.65
M	moyenne W_B (recopie de Table 3)	0.46185	0.4558	0.43242
E	moyenne W_{S+B}	0.9620	0.9305	0.9690
S	S_S	0.5002	0.5098	0.5119
U	$B_S = Std\{W_{S+B}\}$	0.4035	0.1219	0.0523
R	SNR η_S	1.218	3.501	11.883
E	Gain g_1	1.204	3.320	12.186
S	Gain g	12.04	33.20	121.86

TABLE 5 – Sortie Filtre RC - Cas du mélange signal + bruit.

2.2.2

Représentez dans la Figure 6, la sortie de `RCF.m` correspondant au cas $\Delta\nu \times RC = 20$.

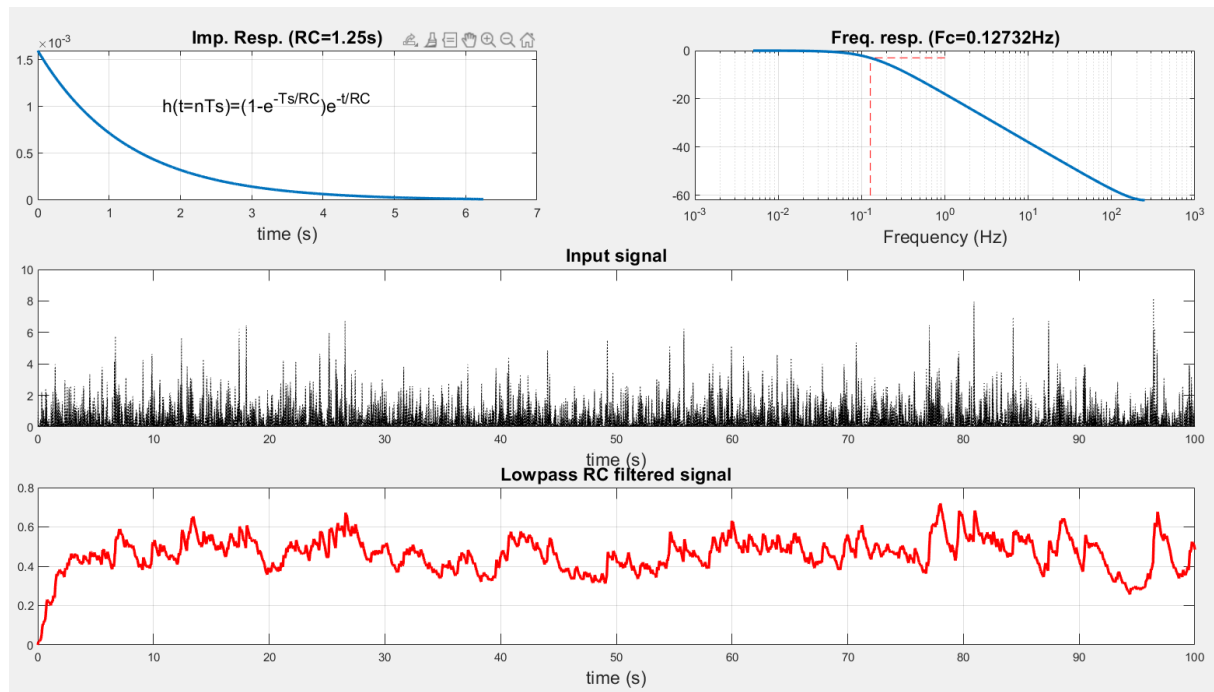


FIGURE 6 – Signal $W_{S+B}(t)$ dans le cas du mélange signal + bruit ($\Delta\nu = 16$ Hz, $\Delta\nu \times RC = 20$)

3 Transmission d'un message binaire

3.1 Modulation binaire périodique

On souhaite à présent transmettre et détecter une séquence périodique binaire.

3.1.1

Avec les paramètres suivant :

- Puissance du bruit $B(t)$, $\overline{P}_B = 5 \text{ V}^2$
- Rapport signal sur bruit en entrée de la chaîne, $\eta_E = -10 \text{ dB}$
- Fréquence du signal modulant $M(t)$, $F_M = 0.05 \text{ Hz}$
- Durée des signaux, $T = 100 \text{ s}$

synthétiser les signaux $S(t)$, $B(t)$ et $X(t)$ correspondant.

En vous basant sur les résultats expérimentaux obtenus dans la partie 2, choisissez un jeu de paramètres pertinent pour calibrer les filtres \mathcal{F}_1 et \mathcal{H}_I . Reporter dans le cadre ci-dessous le code de détection du signal binaire reçu.

```
% -----
% Nom : partie3.m
% But : Étude d'un signal mélangé avec du bruit
% -----
% --- Initialisation des variables
nu0 = 100;
b = 160;
AO = 1;
T = 100;
Fs = 500;
delta_v = 16;

% --- initialisation du filtre passe-bas
Fp = struct('Fs',500,'F0',nu0,'Dnu',16,'order',6,'class','BP filter') ;
% --- Initialisation du bruit ----
Xp = struct('sigma',sqrt(5),'Fs',Fs,'B',b,'T',T) ;
[X,Xp] = CGN(Xp);
% --- Initialisation de la séquence ON-OFF
Sp = struct('Fs',Fs,'A',AO,'Fc',nu0,'FM',0.05,'Phi',0,'T',T,'W',[]);
[S,Sp,M] = OOK(Sp);

% --- Affichage des puissances des signaux d'origine
Zs = SquareSig(S);
moyenneSignalOrigin = mean(Zs.data)
Xs = SquareSig(X);
moyenneBruitOrigin = mean(Xs.data)

% --- Création de la séquence avec du bruit ---
Signalbruite = AddSig(X,S);

% --- Filtrage du signal bruité avec le filtre ---
Y = BPF(Signalbruite,Fp);
Z = SquareSig(Y);

RC = 100/delta_v
RCFp = struct('Fs',Fs,'RC',RC);
[Wb,RCFp] = RCF(Z,RCFp);
```

```

% --- Calcul de la moyenne, l'écart-type et le Kutosis sans le régime transitoire
itrue = Wb.time >= 5*RCFp.RC;
moyenne = mean(Wb.data(itrue));
variance = (std(Wb.data(itrue))).^2;
kurtosi = kurtosis(Wb.data(itrue));
Z_wb = SquareSig(Wb);

% --- Création du signal final ---
nombre_indice = length(Wb.data);
signal_final = zeros(1,nombre_indice);

% --- Parcours du signal en sortit du filtre
for i =1:nombre_indice
    % Si la valeur i du signal est supérieur au seuil alors on l'ajoute dans le nouveaux signal.
    if Wb.data(i) > 0.71
        signal_final(i) = 1;
    end
end

% --- Partie Affichage tronquée ---

```

3.1.2

Visualiser dans la Figure 7 (en organisant avec la commande `subplot(4,1,.)` et en ajoutant une légende pertinente), les signaux :

- $S(t)$
- $X(t)$
- $W(t)$
- Le signal binaire détecté obtenu par seuillage du signal $W(t)$ (commenter le choix du seuil Σ choisi)

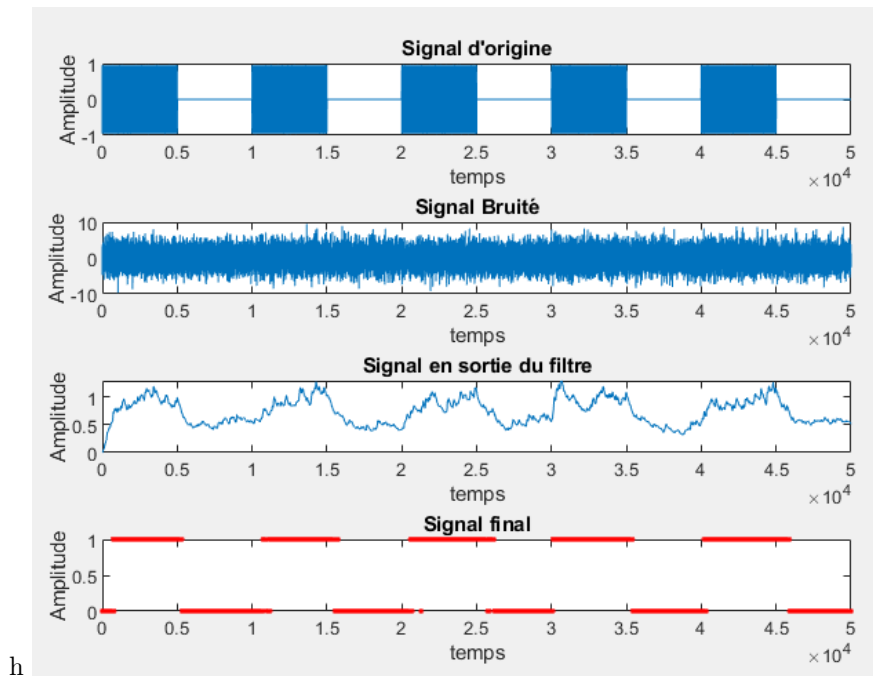


FIGURE 7 – (a) Signal binaire $S(t)$. (b) Mélange Signal (binaire) + bruit avant et après filtrage passe-bande. (c) Sortie $W(t)$ de la chaîne de détection quadratique. (d) Signal binaire détecté après seuillage de la sortie quadratique.

3.1.3

Indiquez les valeurs des paramètres de détection utilisés.

- La fréquence de Coupure F_c est à 100 Hz.
- La partie 2, nous a permis de fixer le filtre RC à 1.25.
- Le seuil a été fixé à 0.71 de manière empirique avec comme donné que l'amplitude du signal était 1.

3.1.4

Essentiellement quel élément de la chaîne de détection va-t-il limiter le débit de transmission ?

Il s'agit du filtre qui va limiter le débit de transmission. Si le filtre n'est pas adapté, il y aura l'apparition de 1 ou 0 non cohérent. Alors, en sortie nous aurons un signal qui ne sera pas régulier et plus cohérent.

3.1.5

Sans chercher à les estimer ici, quel(s) critère(s) permettrai(en)t de mesurer la qualité de la détection ?
Pour estimer

- Pour l'estimation, il faudrait faire la moyenne de chaque composantes et les sommer pour vérifier si nous obtenons une somme proche de la moyenne théorique.
- Pour la variance, il faudrait retirer la condition $signal(i) = 1$, pour ajouter $signal(i) = wb(i)$. Ainsi, nous pourrions étudier la dispersion de l'amplitude.
- La capacité à observer des pics proches en fréquence.

3.2 Décodage d'un message inconnu

Charger le signal reçu 'SignalRecu_j', où j est le numéro de votre binôme.

```
>> load SignalRecu_1
```

Le signal $X(t)$ correspond à un message codé (code ascii 7 bits) transmis par modulation d'amplitude et dégradé par un bruit additif lié au canal de transmission. Exécuter la commande :

```
>> [TxMsg,Xp] = RxMessage_DQ(X,Xp) ;
```

pour lancer une détection quadratique *automatique* sur le signal reçu X (la structure Xp contient tous les paramètres de la transmission). Ajuster en ligne, les différents paramètres de la détection jusqu'à ce que le message décodé vous semble satisfaisant. Recopier ci-dessous, le message décodé.

Aucun numéro de binome ne nous a été distribué donc nous avons choisi de manière arbitraire le 21. Nous avons agis de manière empirique en testant de manière progressive les valeurs de Σ

```
Received message --->
New threshold value [1.27] ? = 1
Received message ---> 2@
New threshold value [1] ? = 0.75
Received message ---> El@Jl'Alassa@e&i'nDraduClfnaB
New threshold value [0.75] ? = 0.5
Received message ---> Demain,lamelassedeвиendrad}cognac
New threshold value [0.5] ? = 0.45
Received message ---> Demay~<la'melawsedeвиendza}'cognak
New threshold value [0.55] ? = 0.55
Received message ---> Demain,lamelassedeвиendraducognac
```

On a obtenue pour $\Sigma = 0.55$, la phrase Demain,lamelassedeвиendraducognac

4 Annexes

4.1 Evolution théorique du gain SNR $g_1 = \frac{\eta_s}{\eta_1}$ en fonction de η_1 et de $RC \times \Delta\nu$

4ETI-TP-DQ-SNRgain-eps-converted-to.pdf

4.2 Documentation routines Matlab

4.2.1 OOK.m

`[S,Sp,M] = OOK(Sp)` Generates a ON-OFF keyed modulated signal whose parameters are specified by the parameter structure Sp
`S(n) = M(n).A.cos(2.pi.Fc.n/Fs + phi)`
M(n) is either a binary periodic signal (0-1) oscillating at frequency FM or a 0-1 sequence defined by W (if specified W overrides FM). OOK.m displays in the current window plot the synthesised signal.

Inputs:

Sp signal structure containing the signal parameters with following fields:

- Fs sampling frequency of the signal (in Hz)
- A amplitude of the carrier
- Fc carrier frequency (in Hz)
- FM modulating frequency (0 = no modulation) (in Hz)
- T duration of the signal (in seconds)

- W binary word to be transmitted (overrides periodic modulation)
- Phi initial phase of the carrier (r.v. unif dist. over $(0, 2\pi)$)
- Class String defining the type of signal S

If varargin is left empty, each field of 'Sp' is defined online

Outputs:

S signal structure containing the synthesised OOK signal with fields:

- data : 1-byN vector containing the data samples
- time : 1-byN vector containing the time samples
- Fs : scalar indicating the sampling frequency

Sp parameter structure (same as input)

M signal structure containing the Modulant signal (same structure as S)

Example :

```
Sp = struct('Fs',50e3,'A',2,'Fc',1e3,'FM',5e1,'Phi',0,'T',1e-1,'W',[])
[S,Sp,M] = OOK(Sp)
plot(S.time,S.data,M.time,M.data,'r')
```

or

```
[S] = OOK()
```

4.2.2 CGN.m

[X,Xp] = CGN(Xp) generates a filtered, centered, gaussian noise X according to the parameters specified in the parameter structure Xp. CGN displays in the current window plot, the synthesised trace and the corresponding estimated power spectrum density.

Input

Xp parameter structure containing the following fields:

- sigma : standard deviation
- Fs : scalar indicating the sampling frequency
- B : the bandwidth (in Hz, $B < F_s/2$)
- T : duration of the generated trace (in seconds)

If varargin is left empty, each field of 'Xp' is defined online

Outputs

- X signal structure with the following fields:
 - data : 1-byN vector containing the data samples
 - time : 1-byN vector containing the time samples
 - Fs : scalar indicating the sampling frequency
- Xp parameter structure (same as input)

Example

```
Xp = struct('sigma',1,'Fs',1000,'B',200,'T',10) ;
[X,Xp] = CGN(Xp) ;
% or
[X,Xp] = CGN() ;
```

4.2.3 AddSig.m

[S] = AddSig(X,Y) Computes the sum Z of the two signals X and Y.

Inputs

X, Y Signal structures with fields:

- data : 1-by-N vector containing the data samples
- time : 1-by-N vector containing the time samples

- Fs : scalar indicating the sampling frequency
- X and Y must have same lengths and same sampling frequencies

Outputs

- Z Signal structures with fields:
- data : 1-by-N vector containing the data samples
 - time : 1-by-N vector containing the time samples
 - Fs : scalar indicating the sampling frequency

4.2.4 BPF.m

[Y,Fp] = BPF(X,Fp) filters the signal structure X with a digital band-pass filter whose parameters are specified in the Fp structure.

BPF displays in a single window plot, the zero-pole diagram, the frequency and the impulse responses of the filter, and superimposed, the input and the output signals.

Inputs

- X input signal structure with the following fields:
 - data : 1-byN vector containing the data samples
 - time : 1-byN vector containing the time samples
 - Fs : scalar indicating the sampling frequency
- Fp parameter structure with following fields:
 - Fs : scalar indicating the sampling frequency (must be identical to that of X)
 - F0 : the central frequency (in Hz)
 - Dnu : the bandwidth
 - order : integer corresponding to the order of the filter
 - class : text string indicating the type of the filter.

Outputs

- Y output signal structure with the following fields:
 - data : 1-byN vector containing the data samples
 - time : 1-byN vector containing the time samples
 - Fs : scalar indicating the sampling frequency
- Fp parameter structure (same as input)

Example:

```
Fp = struct('Fs',1000,'F0',100,'Dnu',32,'order',6,'class','BP filter') ;
[X,Xp] = CGN() ;
Y = BPF(X,Fp) ;
```

4.2.5 SquareSig.m

[Y] = SquareSig(X) Computes the square amplitude Y of signal X.

Inputs

- X Signal structure with fields:
- data : 1-by-N vector containing the data samples
 - time : 1-by-N vector containing the time samples
 - Fs : scalar indicating the sampling frequency

Outputs

- Y Signal structures with fields:
- data : 1-by-N vector containing the data samples
 - time : 1-by-N vector containing the time samples
 - Fs : scalar indicating the sampling frequency

Example:

```
X = OOK() ;
Y = SquareSig(X) ;
```

4.2.6 RCF.m

[Y,RCFp] = RCF (X,RCFp) filters the signal structure X with a digital lowpass RC filter whose parameters are specified in the RCFp structure.

The z-transform of a lowpass RC filter is equal to

$$H(z) = B(z)/A(z) = (1-a) / (1 - a z^{-1})$$

where $a = \exp(-T/RC)$, and T is the sampling period

RCF displays in a single window plot, respectively the time and the frequency responses of the filter, the input signal and the output signal.

Inputs

- X input signal structure with the following fields:
 - data : 1-byN vector containing the data samples
 - time : 1-byN vector containing the time samples
 - Fs : scalar indicating the sampling frequency
- RCFp parameter structure with following fields:
 - Fs : scalar indicating the sampling frequency (must be identical to that of X)
 - RC : scalar defining the time constant RC (must be larger than 1/Fs)

Outputs

- Y output signal structure with the following fields:
 - data : 1-byN vector containing the data samples
 - time : 1-byN vector containing the time samples
 - Fs : scalar indicating the sampling frequency
- RCFp parameter structure (same as input)

4.2.7 RxMessage_DQ.m

[RxMsg,Xp,RxBinMsg] = RxMessage_DQ(X,Xp) performs a Quadratic Detection of binary message conveyed in signal structure X. Xp is a structure that contains all parameters related to the transmission. X and Xp are usually the output of routine TxMessage