

TD Synthèse de Filtres Numériques

FILIERE ELECTRONIQUE, 2^{EME} ANNEE

E. GRIVEL ET J. DANIEL, 2018-2019

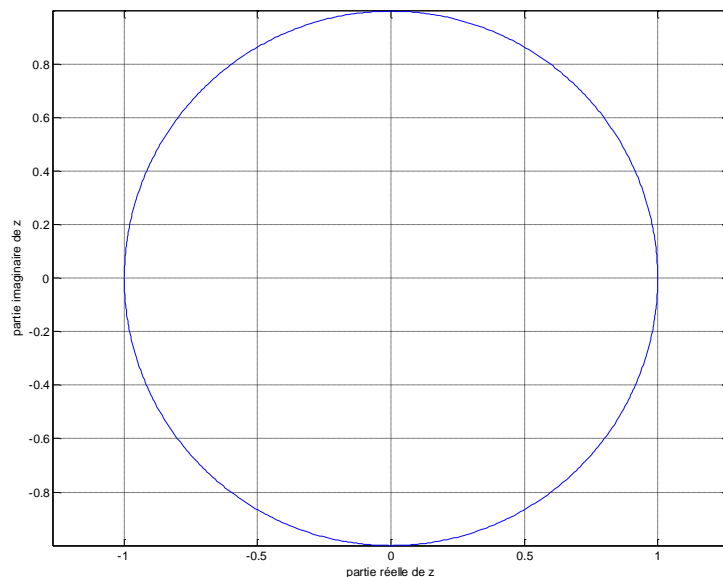
PARTIE N°1

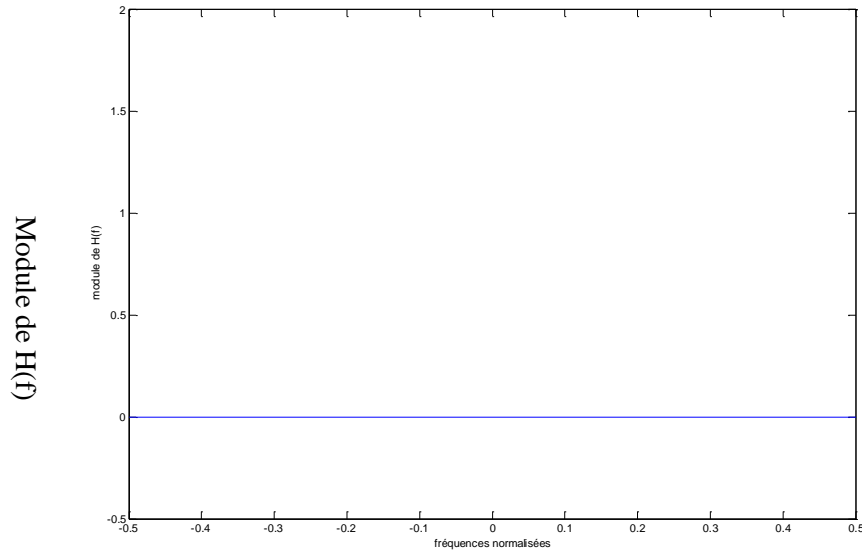
LISTE NON EXHAUSTIVE D'EXERCICES (TD)

1.1 Caractérisation d'un filtre (question d'une épreuve de filtres numériques et estimation, électronique 2013-2014)

Soit le filtre numérique défini par l'équation aux différences suivante $y(n) = 0,8 \times y(n-1) - 0,64 \times y(n-2) + x(n)$ où y et x désignent respectivement la sortie et l'entrée du filtre linéaire. Calculer la fonction de transfert ainsi que les premiers termes de la réponse impulsionnelle. De plus, donner les propriétés (causalité/stabilité/nature RIF ou RII) de ce filtre en justifiant votre réponse. Compléter les figures ci-dessous, en donnant la position des pôles et des zéros et l'allure de la réponse en fréquence du filtre.

Indice : pour les pôles et les zéros, optez pour une expression faisant apparaître le module et l'argument.





1.2 Filtre à échantillonnage en fréquence

Un stagiaire a proposé d'effectuer le traitement du signal $x(n)$ comme présenté dans la figure ci-dessous :

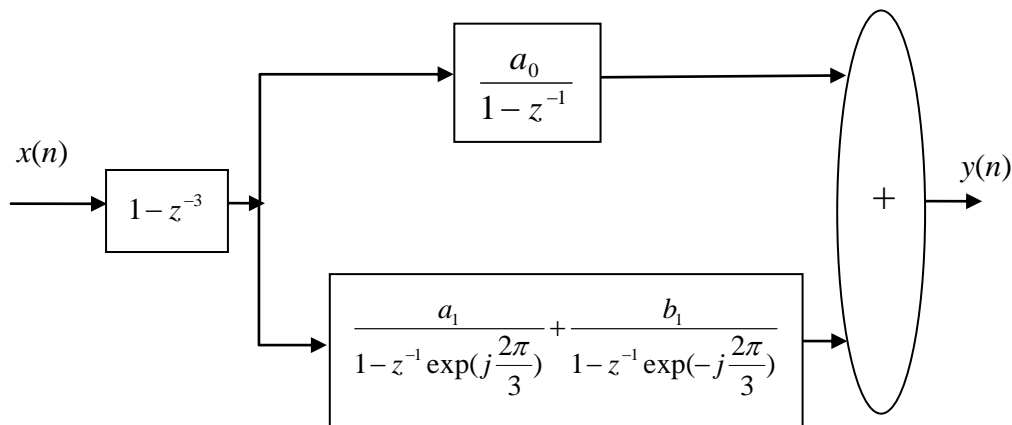


Figure 1 : proposition de filtrage

Question n°1 : commenter ce schéma.

Question n°2 : déterminer les valeurs des coefficients a_0 , a_1 et b_1 , de telle sorte que le filtre envisagé ait une réponse impulsionnelle réelle et que $H(z)|_{z=\exp(j0)} = 3$ et $H(z)|_{z=\exp(j\frac{2\pi}{3})} = 6 + 3j\sqrt{3}$.

1.3 Analyse fréquentielle (exercice d'un examen, filière Electronique, 2012)

On dispose d'observations échantillonnées à 1 KHz comprenant deux composantes : un signal prenant des valeurs réelles et qui est à moyenne nulle et de nature basse fréquence et un bruit additif de nature sinusoïdale haute fréquence, indépendant du signal. Afin de définir le gabarit du filtre à synthétiser pour filtrer le bruit, on effectue préalablement une analyse fréquentielle des données observées, à partir de $N=512$ échantillons. On donne le module au carré de la transformée de Fourier entre 0 Hz et 500 Hz ainsi qu'une représentation temporelle des observations.

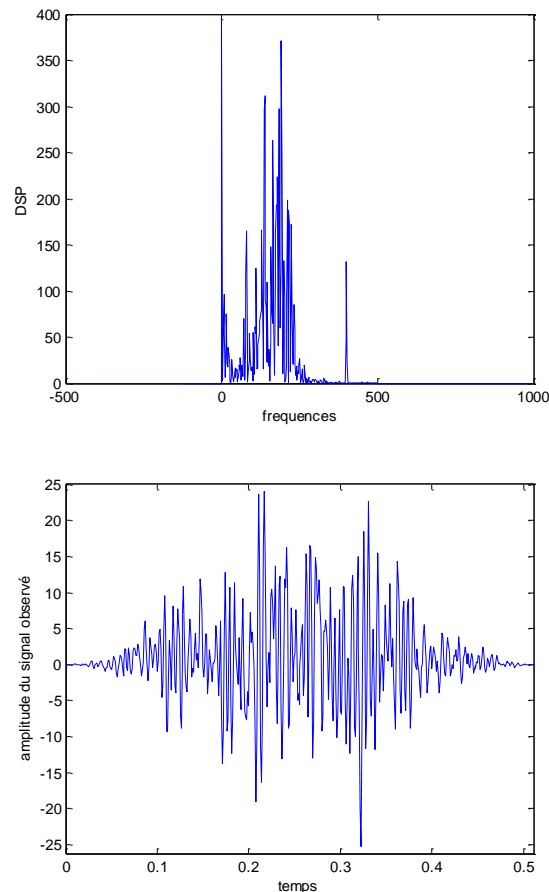


figure à compléter

- Question n°1 : Compléter la figure en précisant ce que l'on obtient entre dans les plages allant de -500Hz à 0Hz et de 500Hz à 1000Hz.
- Question n°2 : Si le signal était à valeurs complexes, pourrait-on, à partir du module au carré de la transformée de Fourier visualisé entre 0 et 500 Hz, en déduire celui entre -500Hz et 0Hz. Justifier votre réponse.
- Question n°3 : Synthétiser un filtre numérique permettant d'éliminer le bruit additif selon la méthode qui vous paraît la plus adaptée. Donner la fonction de transfert du filtre correspondant.

1.4 Transformation bilinéaire et autres

Soit la fonction de transfert d'un filtre continu donnée par

$$H_c(p) = \frac{\sum_{i=0}^N b_i p^i}{1 - \sum_{i=1}^M a_i p^i}.$$

Le système peut être aussi décrit par une équation différentielle contenant le signal d'entrée $x_a(t)$ et le signal de sortie $y_a(t)$, comme suit :

$$y_a(t) = \sum_{i=0}^N b_i \frac{d^i x_a(t)}{dt^i} + \sum_{i=1}^M a_i \frac{d^i y_a(t)}{dt^i}$$

On va s'intéresser à des méthodes de conception de filtres discrets à partir de la fonction de transfert d'un système continu. Etant donné une fonction de transfert d'un filtre continu, on rappelle que la transformation bilinéaire consiste à remplacer p par $\frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}}$ dans la fonction de transfert du filtre continu.

- Question n°1 :
- 1/ Comment aboutit-on à cette transformation ?
 - 2/ Représenter dans le plan z l'image par cette transformation du demi plan gauche du plan p . On rappelle que les axes des abscisses et des ordonnées du plan p sont respectivement ceux des parties réelles et imaginaires de p (de même pour le plan z).
 - 3/ Montrer que la pulsation ω_{continu} dans le domaine continu et celle dans le domaine numérique ω_{discret} sont liées par une relation non linéaire, que l'on précisera.
 - 4/ Quel phénomène s'accompagne de la synthèse d'un filtre numérique RII à partir de la transformation bilinéaire ? Le filtre obtenu est-il toujours stable ?

A présent, partons du fait que l'on peut remplacer la dérivée première du signal de sortie continu par l'approximation suivante :

$$\left. \frac{dy_a(t)}{dt} \right|_{t=nT} \approx \frac{y_a(nT) - y_a(nT - T)}{T}$$

On parlera de différence rétrograde.

- Question n°2 : En notant $y_d(n)$ le $n^{\text{ième}}$ échantillon du signal de sortie d'un système discret, donner une approximation de $\left. \frac{dy_a(t)}{dt} \right|_{t=nT}$ en fonction de $y_d(n)$ et $y_d(n-1)$.

- Question n°3 :
- 1/ Donner une expression des approximations des dérivées d'ordre 2 et 3 de $y_a(t)$ pris aux instants $t=nT$, que l'on notera $y_{d2}(n)$ et $y_{d3}(n)$.
 - 2/ Que valent leurs transformées en z respectives?

- Question n°4 :
- 1/ En déduire une expression de la transformation que l'on peut envisager pour passer du domaine continu au domaine discret, c'est-à-dire une relation entre p et l'opérateur retard z^{-1} .¹
 - 2/ Représenter dans le plan z l'image par cette transformation du demi plan gauche du plan p . On rappelle que les axes des abscisses et des ordonnées du plan p sont respectivement ceux des parties réelles et imaginaires de p (de même pour le plan z).
 - 3/ Etant donné $H_c(p)$, quelle est alors l'expression de $H_d(z)$?

¹ On rappelle que si à $y_a(t)$ correspond $Y_a(p)$, à $\frac{dy_a(t)}{dt}$ correspond $pY_a(p)$.

- Question n°5 : 1/ Si l'on suppose que $\left. \frac{dy_a(t)}{dt} \right|_{t=nT} \approx \frac{y_a(nT+T) - y_a(nT)}{T}$, montrer que l'on peut déduire $H_d(z)$ de $H_c(p)$ par la transformation $p = \frac{z-1}{T}$. On parlera alors de différence progressive.
- 2/ Représenter dans le plan z l'image par cette transformation du demi plan gauche du plan p .
- Question n°6 : Donner alors quelques commentaires sur la stabilité des systèmes obtenus. Quelles conclusions tirez-vous pour chaque méthode ?

1.5 Analyse d'un programme (examen Télécom, rattrapage 2010)

L'objet du programme est de générer un signal comprenant deux modes fréquentiels aux fréquences 100 Hz et 3 KHz, de le perturber par un bruit blanc Gaussien à un rapport signal sur bruit égal à 15dB. Ce signal échantillonné à la fréquence 8 KHz est ensuite filtré soit pour extraire la composante haute fréquence, soit pour extraire la composante basse fréquence. Dans tous les cas, une représentation temporelle et fréquentielle est effectuée sur le signal d'entrée et le signal de sortie des filtres.

Entourer dans ce code *matlab* les maladdresses, incohérences ou erreurs, qui peuvent être de tout type (théorique ou écriture du code) et apporter dans la colonne de droite les corrections nécessaires.

```

N=256 ; .....
RSB=15; .....
k=0:1:N-1; .....
fech=8000; .....
f0=100; .....
f1=3000; .....
x=3*cos(2pi*100/fech*k+pi/4)+5*cos(2*pi*3000/fech*k+pi/3);.....

b=RSB*randn(1,255) .....
x=x+b .....
X=fft(x); .....
subplot(1,1,2) .....
plot(k/fech,x); .....
xlabel('échantillon') .....
ylabel('amplitude') .....
subplot(2,1,2) .....
plot(-fech/2:1:fech/2,abs(fftshift(fft(X)))^2) ; .....
grid .....
xlabel('fréquence normalisée') .....
ylabel('module au carré de la transformée de Fourier de x') .....

title('représentation temporelle et fréquentielle du signal') .....

% filtrage de la composante haute fréquence de x .....
y=filter([1 -4 6 -4 1],1,x); .....

```

```

Y=fft(fftshift(y)); .....
plot(0:1/N:N-1,(abs(Y))^2) ; .....
grid .....
xlabel('fréquence') .....
ylabel('module au carré de la transformée de Fourier de y') .....

title('représentation fréquentielle du signal filtré') .....
% filtrage de la composante basse fréquence (à la fréquence f0) .....
% du signal réel x par un filtre RIF sélectif en fréquence. ....

angle=4*pi*f0/fech ; .....
modulep=0.7; .....
modulez=1 .....
p1=1.1*exp(j*angle); .....
z1=1*exp(j*angle); .....
y2=filter([1 -z1], [1 -p1],x); .....
Y2=fft(y); .....
plot(abs(fftshift(Y)).^2) ; .....
grid .....
xlabel('fréquence normalisée') .....
ylabel('module au carré de la transformée de Fourier de y2') .....

title('représentation fréquentielle du signal filtré y2') .....

```

Remarque : rappel de certaines fonctions matlab

`y = filter(b, a, x)` filters the input data `x` using a rational transfer function defined by the numerator and denominator coefficients `b` and `a`.

`Y = fftshift(X)` rearranges the outputs of `fft`, `fft2`, and `fftn` by moving the zero-frequency component to the center of the array. It is useful for visualizing a Fourier transform with the zero-frequency component in the middle of the spectrum.

`subplot(m, n, p)` divides the current figure into an `m`-by-`n` grid and creates an axes for a subplot in the position specified by `p`. MATLAB® numbers its subplots by row, such that the first subplot is the first column of the first row, the second subplot is the second column of the first row, and so on. If the axes already exists, then the command `subplot(m, n, p)` makes the subplot in position `p` the current axes.

1.6 Filtre inverse (examen de rattrapage, filière Electronique, 2012)

Soit la fonction de transfert d'un filtre numérique défini par

$$H(z) = \frac{1}{1 + 0.8z^{-1} + 0.64z^{-2}} = \frac{1}{A(z)}$$

Question n°1 : Que peut-on dire de l'allure de la réponse en fréquence du filtre de fonction de transfert $H\left(\frac{z}{\gamma}\right)$ lorsque $\gamma > 1$, puis $\gamma < 1$. On analysera pour cela la position des pôles des fonctions de transfert.

Question n°2 : Traiter ensuite le cas d'un filtre dont la fonction de transfert est de la forme $\frac{A(z)}{A\left(\frac{z}{\gamma}\right)}$

Expliquer votre démarche.

Question n°3 : Quelle est la densité spectrale de puissance du signal de sortie du filtre dont la fonction de transfert est $\frac{1}{A(z)}$ quand le signal d'entrée est un bruit blanc Gaussien centré, dont la densité spectrale de puissance vaut σ^2 quelle que soit la fréquence ?

1 Synthèses de filtres numériques par placement des pôles et des zéros

1.1 Objectif

L'objectif de cette partie est de synthétiser un filtre, en se servant uniquement du placement des pôles et des zéros de la fonction de transfert dans le plan complexe. La compréhension fréquentielle de l'influence des pôles et des zéros permet d'obtenir très simplement et efficacement l'expression de la fonction de transfert.

1.2 Expression générale d'un filtre du 2^e ordre

On considère le filtre constitué par deux pôles et deux zéros définis respectivement par :

$$\text{pôles : } \begin{cases} p_1 = \text{Re}^{j\theta_0} = \text{Re}^{j2\pi \frac{f_0}{f_{ech}}} \\ p_2 = \text{Re}^{-j\theta_0} = \text{Re}^{-j2\pi \frac{f_0}{f_{ech}}} \end{cases} \quad (1)$$

$$\text{zéros : } \begin{cases} z_1 = re^{j\theta_0} = re^{j2\pi \frac{f_0}{f_{ech}}} \\ z_2 = re^{-j\theta_0} = re^{-j2\pi \frac{f_0}{f_{ech}}} \end{cases} \quad (2)$$

Sachant que la fonction de transfert $H(z)$ s'exprime sous la forme suivante :

$$H(z) = \frac{\prod_i (1 - z_i z^{-1})}{\prod_i (1 - p_i z^{-1})} \quad (3)$$

exprimer $H(z)$ sous la forme d'une fraction rationnelle de la forme :

$$H(z) = K \frac{\sum_i b_i z^{-i}}{\sum_i a_i z^{-i}} \quad (4)$$

Remarque : On conservera la fonction sous la forme définie par l'équation (4) afin d'être cohérent avec les prototypes des fonctions MATLAB.

1.3 Filtre caractérisé uniquement par des pôles

Soit le filtre dont la fonction de transfert $H_1(z)$ se caractérise uniquement par deux pôles définis par p_1 et p_2 . On opte dans un premier temps pour $\theta_0 = \pi/3$, $r = 0$ et $R = 0,95$.

Étape 1 : fonction de transfert

1. Donner l'expression de la fonction de transfert de $H_1(z)$.
1. Placer ses pôles et ses zéros dans le plan complexe. On pourra utiliser dans ce cas-là, la fonction `zplane` (voir remarque sur `zplane` ci-après).
2. Représenter la réponse en fréquence de son module et de sa phase. On pourra utiliser la fonction `freqz` (voir remarque sur `freqz` ci-après).
3. Commenter.



MATLAB

zplane :

zplane permet d'afficher les pôles et les zéros d'une fonction de transfert discrète. Si elle se présente sous la forme $H(z) = \frac{B(z)}{A(z)} = \frac{b(1)+b(2)z^{-1}+\dots+b(n+1)z^{-n}}{a(1)+a(2)z^{-1}+\dots+a(m+1)z^{-m}}$, la fonction `zplane(b,a)` affiche le cercle unité ainsi que les pôles et les zéros correspondants. On peut également les représenter par leurs parties réelles et imaginaires. L'intérêt est qu'il est possible de superposer sur une même figure les pôles et les zéros de différentes fonctions de transfert. Il faudra dans ce cas représenter également le cercle unité par la fonction `plot`.

```
N = 100;
theta = -pi:2*pi/N:pi-1/N;
figure
plot(cos(theta), sin(theta)) % representation du cercle unité
hold on % valide la superposition des tracés
plot(0,0,'+k') % localization de l'origine
axis image % repère orthonormé
plot(real(p1), imag(p1), '*r')
plot(real(p2), imag(p2), '*r')
plot(real(z1), imag(z1), 'ob')
plot(real(z2), imag(z2), 'ob')
hold off
```



freqz permet d'obtenir la réponse fréquentielle d'un filtre numérique. On désigne respectivement par b et a les coefficients du numérateur et du dénominateur de la fonction de transfert en z rangées suivant les puissances croissantes de z^{-1} :

$$H(z) = \frac{B(z)}{A(z)} = \frac{b(1) + b(2)z^{-1} + \dots + b(n+1)z^{-n}}{a(1) + a(2)z^{-1} + \dots + a(m+1)z^{-m}}$$

Plusieurs prototypes d'appel de la fonction sont possibles. La manière la plus efficace est de déterminer la valeur de la fonction de transfert sur des pulsations que l'on aura définies au préalable :

`h = freqz(b, a, 2*pi*f);`

Il faut donc définir un vecteur contenant la plage de fréquences normalisées souhaitées (entre 0 et $\frac{1}{2}$ ou entre $-\frac{1}{2}$ et $\frac{1}{2}$, etc.), puis d'appeler la fonction pour qu'elle retourne la réponse fréquentielle complexe associée. Il est ainsi plus facile d'afficher son module et son argument en graduant correctement l'axe fréquentiel.

Exemple : on désire afficher le module et la phase du filtre défini par la fonction de transfert $H(z)$, sur N points dans la gamme de fréquences normalisées $\left[-\frac{1}{2} \dots \frac{1}{2} - \frac{1}{N}\right]$ où a et b représentent respectivement les coefficients du dénominateur et du numérateur de la fonction de transfert :

```
. . .
f = -1/2:1/N:1/2-1/N;
h = freqz(b, a, 2*pi*f);
figure
plot(f, abs(h))
ou
plot(f, 20*log10(abs(h)))
. . .
```

Étape 2 : caractérisation du filtre

On considère à présent un bruit blanc gaussien de moyenne m et de variance σ^2 .

1. Quelles sont les propriétés fréquentielles d'un tel signal aléatoire ?
2. Générer ce signal avec $m=0$ et $\sigma^2=1$ à l'aide de la fonction `randn`.
3. Filtrer ce signal à l'aide du filtre défini à l'étape 1 (on utilisera pour cela la fonction `filter`).
4. Tracer sur une même figure et sur deux sous-figures différentes, les représentations temporelles du signal en entrée et du signal en sortie du filtre.
5. Tracer sur une autre figure, le spectre de puissance du signal filtré (voir remarque ci-après).
6. Comparer le résultat obtenu avec la densité spectrale de puissance (DSP) théorique (voir remarque sur la DSP ci-après).
7. Reprendre les points précédents en modifiant la puissance du signal d'entrée.
8. Commenter.



Spectre de puissance et densité spectrale de puissance (DSP) d'un bruit blanc filtré :

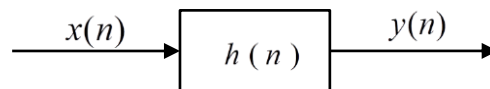
Spectre de puissance :

Le spectre de puissance d'un signal discret est défini comme étant la norme au carré de sa transformée de Fourier divisé par le nombre de points :

$$\text{spectre de puissance} = \frac{|Y(f)|^2}{N}$$

Densité spectrale de puissance :

Soit un filtre caractérisé par sa réponse impulsionnelle $h(n)$ et sa réponse fréquentielle $H(f)$.



$x(n)$ désigne un bruit blanc de variance σ^2 et $y(n)$ le signal $x(n)$ filtré par $h(n)$: $y(n) = x(n) * h(n)$, soit $Y(f) = X(f)H(f)$ et $Y^*(f) = X^*(f)H^*(f)$. La DSP de $y(n)$ est donc :

$$DSP_y = Y(f)Y^*(f) = |H(f)|^2 |X(f)|^2 = |H(f)|^2 \sigma^2.$$

Le spectre de puissance est dépendant de la réalisation du signal $y(n)$, contrairement à la densité spectrale de puissance qui ne l'est pas.

Étape 3 : influence du paramètre θ_0

1. Quelle est la plage de variation de la pulsation angulaire normalisée θ_0 ?
2. Faire varier θ_0 , pour un module R fixé à 0,95 et observer la position des pôles ainsi que la réponse fréquentielle du filtre considéré (on prendra pour cela quelques valeurs pertinentes dans son domaine de définition).
3. Quelle est l'influence de la pulsation normalisée θ_0 sur la réponse fréquentielle du filtre ?

Étape 4 : influence du paramètre R

1. Quelle est la condition sur la valeur de R sur la stabilité du filtre ?
2. Faire varier R , pour $\theta_0 = \pi/3$ et observer la position des pôles ainsi que la réponse fréquentielle du filtre considéré.
3. Quelle est l'influence du module R sur la réponse fréquentielle du filtre ?

1.4 Filtre caractérisé uniquement par des zéros

Soit le filtre dont la fonction de transfert $H_2(z)$ n'a pas de pôles et se caractérise uniquement par deux zéros définis par z_1 et z_2 . On opte pour la pulsation normalisée $\theta_0 = \pi/3$, $r = 0,95$ et $R = 0$.

Étape 1 : fonction de transfert

1. Donner l'expression de la fonction de transfert de $H_2(z)$.

2. Placer ses pôles et ses zéros dans le plan complexe.
3. Représenter la réponse en fréquence de son module et de sa phase.
4. Commenter.

Étape 2 : caractérisation du filtre

1. Générer un bruit blanc gaussien de moyenne nulle et de variance unité ($m=0$ et $\sigma^2=1$)
2. Filtrer ce signal à l'aide du filtre défini à l'étape 1.
3. Tracer les représentations temporelles du signal en entrée et en sortie du filtre.
4. Tracer le spectre de puissance des deux signaux.
5. Quel est la différence de comportement fréquentiel entre un filtre ne comportant que des pôles et celui ne comportant que des zéros ?

Étape 3 : influence du paramètre θ_0

1. Quelle est la plage de variation de la pulsation normalisée θ_0 ?
2. Faire varier θ_0 , pour un module r fixé à 0,95 et observer la position des zéros ainsi que la réponse fréquentielle du filtre considéré.
3. Quelle est l'influence de la pulsation normalisée θ_0 sur la réponse fréquentielle du filtre ?

Étape 4 : influence du paramètre r

1. Quelle est la plage de variations du paramètre r ?
2. Faire varier r , pour une pulsation normalisée fixée à $\theta_0 = \pi/3$ et observer la position des zéros ainsi que la réponse fréquentielle du filtre considéré.
3. Quelle est l'influence du module r sur la réponse fréquentielle du filtre ?

1.5 Filtre caractérisé par des pôles et des zéros

On considère à présent la fonction de transfert $H(z)$ comportant à la fois des pôles et des zéros ($R \neq 0$ et $r \neq 0$).

Étape 1 : $r < R$

1. On considère $\theta_0 = \frac{\pi}{3}$, $R = 0,99$ et $r = 0,8$.
2. Tracer sur une même figure, les réponses fréquentielles en module et en phase des filtres suivants :

$$H_1(z) = \frac{1}{(1 - p_1 z^{-1})(1 - p_2 z^{-2})}$$

$$H_2(z) = (1 - z_1 z^{-1})(1 - z_2 z^{-2})$$

$$H(z) = H_1(z)H_2(z)$$

3. Commenter le résultat obtenu en mettant en évidence les avantages qu'apporte la contribution des zéros dans la réponse fréquentiel du filtre $H_1(z)$.

Étape 2 : $r > R$

1. Reprendre les points de l'étape 1 avec cette fois-ci les paramètres suivants $\theta_0 = \pi/3$, $R = 0,9$ et $r = 0,99$.
2. Commenter le résultat en le comparant à celui obtenu à l'étape 1.

1.6 Comment faire de la musique avec du bruit ?

On sait que la densité spectrale de puissance d'un bruit blanc à moyenne nulle est une constante. Ainsi, il est possible en utilisant des filtres très sélectifs d'obtenir à partir d'un bruit blanc, une fréquence particulière. L'objectif de cette manipulation est de générer un banc de filtres suffisamment sélectifs afin de pouvoir jouer un morceau de musique à partir d'une source de bruit blanc. Les notes composant le morceau à jouer appartiennent toutes à la même gamme d'octave et les fréquences des notes à utiliser sont les suivantes :

N° de la note	1	2	3	4	5	6	7
Note	Do	Ré	Mi	Fa	Sol	La	Si
Fréquence f_{note} (Hz)	261.6	293.7	329.7	349.2	392	440	493.9

Figure. 1 Liste des fréquences des notes utilisées

Paramètres :

Durée d'une note : 0,4s

Fréquence d'échantillonnage : $f_{ech} = 8 \text{ KHz}$

Préliminaires :

Dans notre cas, la note de musique (qui est une "sinusoïde") est obtenue par filtrage d'un bruit blanc par un filtre très sélectif. Celui-ci est modélisé par une fonction de transfert ne possédant pas de zéros et caractérisée uniquement par deux pôles complexes conjugués définis comme suit :

$$\begin{cases} p_1 = \text{Re}^{j\theta_{note}} = \text{Re}^{j2\pi \frac{f_{note}}{f_{ech}}} \\ p_2 = \text{Re}^{-j\theta_{note}} = \text{Re}^{-j2\pi \frac{f_{note}}{f_{ech}}} \end{cases}$$

1. Réaliser un filtre très sélectif autour d'une fréquence musicale quelconque.
2. Générer un bruit blanc dont la durée est celle d'une note.
3. Filtrer ce signal et écouter le résultat.
4. Assurez-vous d'écouter la bonne fréquence : soit par votre oreille musicale soit dans le domaine fréquentiel.
5. Répéter l'opération sur les autres notes.

Génération de la partition musicale

La partition à jouer est la suivante (figure 2) :



Note	Do	Do	Do	Ré	Mi	Ré	Do	Mi	Ré	Ré	Do
N° de la note	1	1	1	2	3	2	1	3	2	2	1
Durée (temps)	1	1	1	1	2	2	1	1	1	1	1.5

Figure. 2 Partition musicale

Mise en œuvre : A chaque note est associée une durée, la noire compte pour 1 temps, la blanche pour 2 temps, etc. Un temps correspond à 0,4 secondes. Le formalisme à utiliser pour jouer une note est donc le suivant :

1. Constituer un vecteur indexant la fréquence de la note à jouer en Hz en fonction de l'index de la note (1 pour Do, 2 pour Ré, etc.) (Cf. Figure. 1)
2. Constituer la partition sous la forme d'une matrice $2N$, où N représente le nombre de notes à jouer. Sur la première ligne est codé l'index des notes, tandis que sur la deuxième est codée la durée associée en nombre de temps (Cf. Figure. 2) :

```
Partition = [ 1 1 1 2 3 2 1 3 2 2 1 ; ... % indexe de la fréquence de la note
             1 1 1 1 2 2 1 1 1 1 1.5 ]; % durée de la note
```

3. Balayer par un index les colonnes du tableau partition, puis générer un bruit blanc dont la durée correspond à la valeur de la deuxième ligne du tableau, ainsi que le filtre adéquat permettant de générer la fréquence désirée.
4. Quelle célèbre comptine reconnaissez-vous ?

Remarque : La fonction Matlab permettant de jouer une partition est `soundsc`.



MATLAB

soundsc :

Scale data and play as sound

soundsc (y, Fs) sends audio signal y to the speaker at sample rate F_s .

2 Localisation d'une cible par signal RADAR

- **Contexte** : estimer la distance séparant une cible d'une antenne
- **Données** : signal émis, écho reçu, à savoir le signal émis retardé, déformé et bruité.
- **But** : débruiter l'écho reçu afin d'en mesurer le décalage temporel par rapport au signal émis
- **Méthode** : utilisation d'un filtre adapté

2.1 Introduction

On désire estimer la distance qui sépare une antenne d'une cible. Pour cela, on considère un système radar comportant une antenne émettant un signal noté $x(t)$. Ce signal est réfléchi par une cible située à une distance r , le signal rétrodiffusé $y(t)$ est ensuite capté par l'antenne du radar au bout d'un temps t_0 (cf. Figure 1). Dans tout le problème, on considère le signal $x(t)$ comme étant à support fini $d_t(t) = 0$ pour $t \in]-\infty; 0[\cup]t_d; +\infty[$ (cf Figure 2).

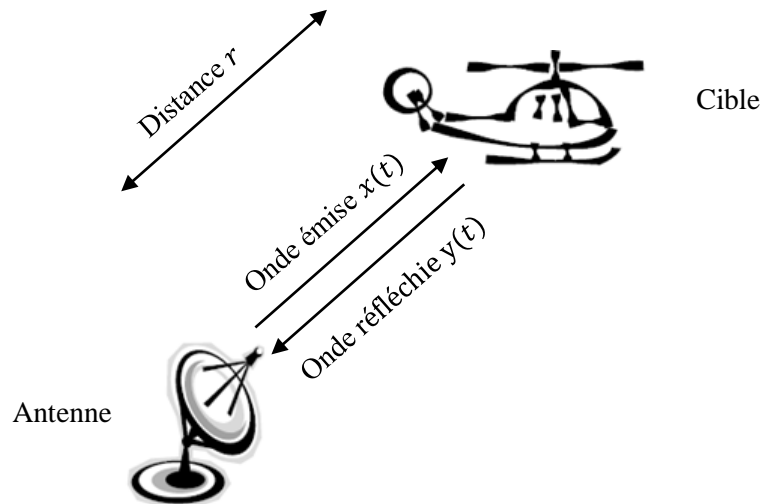


Figure 1 : Schéma de principe d'un radar



Figure 2 : Exemple de forme d'onde émise $x(t)$

Dans le cas où l'onde réfléchie $y(t)$ n'est ni déformée ni bruitée, l'estimation du temps t_0 est directement accessible, il correspond à la différence de temps qui sépare l'émission de l'onde $x(t)$ à l'apparition de l'onde $y(t)$ (cf. Figure 3)

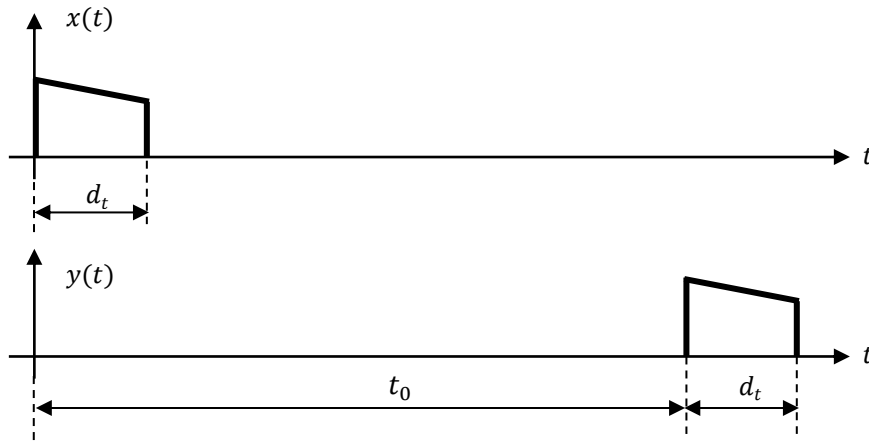


Figure 3 : Onde émise $x(t)$ et réfléchie $y(t)$

En pratique, le signal réfléchi $y(t)$ est déformé et bruité à tel point qu'il est difficile d'estimer le temps à partir duquel celui-ci apparaît réellement (cf. Figure 4). Dans cette configuration, il est nécessaire de filtrer le signal $y(t)$, afin d'estimer le temps t_0 .

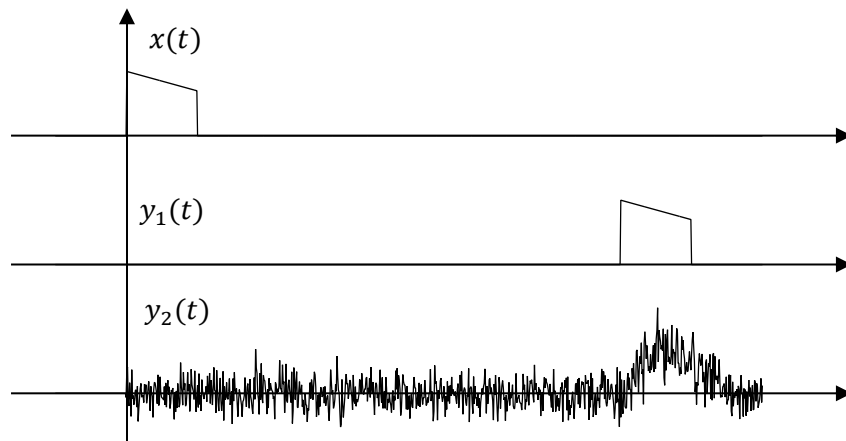


Figure 4 : Onde émise $x(t)$ et réfléchie $y(t)$ dans les cas idéal et réel

2.2 Considérations théoriques

On se place dans le cas où le signal émis $x(t)$ est réfléchi sans déformation et avec une atténuation d'un facteur .

Question 1 : Donner la relation qui lie le signal rétrodiffusé $y_1(t)$ capté par l'antenne au signal émis $x(t)$.

Question 2 : Donner la relation qui lie le retard t_0 à la distance r qui sépare la cible de l'antenne.

Question 3 : Donner l'expression de la fonction de corrélation $R_{y_1x}(\tau)$ entre les signaux $x(t)$ et $y_1(t)$ en fonction de la fonction d'autocorrélation de $x(t)$, $R_{xx}(\tau)$, (on rappelle que $y_1(t) = x(t - t_0)$).

Question 4 : Comment, à partir de l'expression de $R_{y_1x}(\tau)$, estimer le temps t_0 ?

Question 5 : On se place maintenant dans un cas réel où le signal reçu est bruité : $y_2(t) = \alpha x(t - t_0) + b(t)$, où $b(t)$ désigne un bruit blanc gaussien de moyenne nulle et de variance σ^2 . Donner l'expression de la fonction de corrélation $R_{y_2x}(\tau)$ entre les signaux $x(t)$ et $y_2(t)$. On l'exprimera en fonction de la fonction d'autocorrélation de $x(t)$ et de la fonction d'intercorrélation entre $x(t)$ et $b(t)$ désignées respectivement par, $R_{xx}(\tau)$, et $R_{xb}(\tau)$.

Question 6 : Comment, à partir de l'expression de $R_{y_2x}(\tau)$, estimer le temps t_0 et quelle condition doit être respectée ?

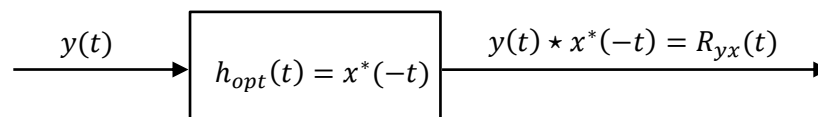


Produit de convolution et fonction de corrélation, filtrage optimal et intercorrélation

La fonction d'intercorrélation $R_{yx}(\tau)$ entre les signaux $y(t)$ et $x(t)$, peut être obtenue en filtrant le signal $y(t)$ par $x^*(-\tau)$, en effet :

$$R_{yx}(\tau) = \int_{-\infty}^{+\infty} y(t) x^*(t - \tau) dt = \int_{-\infty}^{+\infty} y(t) x^*(-(\tau - t)) dt = y(\tau) * x^*(-\tau)$$

On montre que $y^*(-\tau)$ est le filtre optimal pour extraire les paramètres α et t_0 , c'est celui qui maximise le rapport signal/bruit en sortie (cf. cours).



2.3 Mise en œuvre

On se propose de filtrer différents signaux par des filtres optimaux, afin d'estimer la distance r , pour cela on dispose de plusieurs couples de signaux $x(t)$ et $y(t)$ référencées dans le tableau 1.

N°	Nom du fichier	$x(t)$	$y(t)$
1	signal_radar_config1.mat	$\Pi_{d_t} \left(t - \frac{d_t}{2} \right)$	$x(t - t_0)$
2	signal_radar_config2.mat	$\cos(2\pi k t^2) \Pi_{d_t} \left(t - \frac{d_t}{2} \right)$	$x(t - t_0)$
3	signal_radar_config3.mat	$\Pi_{d_t} \left(t - \frac{d_t}{2} \right)$	$x(t - t_0) + b_1(t)$
4	signal_radar_config4.mat	$\cos(2\pi k t^2) \Pi_{d_t} \left(t - \frac{d_t}{2} \right)$	$x(t - t_0) + b_1(t)$
5	signal_radar_config5.mat	$\Pi_{d_t} \left(t - \frac{d_t}{2} \right)$	$x(t - t_0) + b_2(t)$
6	signal_radar_config6.mat	$\cos(2\pi k t^2) \Pi_{d_t} \left(t - \frac{d_t}{2} \right)$	$x(t - t_0) + b_2(t)$

Tableau 1 : Différents signaux émis et reçus ($\Pi_T(t)$ désignant la fonction porte centrée de largeur T)

Les signaux $b_1(t)$ et $b_2(t)$ désignent des bruits blancs gaussiens. Dans les configurations (3) et (4) le rapport signal sur bruit est plus important que dans les configurations (5) et (6).

Remarques :

- Ces signaux sont téléchargeables sur le site : <http://daniel.enseirb.free.fr/TS208>
- Les variables **x** et **y** correspondant respectivement aux signaux émis et rétrodiffusés $x(t)$ et $y(t)$ sont accessibles dans chacune des configurations en chargeant le fichier sous Matlab par la commande `load('signal_radar_config1.mat');`
- Durée du signal $d_t = 10 \mu s$
- Fréquence d'échantillonnage : $f_{ech} = 200 MHz$

Question 7 : Pour chacune des configurations proposées :

1. Tracer les signaux émis et rétrodiffusés $x(t)$ et $y(t)$
2. Tracer la réponse impulsionnelle du filtre adapté $h_{opt}(t)$
3. Tracer le signal filtré $w(t) = y(t) * h_{opt}(t)$
4. A partir du signal $w(t)$, estimer le temps t_0 , ainsi que la distance r lorsque cela est possible
5. Commenter

Question 8 : Dans le cas où le signal $y(t)$ est bruité, quel est selon vous, la configuration qui permet d'estimer le temps t_0 avec la meilleure précision ?

Question 9 : En partant des signaux des configurations (1) et (2) :

1. Ajouter un bruit blanc gaussien de moyenne nulle, en diminuant progressivement le rapport signal sur bruit du signal $y(t) = x(t - t_0) + b(t)$.
2. Indiquer pour chacune d'elles, la valeur du rapport signal/bruit à partir duquel il n'est plus possible d'estimer correctement t_0 .
3. Commenter

3 Annulation d'écho d'une conversation téléphonique

- **Contexte** : annuler l'écho lors d'une réception téléphonique
- **Données** : signal émis, signal reçu avec écho
- **But** : supprimer l'écho
- **Méthode** : utilisation d'un filtre adapté

3.1 Contexte

Lors d'une conversation téléphonique entre deux interlocuteurs A et B, où l'un d'eux utilise un système mains-libres, il se produit un phénomène d'écho acoustique. L'un des interlocuteurs recevra le message vocal de son correspondant superposé de son propre message atténué et décalé (cf. Figure 1).

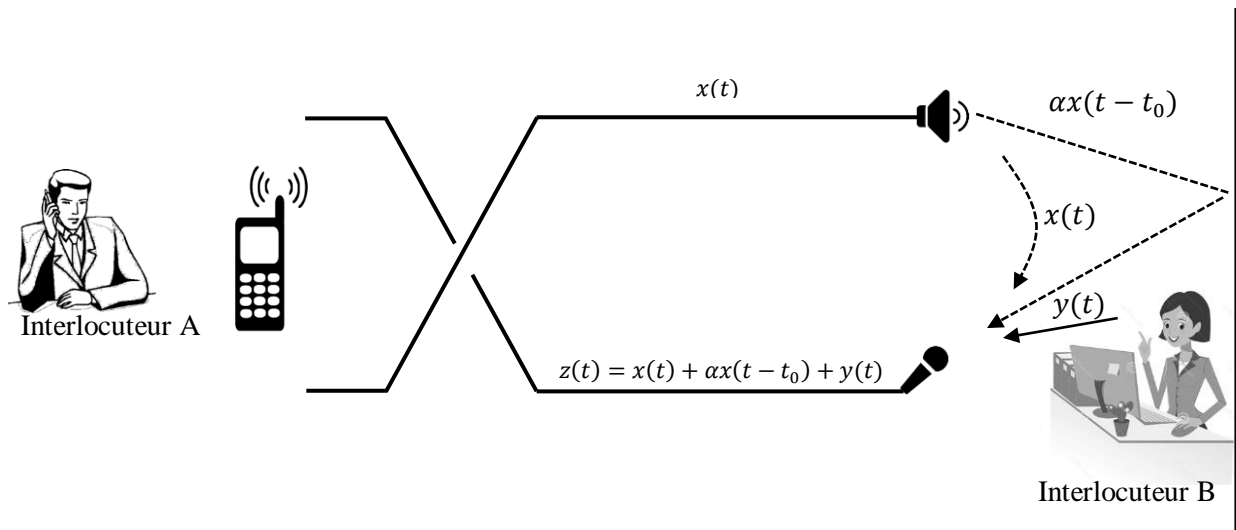


Figure 1 : Conversion téléphonique avec écho

L'interlocuteur A émet le signal $x(t)$, qui par couplage lui revient décalé et atténué, son correspondant B, ayant en même temps émis le signal $y(t)$. Le haut-parleur de A reçoit par conséquent la superposition du message transmis par B auquel vient s'ajouter son propre message atténué et décalé. L'interlocuteur B, reçoit donc le message $z(t) = x(t) + \alpha x(t - t_0) + y(t)$. Le but est de trouver le filtre permettant à l'interlocuteur A de ne recevoir que le message $y(t)$. Le filtre doit pouvoir supprimer l'écho issu de $x(t)$. Dans ce contexte, le signal utile est $y(t)$, le bruit étant $b(t) = x(t) + \alpha x(t - t_0)$.

3.2 Mise en œuvre

La mise en œuvre de l'annulation d'écho dans le cas présent, est de supprimer le signal perturbateur $b(t) = x(t) + \alpha x(t - t_0)$ au signal reçu $z(t) = x(t) + \alpha x(t - t_0) + y(t)$ afin de n'en conserver que le signal utile $y(t)$ (figure 2). Pour cela il est nécessaire d'estimer les paramètres α et t_0 .

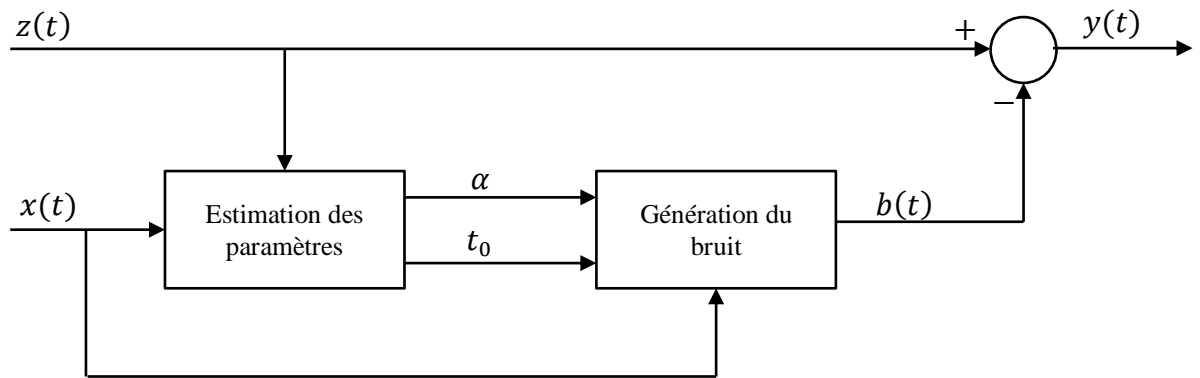


Figure 2 : schéma de principe de suppression de l'écho

L'annulation d'écho se déroule en trois phases :

- Estimation des paramètres de retard et d'atténuation t_0 et α
- Génération du bruit $b(t) = x(t) + \alpha x(t - t_0)$ à partir de ces paramètres
- Soustraction le bruit généré au signal reçu

Remarques :

- Les signaux $x(t)$ et $z(t)$ sont téléchargeable sur le site : <http://daniel.enseirb.free.fr/TS208>
- Les variables $x(t)$ et $z(t)$ sont accessibles en chargeant le fichier sous Matlab par la commande `load('annulation_echo.mat');`
- Fréquence d'échantillonnage : $f_{ech} = 44600 \text{ Hz}$

Question 1 : Connaissant les signaux $x(t)$ et $z(t)$, donner l'expression de la réponse impulsionnelle du filtre optimal $h_{opt}(t)$ permettant d'estimer les paramètres t_0 et α .

Question 2 : Donner l'expression du signal $w(t)$ résultant du filtrage de $z(t)$ par le filtre $h_{opt}(t)$. On fera intervenir la fonction d'autocorrélation du signal $x(t)$, ainsi que la fonction d'intercorrrelation du signal $x(t)$ et $z(t)$. Quelles conditions doivent être respectées afin d'estimer correctement les paramètres ?

Question 3 : Filtrer le signal $z(t)$ par le filtre $h_{opt}(t)$ et le tracer.

Question 4 : Estimer les paramètres t_0 et α à partir du signal filtré $w(t) = y(t) * h_{opt}(t)$ (on se servira également de l'expression obtenue à la question 2).

Question 5 : Donner l'expression du filtre $h(z)$ permettant d'obtenir le bruit $b(t)$ à partir du signal $x(t)$.

Question 6 : Générer le bruit $b(t) = x(t) + \alpha x(t - t_0)$, et le soustraire du signal reçu $y(t)$. Ecouter le résultat. Commenter.

4 Synthèse de filtres numériques à partir de filtres analogiques

4.1 Objectif

Lors de l'acquisition de certains signaux, les données recueillies peuvent être entachées de bruits (bruits issus des capteurs, des lignes de transmissions, du codage, etc.). L'objectif de cette partie, est de pouvoir synthétiser correctement des filtres afin d'atténuer l'influence de ces données parasites, et ainsi de ne conserver que l'information pertinente. On supposera par la suite que le signal utile n'est situé que dans les basses fréquences, alors que le signal parasite l'est dans les hautes fréquences. L'opération permettant d'isoler l'information pertinente est obtenue par un filtrage passe bas du signal considéré.

4.2 Mode opératoire

Afin de répondre aux contraintes définies précédemment, il convient de synthétiser un filtre passe bas numérique, dont les fréquences de coupures ainsi que les atténuations sont à adapter suivant le signal considéré. Plusieurs méthodes permettent de synthétiser un filtre numérique, l'une d'elles consiste à tirer profit des nombreux résultats établis sur la synthèse de filtres à temps continu, puis de le ramener dans l'espace discret. On se propose donc dans un premier temps, de synthétiser un filtre dans le domaine des fréquences continues, puis de le ramener dans celui des fréquences discrètes.

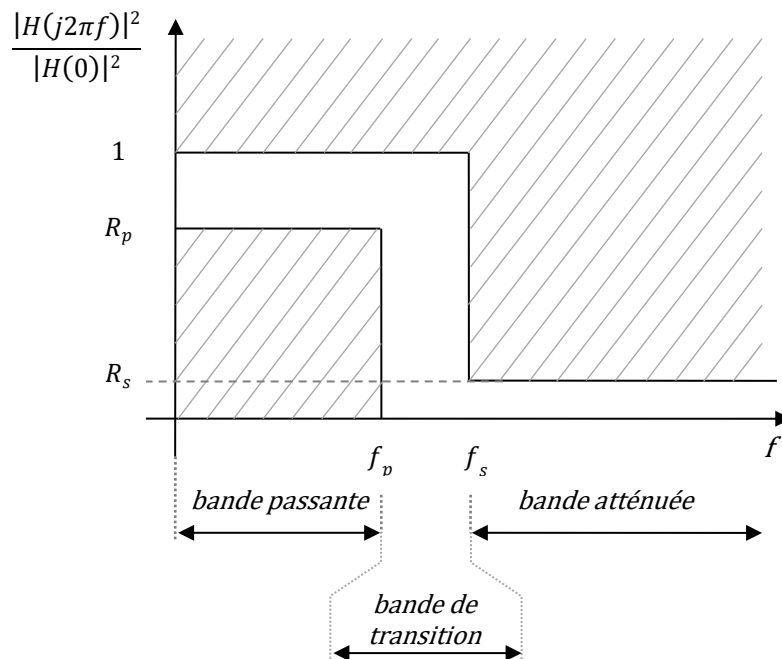


Figure 1 : Gabarit d'un filtre passe-bas

4.3 Synthèse de filtres analogiques

Commençons par synthétiser un filtre analogique passe-bas dont les caractéristiques sont les suivantes :

- limite de la bande passante : 0; 5 dB à 3200 Hz
- limite de la bande atténuée : 40 dB à 4000 Hz

Filtre de Butterworth

(On utilisera les fonctions `butter`, `buttord` avec l'option 's' ainsi que la fonction `freqs`.)

1. Rappeler brièvement les principales propriétés du filtre de Butterworth.
2. Synthétiser le filtre de Butterworth suivant les caractéristiques demandées.
3. Quel est l'ordre du filtre et sa fréquence de coupure ?
4. Représenter la réponse en fréquence du filtre.

Filtre de Tchebychev de type I

(On utilisera les fonctions `cheby1` et `cheb1ord` avec l'option 's' ainsi que la fonction `freqs`.)

1. Rappeler brièvement les principales propriétés d'un filtre de Tchebychev, et quelles sont ses avantages par rapport à un filtre de Butterworth.
2. Synthétiser le filtre de Tchebychev de type I suivant les caractéristiques demandées.
3. Quel est l'ordre du filtre ?
4. Représenter la réponse en fréquence du filtre.

Filtre de Tchebychev de type II

(On utilisera les fonctions `cheby2`, `cheb2ord` avec l'option 's' ainsi que la fonction `freqs`.)

1. Rappeler les différences entre un filtre de Tchebychev de type I et celui de type II.
2. Synthétiser le filtre de Tchebychev de type II suivant les caractéristiques demandées.
3. Quel est l'ordre du filtre ?
4. Représenter la réponse en fréquence du filtre.

4.4 Synthèse de filtres numériques

Concevons à présent un filtre numérique par la méthode de la transformation bilinéaire respectant le gabarit suivant :

- fréquence d'échantillonnage : 1 Hz
- atténuation maximale de 1 dB pour la fréquence 0,1 Hz
- atténuation minimale de 15 dB pour la fréquence 0,15 Hz.
- On traitera le cas d'un filtre de Butterworth, puis celui d'un filtre de Tchebychev de type I.

La transformation bilinéaire

1. Rappeler la relation qui permet de passer de la variable continue p à la variable discrète z : c'est-à-dire la relation qui permet de passer du demi-plan correspondant à la partie réelle négative de la variable p , au disque unité du plan de la variable z (figure 2).

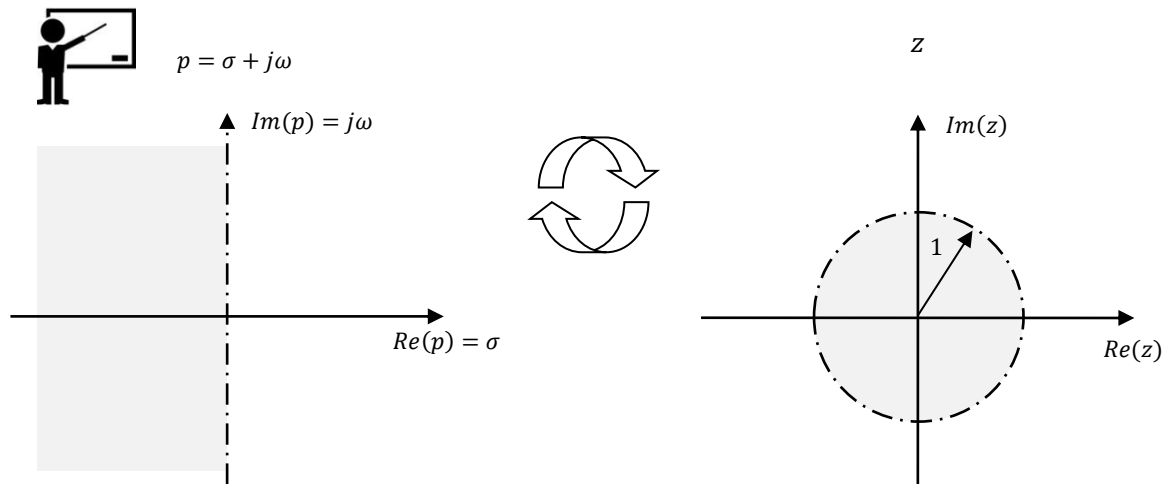


Figure 2 : Correspondance entre les plans complexes p et z

Rappel : La transformation bilinéaire permet à partir du gabarit d'un filtre continu, d'obtenir le gabarit équivalent discret et inversement. La relation permettant de passer des pulsations du domaine continu à celles du domaine discret, et qui traduit le phénomène de warping, est la suivante :

$$\omega_{\text{continu}} = \frac{2}{T_{\text{ech}}} \tan\left(\frac{\omega_{\text{discret}} T_{\text{ech}}}{2}\right)$$

2. Rappeler quelles sont les conséquences de cette transformation sur les basses et les hautes fréquences.

Synthèse du filtre

1. Étant donné le phénomène de warping dû à la transformation bilinéaire, déduire du gabarit du filtre numérique, le gabarit du filtre analogique sur lequel il faudra effectuer la transformation bilinéaire (Figure 3).
2. A partir des pulsations continues que vous venez de calculer, synthétiser les filtres analogiques de Butterworth et de Tchebychev de type I satisfaisant le gabarit analogique de l'étape 1. Représenter leur réponse en fréquence.
3. Appliquer la transformation bilinéaire en utilisant `bilinear`. Représenter alors la réponse en fréquence et en phase du filtre numérique en utilisant `freqz` et `phasez`.
4. Reprendre la même démarche en utilisant directement pour les fonctions dédiées aux filtres numériques de Butterworth et de Tchebychev. On évitera donc dans ce cas d'utiliser 's' dans la fonction `butter`, `cheby1`.
5. Commenter.

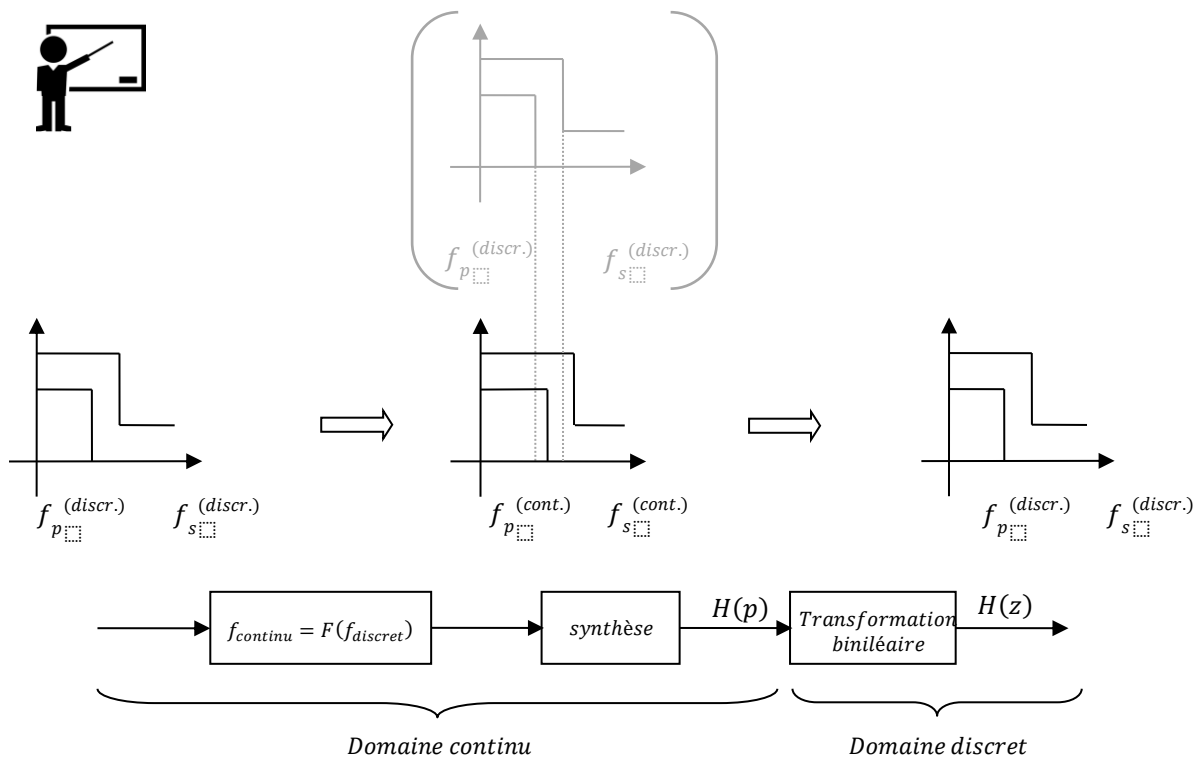


Figure 3 Étapes de conception d'un filtre numérique par la transformation bilinéaire

4.5 Application 1 : filtrage d'un signal de très basse fréquence

On considère le signal issu de l'enregistrement du débit d'une rivière en fonction du temps, il est généré au rythme d'une acquisition par seconde ($f_{ech} = 1Hz$) et les capteurs ont induit une perturbation de type haute fréquence. On désire filtrer ce signal afin de ne conserver que les informations basses fréquences situées entre 0 Hz et 0,1 Hz.

1. Les signaux sont téléchargeables sur le site : <http://daniel.enseirb.free.fr/TS208>
2. Charger le signal hydrology par la commande `load('hydrology.mat')`.
3. Représenter temporellement ce signal ainsi que sa densité spectrale de puissance.
4. A l'aide des filtres synthétisés au §4.4, filtrer ce signal afin de n'en conserver que les informations utiles.
5. Représenter sur une même figure le signal d'origine et le signal filtré.
6. Représenter sur une même figure les densités spectrales correspondantes.
7. Commenter.

4.6 Application 2 : filtrage d'un signal biomédical

On considère à présent un enregistrement correspondant à l'acquisition d'un électrocardiogramme dont la fréquence d'échantillonnage $f_{ech} = 1500 Hz$. Celui-ci est perturbé par un signal de haute fréquence.

1. Les signaux sont téléchargeables sur le site : <http://daniel.enseirb.free.fr/TS208>
2. Charger le signal cardio 2 par la commande `load('cardio.mat')`.
3. Représenter temporellement ce signal ainsi que sa densité spectrale.
4. Repérer la gamme de fréquences correspondant au signal utile, ainsi que celle correspondant au signal parasite.

5. En reprenant la démarche décrite au §4.4, concevoir un filtre passe-bas capable de supprimer les fréquences indésirables.
6. Filtrer le signal.
7. Représenter sur une même figure le signal d'origine et le signal filtré.
8. Représenter sur une même figure les densités spectrales correspondantes.
9. Commenter.

Liste (non exhaustive) des fonctions MATLAB utiles

La liste suivante est tirée de l'aide de chacune des fonctions, relative à la toolbox signal processing.

zplane: this function displays the poles and zeros of discrete-time systems. `zplane(b,a)` where `b` and `a` are row vectors.

The transfer function is defined in terms of z^{-1}

$$H(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_{n_b} z^{-n_b}}{a_0 + a_1 z^{-1} + \dots + a_{n_a} z^{-n_a}}$$

filter: `y = filter(b, a, x)` filters the input data `x` using a rational transfer function defined by the numerator and denominator coefficients `b` and `a`

$$H(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_{n_b} z^{-n_b}}{a_0 + a_1 z^{-1} + \dots + a_{n_a} z^{-n_a}}$$

buttord: calculates the minimum order of a digital or analog Butterworth filter required to meet a set of filter design specifications.

Digital domain : `[n,Wn] = buttord(Wp,Ws,Rp,Rs)`.

Analog domain : `[n,Wn] = buttord(Wp,Ws,Rp,Rs,'s')`.

butter: **Digital domain :** `[b,a] = butter(n,Wn)` returns the transfer function coefficients of an `n`th-order lowpass digital Butterworth filter with normalized cutoff frequency `Wn`.

Analog domain : `[b,a] = butter(n,Wn,'s')` returns the transfer function coefficients of an `n`th-order lowpass digital Butterworth filter cutoff angular frequency `Wn`.

chebord1 calculates the minimum order of a digital or analog Chebychev Type I filter required to meet a set of filter design specifications.

Digital domain : `[n,Wn] = cheb1ord(Wp,Ws,Rp,Rs)`.

Analog domain : `[n,Wn] = cheb1ord(Wp,Ws,Rp,Rs,'s')`.

cheby1 **Digital domain :** `[b,a] = cheby1(n,Rp,Wn)` returns the transfer function coefficients of an `n`th-order lowpass digital Chebychev Type I filter with normalized cutoff frequency `Wn` and `Rp` decibels of peak-to-peak passband ripple.

Analog domain : `[b,a] = cheby1(n,Rp,Wn,'s')` returns the transfer function coefficients of an `n`th-order lowpass digital Chebychev filter cutoff angular frequency `Wn` and `Rp` decibels of peak-to-peak passband ripple.

chebord2 calculates the minimum order of a digital or analog Chebychev Type II filter required to meet a set of filter design specifications.

Digital domain : `[n,Wn] = cheb2ord(Wp,Ws,Rp,Rs)`.

Analog domain : `[n,Wn] = cheb2ord(Wp,Ws,Rp,Rs,'s')`.

cheby2 **Digital domain :** `[b,a] = cheby2(n,Rs,Wn)` returns the transfer function coefficients of an `n`th-order lowpass digital Chebychev Type II filter with normalized cutoff frequency `Wn` and `Rs` decibels of stopband attenuation down from the peak passband value.

Analog domain : `[b,a] = cheby2(n,Rs,Wn,'s')` returns the transfer function coefficients of an `n`th-order lowpass digital Chebychev Type II filter cutoff angular frequency `Wn` and `Rs` decibels of stopband attenuation down from the peak passband value.

freqs freqs returns the complex frequency response $H(j\omega)$ (Laplace transform) of an analog filter $[h,w] = \text{freqs}(b,a,n)$ uses n frequency points to compute the frequency response, h , where n is a real, scalar value. The frequency vector w is auto-generated and has length n . If you omit n as an input, 200 frequency points are used. If you do not need the generated frequency vector returned, you can use the form $h = \text{freqs}(b,a,n)$ to return only the frequency response, h .

$h = \text{freqs}(b,a,w)$ returns the complex frequency response of the analog filter specified by coefficient vectors b and a . freqs evaluates the frequency response along the imaginary axis in the complex plane at the angular frequencies in rad/s specified in real vector w , where w is a vector containing more than one frequency.

freqz $[h,w] = \text{freqz}(b,a,n)$, returns the n -point frequency response vector, h , and the corresponding angular frequency vector, w , for the digital filter with numerator and denominator polynomial coefficients stored in b and a , respectively.

$h = \text{freqz}(b,a,w)$ returns the frequency response vector, h , at the normalized frequencies supplied in w .

bilinear $[\text{numd},\text{dend}] = \text{bilinear}(\text{num},\text{den},\text{fs})$ convert an s -domain transfer function given by num and den to a discrete equivalent. Row vectors num and den specify the coefficients of the numerator and denominator, respectively, in descending powers of s . Let $B(s)$ be the numerator polynomial and $A(s)$ be the denominator polynomial. The transfer function is :

$$\frac{B(s)}{A(s)} = \frac{b_0 s^n + \dots + b_{n-1} s + b_n}{a_0 s^n + \dots + a_{n-1} s + a_n}$$

fs is the sampling frequency in hertz.

bilinear returns the discrete equivalent in row vectors numd and dend in descending powers of z (ascending powers of z^{-1}).