

# Traitement Numérique du Signal

(Support de cours)

Master 1 : Systèmes de Télécommunications

Faculté d'Electronique et d'Informatique

Université des Sciences et de la Technologie Houari Boumediene

Site perso : <http://perso.usthb.dz/~akourgli/>

e-mails: [akourgli@usthb.dz](mailto:akourgli@usthb.dz) , [assiakourgli@gmail.com](mailto:assiakourgli@gmail.com)

*FEI, USTHB [assiakourgli@gmail.com / http://perso.usthb.dz/~akourgli/*

Introduction et rappels divers

- Analyse temporelle des signaux discrets
- Systèmes linéaires et invariants discrets (SLID)
- Filtres à réponses impulsionnelles finies ou infinies
- Rappels succincts sur la Transformée de Fourier (TF)

I. Analyse fréquentielle des SLID

- Quelques rappels sur l'échantillonnage idéal
- Transformée de Fourier à temps discret
- Transformée de Fourier Discrète
- Calcul de la FFT (TFR)
- TFD des signaux de longueur illimitée
- Fenêtres de pondération
- Exercices

II. Analyse des filtres numériques par la transformée en Z (TZ)

- Transformées en Z
- Propriétés de la TZ
- TZ rationnelles
- Détermination de la réponse en fréquence des FN
- Détermination de la réponse impulsionnelle des FN(TZ inverse)
- Caractéristiques des FN
- Exercices

III. Synthèse des filtres numériques RIF

- Filtres RIF à phase linéaire (les quatre cas)
- Synthèse des filtres RIF par la méthode des fenêtres
- Synthèse des filtres RIF par la méthode de l'échantillonnage fréquentiel
- Constitution et réalisation de filtres RIF
- Exercices

#### IV. Synthèse des filtres numériques RII

- Rappels sur les filtres analogiques
- Synthèse des filtres RII par la méthode des pôles et des zéros
- Synthèse des filtres RII par la méthode de la réponse impulsionnelle
- Synthèse des filtres RII par la méthode de la transformation bilinéaire
- Constitution et réalisation de filtres numériques RII
- Exercices

#### V. Filtres numériques multicaudences

- Sous échantillonnage et suréchantillonnage
- Systèmes multicaudence et analyse spectrale
- Banc de filtres et décomposition polyphases
- Applications de traitement multicaudence
- Exercices

#### VI. Transformation en ondelettes discrètes (DWT)

- Dualité temps-fréquence et transformée de Fourier à court terme. Inconvénients.
- Ondelettes continues, discrètes (DWT) et ondelettes dyadiques
- Exemples de DWT (Haar, Daubechies, ...etc)
- Analyse multi-résolution
- Exemples d'applications
- Exercices

#### Travaux Pratiques

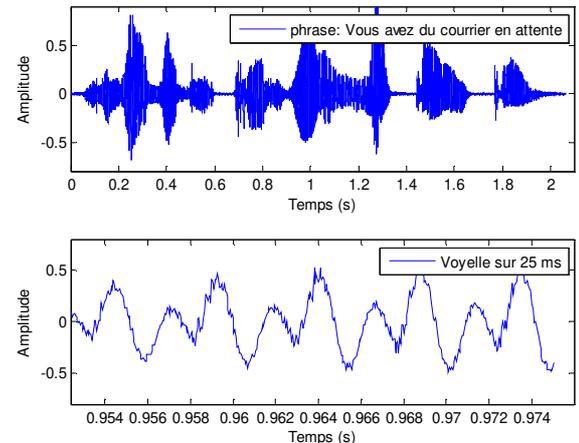
1. Analyse fréquentielle des filtres numériques (TFD) et Fenêtrage
2. Analyse des filtres numériques par la TZ
3. Conception de filtres numériques RIF
4. Conception de filtres numériques RII
5. Filtrage Multi-Cadences
6. Analyse temps-fréquence et temps-échelle

## Introduction et rappels divers

**Un signal** est la représentation physique de l'information qu'il transporte de sa source à son destinataire. Il sert de vecteur à une information. Il constitue la manifestation physique d'une grandeur mesurable (courant, tension, force, température, pression, etc.). Les signaux sont des grandeurs électriques variant en fonction du temps  $x(t)$  obtenues à l'aide de capteurs. Sur le plan analytique : Un signal sera une fonction d'une variable réelle, en général le temps.

### Exemples :

- Onde acoustique : délivré par un microphone (parole, musique, ...)
- Signaux biologiques : EEG, ECG
- Tension aux bornes composant électronique
- Signaux géophysiques : vibrations sismiques
- Finances : cours du pétrole
- Images, Vidéos



**Remarque:** Tout signal physique comporte une *composante* aléatoire (perturbation externe, bruit, erreur de mesure, etc ...). **Le bruit** est défini comme tout phénomène perturbateur gênant la perception ou l'interprétation d'un signal, par analogie avec les nuisances acoustiques (interférence, bruit de fond, etc.). La différenciation entre le signal et le bruit est artificielle et dépend de l'intérêt de l'utilisateur : les ondes électromagnétiques d'origine galactique sont du bruit pour un ingénieur des télécommunications par satellites et un signal utile pour les radioastronomes.

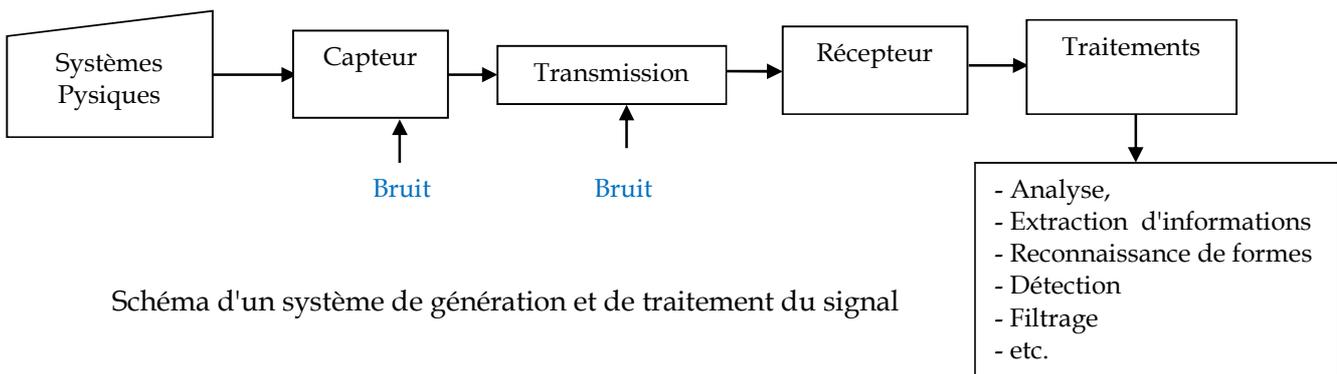


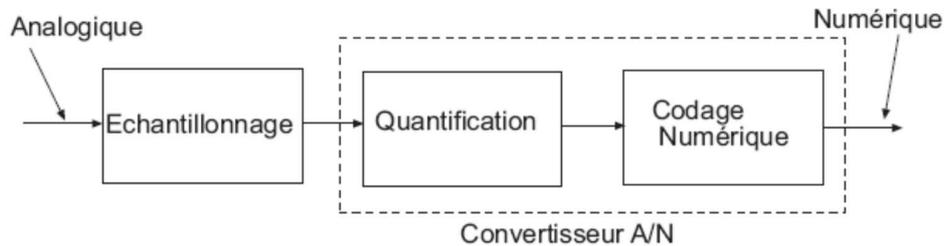
Schéma d'un système de génération et de traitement du signal

Les fonctions du traitement du signal peuvent se diviser en deux catégories : l'élaboration des signaux (incorporation des informations) et l'interprétation des signaux (extraction des informations). Les principales fonctions intégrées dans ces deux parties sont les suivantes [1]:

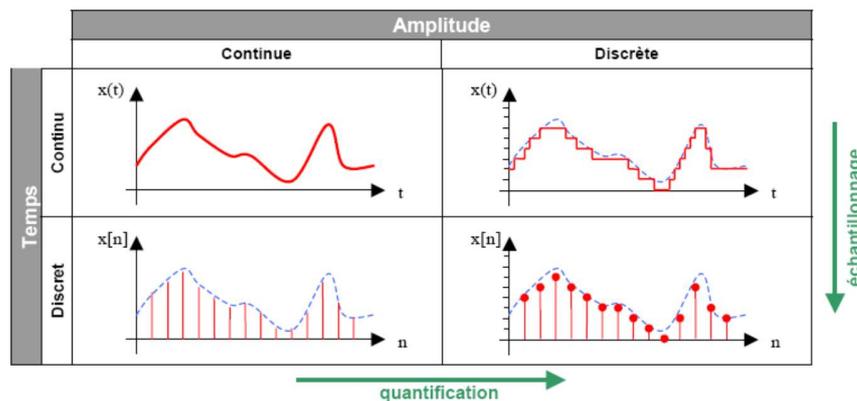
*Élaboration des signaux* : synthèse, modulation, codage/compression, etc.

*Interprétation des signaux* : filtrage, détection, identification, analyse, mesure, etc.

La numérisation d'un signal est l'opération qui consiste à faire passer un signal de la représentation dans le domaine des temps et des amplitudes continus au domaine des temps et des amplitudes discrets. Cette opération de numérisation d'un signal peut être décomposée en deux étapes principales : échantillonnage et quantification.



La restitution (ou l'interpolation) constitue le processus inverse qui intervient lors du passage du signal numérique au signal analogique : commande d'un actionneur. Ces trois étapes sont indissociables. En effet, le signal, étant le support physique d'une information, doit conserver au cours de ces modifications tout le contenu informatif initial. Cette condition, ajoutée à la notion de coût limite d'un système, va être à la base de la numérisation des signaux et de l'étude du traitement numérique [1].



Les traitements numériques sont aisément réalisés grâce à des additionneurs, des multiplieurs numériques, et des mémoires. En outre, les systèmes numériques possèdent de nombreux avantages comparés à ceux analogiques, entre autres [2]:

- Simplicité: Les systèmes numériques sont intrinsèquement plus simples à analyser (et donc à synthétiser) que les systèmes analogiques
- Possibilités de traitement accrues: Il est possible de réaliser, en numérique, des opérations beaucoup plus complexes qu'en analogique, notamment des opérations non-linéaires.
- Robustesse aux bruits. Les systèmes numériques sont par essence insensibles aux bruits parasites électromagnétiques. Le transcodage de l'information sous forme numérique joue un peu le rôle de « firewall ».
- Précision et stabilité. Puisque les seuls « bruits » sont liés à la précision des calculs, cette dernière dépend uniquement du calculateur utilisé ; elle est insensible à la température et ne varie pas avec l'âge du système.
- Flexibilité. Dans un grand nombre de systèmes numériques, le traitement est défini par un logiciel chargé en mémoire. Il est dès lors très facile de modifier ce traitement, sans devoir modifier la machine qui le réalise.

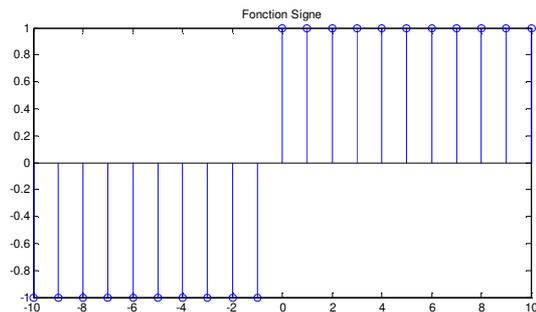
1. Analyse temporelle des Signaux Discrets

Rappelons que les signaux déterministes renferment une information dont l'évolution en fonction du temps peut être parfaitement prédite par un modèle mathématique (au contraire des signaux aléatoires/stochastiques). Nous présentons dans cette section quelques fonctions mathématiques supports de signaux élémentaires et utilisées tout au long du cours de traitement du signal numérique. Nous y introduisons, également, quelques outils permettant de caractériser et d'établir une analyse temporelle des signaux numériques.

Rappelons qu'un signal à temps discret provient souvent de l'échantillonnage à la cadence  $f_e = 1/T_e$ , d'un signal  $x(t)$  déterministe à temps continu qui est supposé à bande limitée  $(-f_e/2, f_e/2)$ . Nous noterons  $x(n) = x(nT_e)$ .

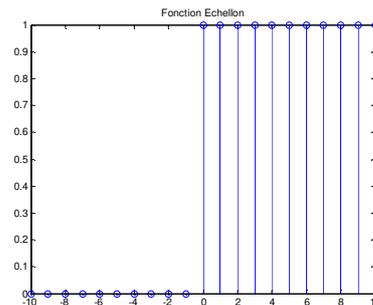
- Fonction signe

$$\text{sgn}(n) = \begin{cases} 1 & n \geq 0 \\ -1 & n < 0 \end{cases}$$



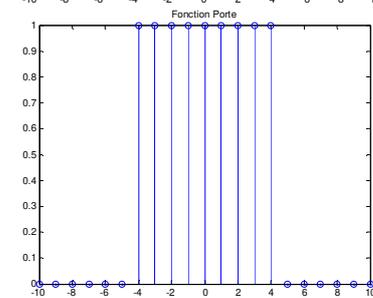
- Fonction échelon (unité)

$$U(n) = \Gamma(n) = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}$$



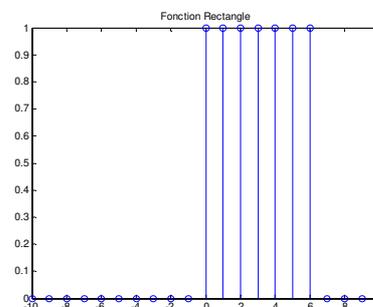
- Fonction porte

$$\Pi_{N+1}(n) = \begin{cases} 1 & -N/2 \leq n \leq N/2 \\ 0 & \text{ailleurs} \end{cases}$$



- Fonction rectangle causal

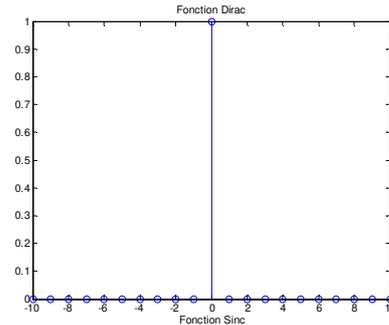
$$\text{rect}(n/N) = \begin{cases} 1 & 0 \leq n \leq N-1 \\ 0 & \text{ailleurs} \end{cases}$$



- Fonction Dirac (impulsion unité)

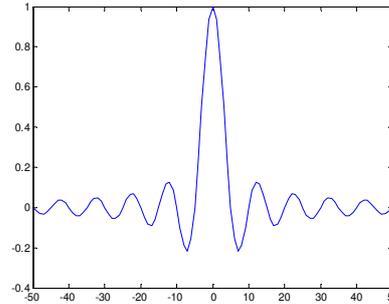
$$\delta(n) = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} = U(n) - U(n-1)$$

•  $x(n) \cdot \delta(n-n_0) = x(n_0)$       •  $x(n) * \delta(n-n_0) = x(n-n_0)$



- Fonction sinus cardinal

$$\text{sinc}(\theta n) = \frac{\sin(\pi\theta n)}{\pi\theta n}$$



Energie et puissance

Toute transmission d’information s’accompagne de transferts d’énergie. En effet, les signaux continus ou discrets sont essentiellement caractérisés par l’énergie ou la puissance qu’ils véhiculent. Ce sont les seules grandeurs physiques auxquelles sont sensibles les détecteurs. Beaucoup de capteurs physiques mesurent une énergie ou une quantité quadratique. Par exemple, les capteurs optiques mesurent une intensité, les compteurs d’électricité mesurent une énergie, etc. Compte tenu de la définition fondamentale, l’énergie du signal entre les instants t et t+dt est :  $|x(t)|^2 dt$  (puissance instantanée multipliée par le temps).

Soit un signal  $x(n)$  à temps discret, tel que  $\sum_{-\infty}^{+\infty} |x(n)|^2$  existe et converge. Alors le signal est dit à énergie finie et la valeur de cette somme est appelée énergie du signal :  $E_x = \sum_{-\infty}^{+\infty} |x(n)|^2$

Exemples:

$x(n) = \text{Rect}(n/N)$  énergie finie.  $x(n) = a$  (constante) et  $x(n) = A \sin(2\pi f_0 n)$  ne sont pas à énergie finie

Pour un signal périodique, cette somme ne converge pas. On peut néanmoins définir la puissance d’un signal  $x(n)$  périodique de période N par :

$$P_x = \frac{1}{N} \sum_{-N/2}^{N/2-1} |x(n)|^2 \quad \text{ou} \quad P_x = \frac{1}{2.N} \sum_{-N}^{N-1} |x(n)|^2$$

Dans le cas général, on parle de signaux à puissance moyenne finie définie par:

$$P_x = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{-N/2}^{N/2-1} |x(n)|^2 \quad \text{ou} \quad P_x = \lim_{N \rightarrow \infty} \frac{1}{2.N} \sum_{-N}^{N-1} |x(n)|^2$$

Exemples:

signal continu  $x(t)=a, A \sin(2\pi f_0 t)$ , signaux périodiques, échelon unité, peigne de Dirac.

Il existe des signaux ni périodiques, ni d'énergie finie, pour lesquels la puissance ne peut être définie, comme par exemple la rampe  $x(n)=n$ . Il s'agit là de définitions mathématiques, en pratique, un signal mesuré ne l'est jamais sur un intervalle de temps infini. On peut commencer à visualiser un signal à un instant qu'on prendra comme origine des temps, et dans ce cas on arrêtera son examen au bout d'un temps  $T_{obs}$ :

$$E_x = \sum_{n=0}^{N_{obs}} |x(n)|^2$$

Remarques

Signal à énergie finie  $\Rightarrow$  puissance nulle      Signal à puissance finie  $\Rightarrow$  énergie infinie

Le calcul de l'énergie ou la puissance permet d'obtenir une première caractérisation du signal. Par ailleurs, la théorie du signal a largement développé des méthodes d'étude basées sur la corrélation pour caractériser le comportement temporel du signal.

Exercice d'application : Calculer l'énergie et la puissance des signaux :  $\Pi_7(n), A \cos(2\pi f_0 n)$ .

Corrélation et auto-corrélation

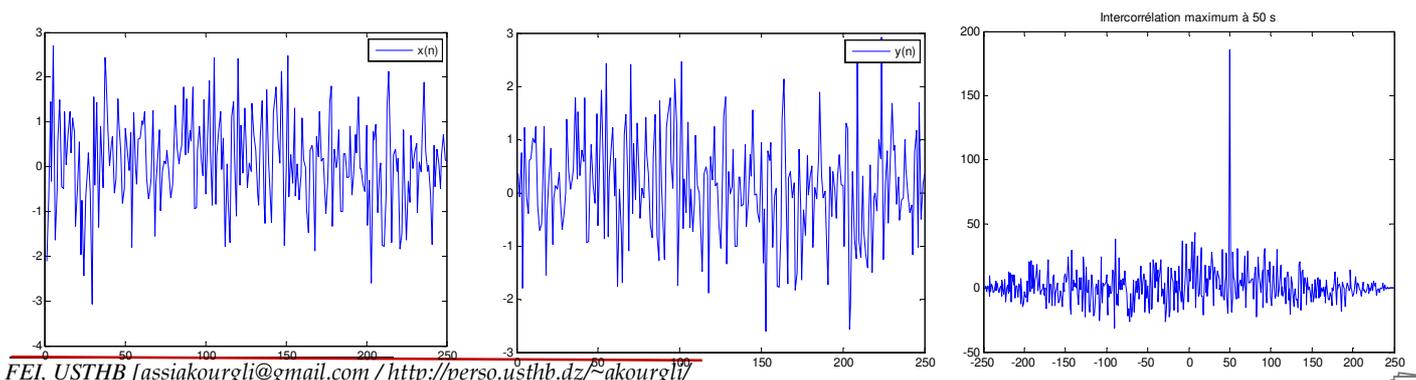
La fonction de corrélation permet de mesurer le degré de ressemblance entre deux signaux en fonction d'un décalage. Considérons  $x(n)$  et  $y(n)$  deux signaux d'énergie finie, la fonction d'intercorrélation  $R_{x,y}(k)$  est définie

par: 
$$R_{xy}(k) = \sum_{n=-\infty}^{\infty} x(n)y^*(n-k)$$

L'inter-corrélation entre  $x(t)$  et  $y(t)$  atteint un maximum pour un retard  $k$  si  $x(n)=y(n-k)$

Pour des signaux à puissance moyenne finie, elle vaut : 
$$R_{xy}(k) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=-N/2}^{N/2-1} x(n)y^*(n-k)$$

Exemples: Soient un signal aléatoire et sa version décalée de 50s. On remarque que les signaux se ressemblent le plus quand  $y(n)$  est décalé de 50 secondes.



Pour l'auto-corrélation, on remplace  $y(n)$  par  $x(n)$  on obtient l'expression de l'auto-corrélation pour les signaux à énergie finie:  $R_x(k) = \sum_{n=-\infty}^{\infty} x(n)x^*(n-k)$

L'auto-corrélation permet de détecter des régularités, des profils répétés dans un signal comme un signal périodique perturbé par beaucoup de bruit (Voir TP n°0)

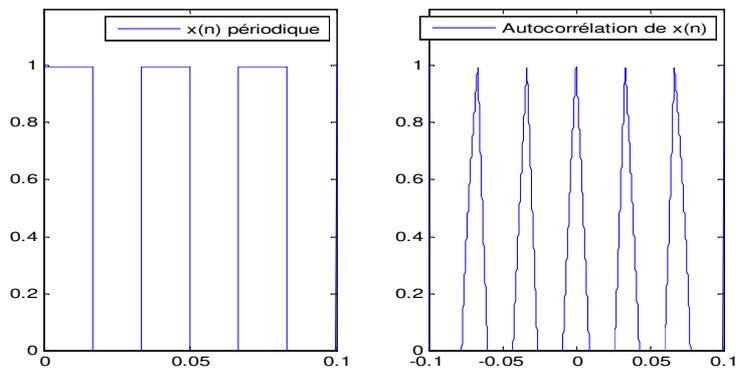
Propriétés :

- Pour  $k=0$ , on retrouve l'énergie du signal  $R_{xx}(0) = E_x$  et  $R_{xx}(k)$  est maximale en  $k=0$
- Si  $x(n)$  est réel, l'auto-corrélation est réelle et paire.
- L'auto-corrélation d'un signal de durée  $N$  aura une taille  $2*N-1$

Auto-corrélation des signaux périodiques : Le calcul sur une seule période suffit. L'auto-corrélation d'un signal périodique est-elle même périodique. Par définition, le signal périodique ressemble parfaitement à lui-même, décalé d'une ou plusieurs périodes.

- signaux périodiques

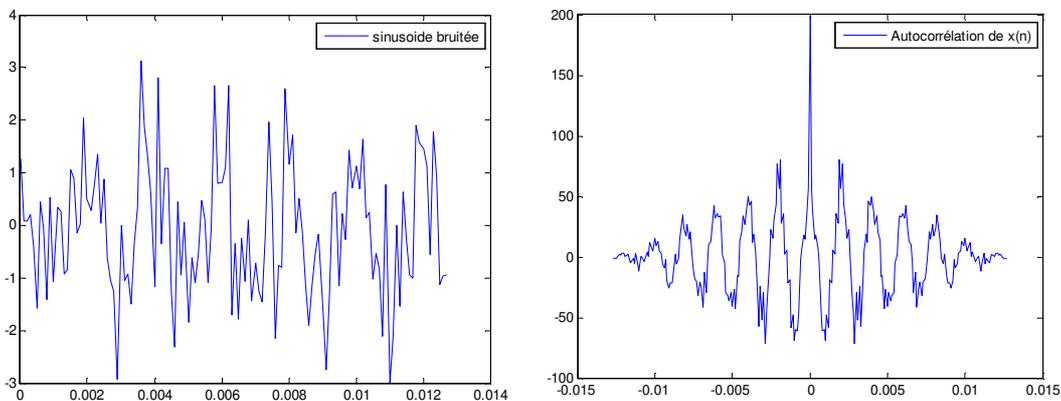
$$R_x(k) = \frac{1}{N} \sum_{n=1}^{N-n} x(n) x^*(n-k)$$



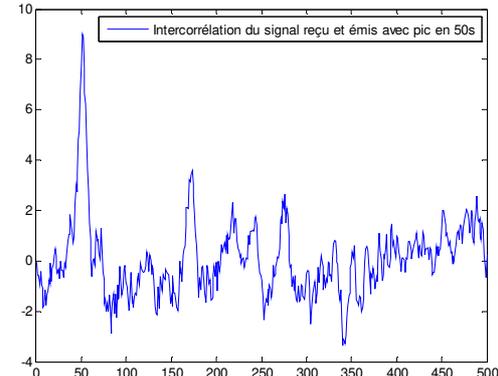
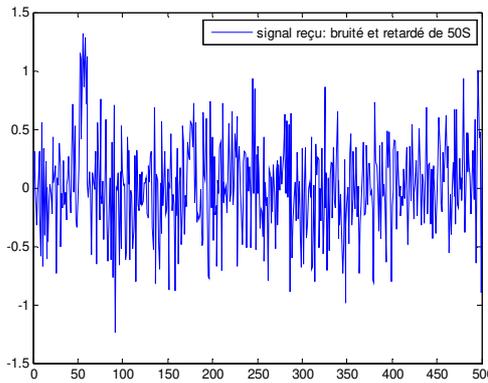
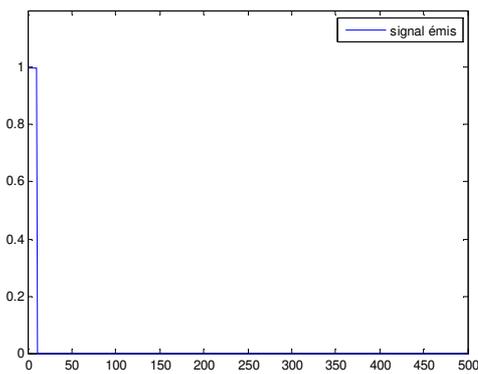
Exercice d'application Soit le signal  $x(n)=(n+1)$  pour  $n=0$  à 3

Calculer l'auto-corrélation de  $x$  et déduire son énergie

Exemples d'application : Extraction d'un signal noyé dans du bruit, mesure d'un temps ou retard, détection d'un signal périodique (Voir TP n° 0). L'exemple ci-dessous illustre l'auto-corrélation d'un signal sinusoïdal d'amplitude 1 noyé dans du bruit Gaussien de variance 1.

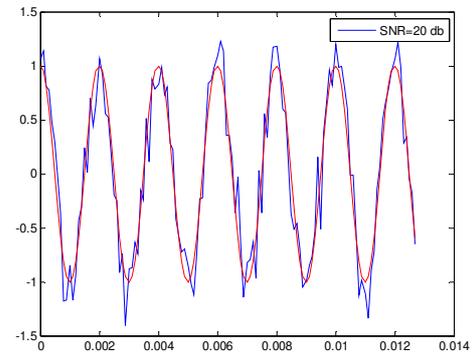
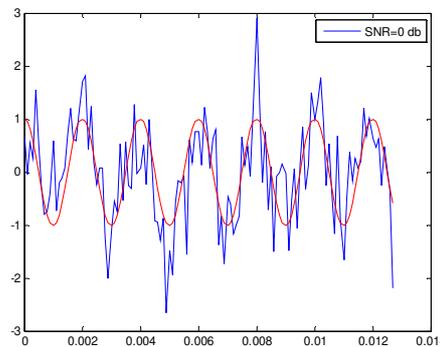


La corrélation est largement utilisée dans les systèmes radar. Ainsi, pour détecter un avion, on envoie une impulsion, puis on reçoit une version retardée, atténuée et bruitée de cette impulsion. L'intercorrélation du signal reçu et émis présentera un pic à l'instant correspondant au retard.



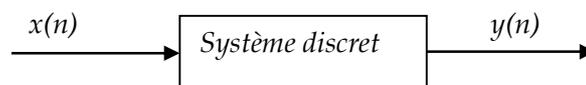
Remarques : La notion de bruit est relative, elle dépend du contexte. Le rapport signal/bruit désigne la qualité de la transmission d'une information par rapport aux parasites. Il est défini par:

$$SNR_{db} = 20 \text{ Log}(P_S / P_B)$$

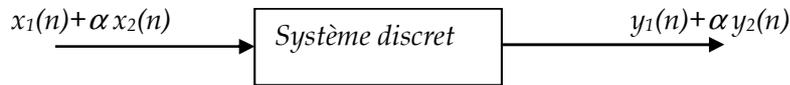


## 2. Théorie des systèmes linéaires et invariants discrets (SLID)

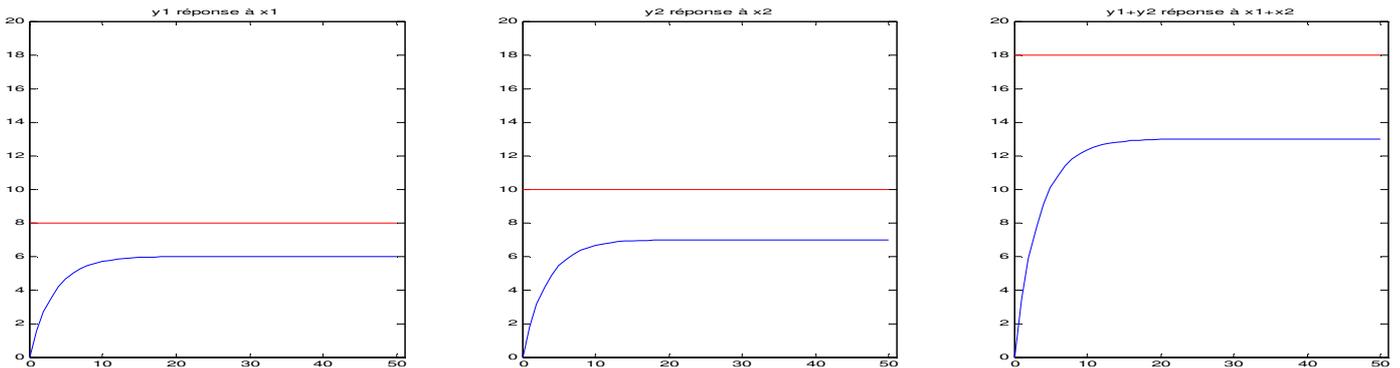
Un système linéaire est un modèle de système qui applique un opérateur linéaire à un signal d'entrée. C'est une abstraction mathématique très utile en automatique, traitement du signal, mécanique et télécommunications. Les systèmes linéaires sont ainsi fréquemment utilisés pour décrire un système non linéaire en ignorant les petites non-linéarités. Un système est *discret*, si à la suite d'entrée discrète  $x(n)$  correspond une suite de sortie discrète  $y(n)$ .



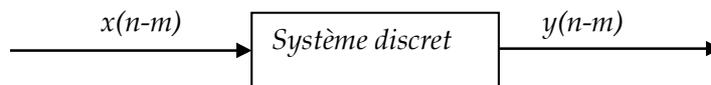
- Si l'entrée  $x(n)$  produit une sortie  $y(n)$ , quand on applique une entrée  $k.x(n)$ , la sortie sera  $k.y(n)$ . Si deux entrées  $x_1(n)$  et  $x_2(n)$  engendrent deux sorties  $y_1(n)$  et  $y_2(n)$  alors  $x_1(n) + x_2(n)$  engendrera  $y_1(n) + y_2(n)$



Exemple



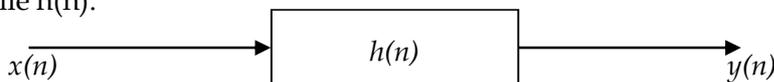
- S'il y a invariance dans le temps, une translation de l'entrée ( $x(n) \Rightarrow x(n-m)$ ) se traduira par une même translation dans le temps de la sortie ( $y(n) \Rightarrow y(n-m)$ ).



Si le système est invariant, cela implique que le système réagit de la même façon quel que soit l'instant auquel nous appliquons ses excitations. Cette propriété exprime que la caractéristique du système ne dépend pas de l'origine du temps, on parle encore de stationnarité.

Convolution

Si les hypothèses de linéarité et d'invariance temporelle sont vérifiées, on peut caractériser le système par sa réponse impulsionnelle  $h(n)$ .



On peut en déduire l'effet d'une entrée quelconque sous la forme d'une convolution. Cette dernière est l'opération de traitement de signal la plus fondamentale. Elle indique que la valeur du signal de sortie à l'instant  $n$  est obtenue par la sommation (intégrale) pondérée des valeurs passées du signal d'excitation  $x(n)$ . La fonction de pondération est précisément la réponse impulsionnelle  $h(n)$ :

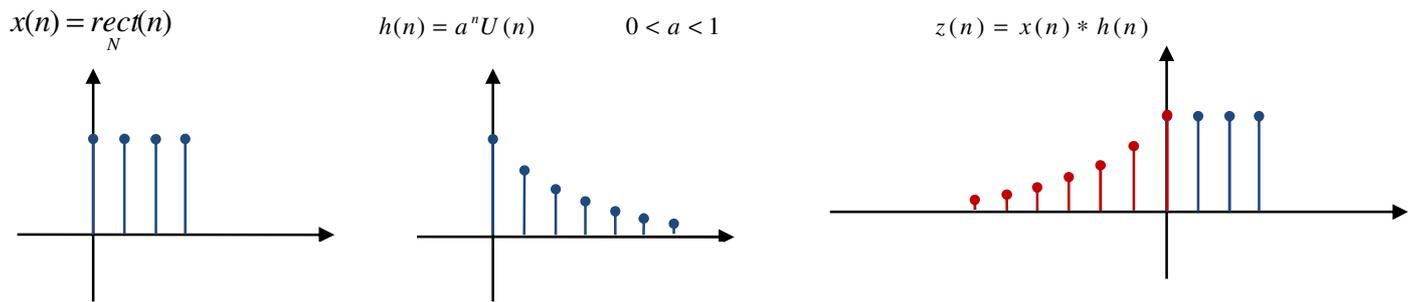
$$y(n) = h(n) * x(n) = \sum_{m=-\infty}^{\infty} h(m)x(n-m) = \sum_{m=-\infty}^{\infty} x(m)h(n-m)$$

La réponse impulsionnelle  $h(n)$  est le signal qu'on obtient en sortie  $y(n)=h(n)$  si on applique en entrée une impulsion "de Dirac"  $x(n)=\delta(n)$ . Le Dirac est l'élément neutre de l'opération de convolution:

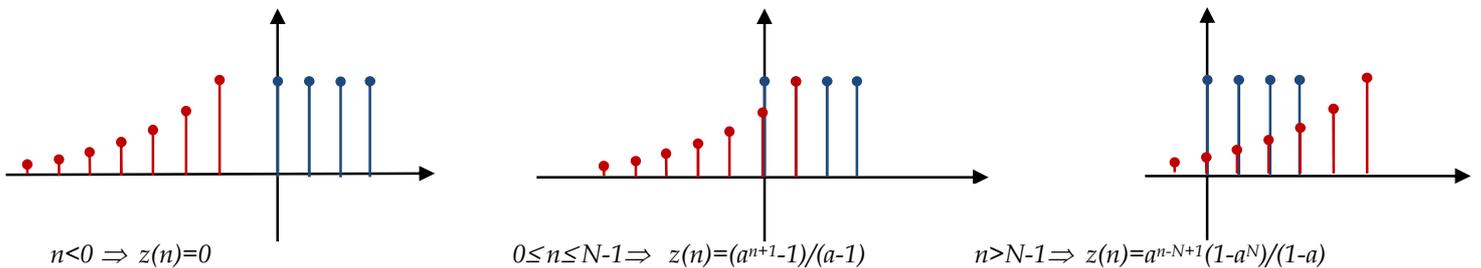
$$\delta(n) * x(n) = x(n)$$

Le calcul de la convolution consiste donc à calculer la somme du produit  $x(m)h(n-m)$ . Le signal  $h(n-m)$  est simplement le signal initial  $h(m)$ , retourné dans le temps pour donner  $h(-m)$  puis translaté de  $n$ . En calculant alors l'ensemble des produits obtenus en faisant « glisser »  $h$ , c'est-à-dire pour tous les décalages de  $n$ , on obtient le produit de convolution pour tout  $n$ .

Exemple 1:



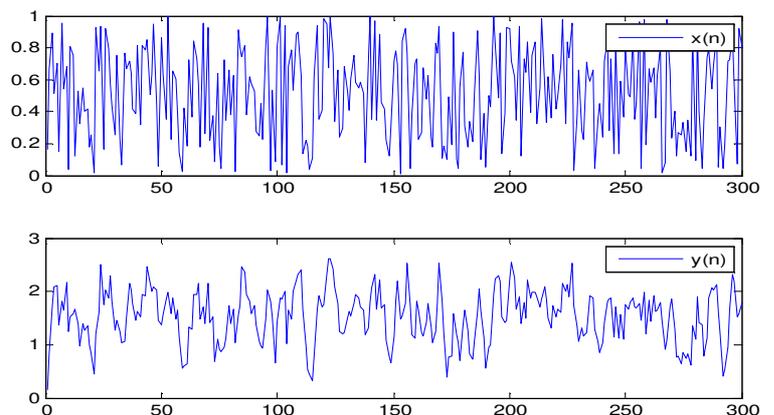
On distingue 3 cas :



Exemple 2 :

$$h(n) = \frac{1}{N+1} \prod_{N+1}(n) \Rightarrow y(n) = \frac{1}{N+1} \sum_{m=-N/2}^{N/2} x(n+m)$$

(Voir TP n°0)



Exemple 3: Soit le signal  $x(n) = \{2, -1, 3\}$  et  $h(n) = \{1, 2, 2, 3\}$

Calculer  $y(n)=x(n)*h(n)$

$h[n]$	=	1	2	2	3		
$x[n]$	=	2	-1	3			
		2	4	4	6		
			-1	-2	-2	-3	
				3	6	6	9
		2	3	5	10	3	9

Pour des séquences finies, on peut utiliser la méthode des colonnes

$$Y(n) = \{2, 3, 5, 10, 3, 9\}$$

Remarques:

- Les opérations de corrélation et convolution sont liées. Mathématiquement, on peut écrire une relation qui permet d'exprimer la fonction de corrélation comme un produit de convolution (et réciproquement).

$$\text{En effet: } R_{xy}(k) = x(n) * y^*(-n)$$

On peut donc considérer l'opération d'un SLI comme une mesure de la corrélation entre deux signaux ( $x^*(-n)$  et  $h(n)$ ). En fait, le signal de sortie est "construit" à partir des composantes fréquentielles communes au signal d'entrée et à la réponse impulsionnelle.

- Si on applique à un SLIT une entrée sinusoïdale réelle ou complexe de fréquence  $f_0$ , alors, la sortie sera une sinusoïde dont l'amplitude et la phase pourront être modifiées mais qui conservera la même forme (une sinusoïde) et la même fréquence  $f_0$ . On dit que les sinusoïdes sont les fonctions propres des SLIT.

- Un système linéaire invariant est un système dont le comportement dans le temps, peut-être décrit par une

$$\text{équation aux différences : } \sum_{i=0}^M a_i y(n-i) = \sum_{i=0}^N b_i x(n-i),$$

Stabilité et causalité d'un SLID

Une contrainte importante pour la formalisation de nombreux problèmes est de respecter la notion de *causalité* (les effets ne peuvent pas précéder la cause). Dans le cas des SLIT, cette causalité se traduit par le fait que pour:  $h(n) = 0$  pour  $n < 0$ .

$$x(n) = 0, n < n_0 \text{ alors } y(n) = 0, n < n_0 \quad \Rightarrow \quad h(n) = 0, n < 0, y(n) = \sum_{m=-\infty}^n x(m)h(n-m) = \sum_{m=0}^{+\infty} x(n-m)h(m),$$

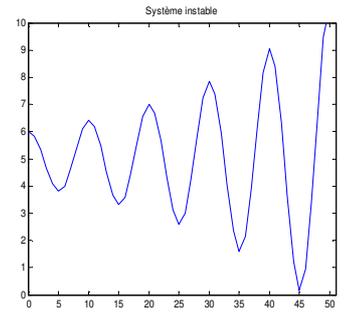
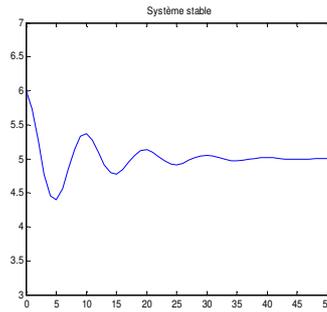
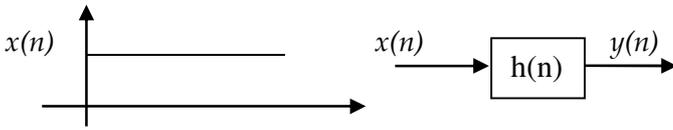
$$\text{- si h et x sont causaux } y(n) = \sum_{m=0}^n h(n-m)x(m)$$

Remarque:

Nous pouvons envisager mémoriser les signaux d'entrée et faire un traitement de ceux-ci en temps différé, les systèmes utilisés ne sont plus alors nécessairement causaux car pour élaborer la sortie à l'instant  $n_i$ , nous disposons en mémoire des entrées aux instants suivants. C'est souvent le cas en traitement d'image, en traitement de parole effectué après mémorisation du signal à traiter.

Une autre notion fondamentale est la *stabilité* des systèmes. La propriété de stabilité des systèmes bouclés est non seulement une performance mais une exigence pour le bon fonctionnement d'une boucle

d'asservissement ou de régulation. Une boucle instable est une boucle inutilisable. La définition la plus courante de cette stabilité est la suivante :



On dit qu'un système est stable si, en lui appliquant une entrée bornée quelconque, la sortie reste bornée, ce qui implique dans le cas des SLIT:  $\sum_n |h(n)| < \infty$

3. Filtrés à réponses impulsionnelles finies ou infinies

- Si les  $a_i$  sont  $\neq$  de 0, le système est dit récursif (RII), il est non récursif s'il ne dépend que des  $x(n-i)$  (RIF)
- Si le système est à réponse impulsionnelle de durée finie (RIF), alors :  $y(n) = \sum_{m=0}^K h(m)x(n - m)$

Dans ce cas, le système numérique est une fenêtre centrée sur les K plus récents échantillons.

- Si le système est à réponse impulsionnelle de durée infinie (RII) :  $y(n) = \sum_{m=0}^{+\infty} h(m)x(n - m)$

Dans ce cas, il est nécessaire de connaître tous les échantillons présents et passés, le système à une mémoire de longueur infinie.

Exemple 1  $y(n)=x(n)+a_1x(n-1)+a_2x(n-2)+.....+a_kx(n-k)$  est l'équation aux différences finies d'un filtre RIF

avec comme réponse impulsionnelle  $h(n)=\delta(n)+a_1\delta(n-1)+a_2\delta(n-2)+.....+a_k\delta(n-k)$  qui, on peut le constater, est bien finie.

Exemple 2  $y(n)=x(n)+a_1y(n-1)$  est l'équation aux différences finies d'un filtre RII

avec  $y(n-1)=x(n-1)+a_1y(n-2) \Rightarrow y(n)=x(n)+a_1x(n-1)+a_1^2y(n-2)$

de même  $y(n-2)=x(n-2)+a_1y(n-3) \Rightarrow y(n)=x(n)+a_1x(n-1)+a_1^2x(n-2))+a_1^3y(n-3)$

$\Rightarrow y(n)=x(n)+a_1x(n-1)+a_1^2x(n-2))+a_1^3x(n-3)+a_1^4x(n-3)+.....+ a_1^m y(n-m)$

En poursuivant le procédé à l'infini  $y(n)$  dépend d'une infinité de  $x(n-k)$  ce qui en fait un filtre RII.

Exemple d'application

Les séquences  $x(n)$  (réel) et  $y(n)$  représentent respectivement l'entrée et la sortie d'un système discret. Pour chaque cas, identifiez celles représentant

- a) des systèmes linéaires, b) des systèmes causals, c) des systèmes invariants aux translations de  $n$ ,
- d) des systèmes assurément ou possiblement stables (en fonctions des constantes)

1.  $y(n) = x(n) + bx(n-1)$       2.  $y(n) = x(n) + bx(n+1)$       6.  $y(n) = b^{x(n)}$      $b$  : constante réelle  
 3.  $y(n) = nx(n)$       5.  $y(n) = x(n)e^n$       7.  $y(n) = |x(n)|$   
 4.  $y(n) = x(n) \sin(2\pi f_0 n)$        $N$  : constante entière

- a) Tous sauf 6 et 7    b) Tous sauf 2  
 c) Les systèmes 1, 2, 6 et 7 d) 1 (b finie), 2 (b finie), 4, 6 (b finie) et 7.

**4. Rappels sur la Transformée de Fourier**

La transformée de Fourier est une technique mathématique permettant de déterminer le spectre de fréquences d'un signal (par exemple un son). La transformation de Fourier correspond à un changement de base dans l'espace des fonctions de carré sommable. La définition mathématique est la suivante :

$$TF\{x(t)\} = X(f) = \int_{-\infty}^{+\infty} x(t)e^{-2\pi j f t} dt \quad \text{et} \quad x(t) = TF^{-1}\{X(f)\} = \int_{-\infty}^{+\infty} X(f)e^{2\pi j f t} df$$

$x(t)$  et  $X(f)$  sont deux descriptions équivalentes du même signal. Ainsi, tous les signaux à énergie finie possède une transformée de Fourier. Cette dernière est une fonction complexe même si  $x(t)$  est réel

Si  $X(f)$  = fonction réel  $\Leftrightarrow x(t)$  est *paire*      Si  $X(f)$  = fonction imaginaire pure  $\Leftrightarrow x(t)$  est *impaire*

Remarques :

$X(f)$  pourra être exprimée sous la forme:

$$\left\{ \begin{aligned} |X(f)| &= \sqrt{A^2(f) + B^2(f)} \quad \text{et} \quad \varphi = \arg(X(f)) = \arctg \frac{B(f)}{A(f)} \end{aligned} \right.$$

Où  $|X(f)|$  et  $\varphi$  sont respectivement le module et la phase de  $X(f)$ .

La TF d'un signal périodique est divergente, mais on peut définir une TF au sens des distributions en utilisant la décomposition en Série de Fourier. Le résultat correspond à un spectre de raies (non continu):

sachant que:  $C_n = \frac{1}{T} \int_{-T/2}^{T/2} x(t) \exp(-2\pi j n f_0 t) dt$     et     $X(f) = \int_{-\infty}^{+\infty} x(t) e^{-2\pi j f t} dt$     alors  $X(f) = \lim_{T \rightarrow +\infty} (T \cdot C_n)$

Pour les signaux à énergie finie, la TF conserve l'énergie (relation de Parseval) :  $E_x = \int_{-\infty}^{+\infty} |x(t)|^2 dt = \int_{-\infty}^{+\infty} |X(f)|^2 df$

On peut donc définir une notion d'énergie par unité de fréquence, la densité spectrale d'énergie (DSE). La DES est la TF de l'autocorrélation (Théorème de Wiener-Kintchine)

$$S_x(f) = |X(f)|^2 = \int_{-\infty}^{+\infty} R_x(\tau) e^{-2\pi j f \tau} d\tau$$

Pour les signaux à puissance moyenne finie, on définit alors une densité spectrale de puissance (DSP):

$$P_x(f) = \lim_{T \rightarrow \infty} \frac{|X(f)|^2}{T}$$

- La propriété de changement d'échelle indique que plus le support temporel d'une fonction est étroit plus le support de sa TF est large.

- La translation d'un signal temporel se traduit par un déphasage en fréquence. Une translation en fréquence équivaut à une modulation temporelle.

- La propriété de dualité permet d'obtenir facilement de nouvelles paires de transformées de FOURIER à partir des paires déjà connues.

Exemple d'Application

Soient  $x(t) = A \Pi_{\theta}(t)$  et  $y(t) = A \Lambda_{\theta/2}(t)$  signal porte et signal triangulaire d'amplitude  $A > 0$  et de largeur  $\theta$ .

- Tracer  $x(t)$  et  $y(t)$  sur le même graphe
- Utiliser les dérivées pour trouver  $X(f)$  et  $Y(f)$  qui seront représentés sur le même graphe puis commenter et interpréter les deux graphes pour  $A=1$  et  $\theta=20$ .
- Soit  $z(t) = \cos(2\pi f_0 t)$ , tracer  $Z(f)$ ,  $Z_1(f)$  la TF de  $x(t).z(t)$  et  $Z_2(f)$  la TF de  $y(t).z(t)$  sur le même graphe et comparer  $Z(f)$  et  $Z_1(f)$  puis  $Z(f)$  et  $Z_2(f)$ .

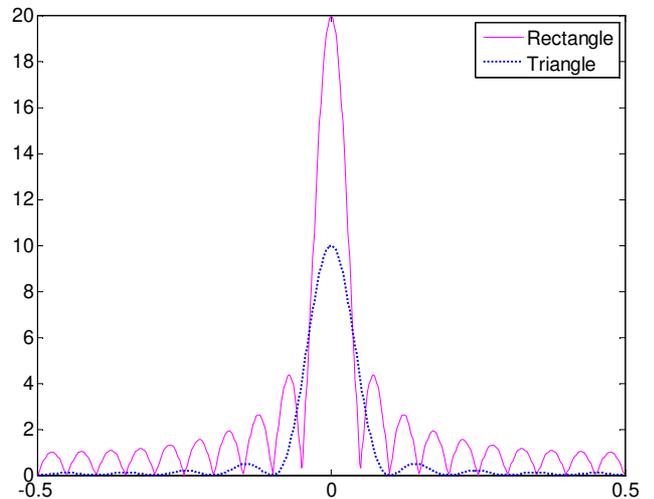
Solution

$$X(f) = A\theta \operatorname{sinc}(f\theta) \quad Y(f) = A \frac{\theta}{2} \operatorname{sinc}^2(f\theta/2)$$

$$Z(f) = \frac{1}{2} \delta(f - f_0) + \frac{1}{2} \delta(f + f_0)$$

$$Z_1(f) = \frac{A}{2} \theta \operatorname{sinc}(\theta(f - f_0)) + \frac{A}{2} \theta \operatorname{sinc}(\theta(f + f_0))$$

$$Z_2(f) = \frac{A}{4} \theta \operatorname{sinc}^2(\theta(f - f_0)) + \frac{A}{4} \theta \operatorname{sinc}^2(\theta(f + f_0))$$



Exercices d'application

1. Soit le signal  $x(t) = \sum_{n=-\infty}^{+\infty} \delta(t - nT)$ , montrer que :  $\sum_{n=-\infty}^{+\infty} e^{j2\pi n t/T} = T \sum_{n=-\infty}^{+\infty} \delta(t - nT)$  et déterminer la TF de  $x(t)$

Solution  $\sum_{n=-\infty}^{+\infty} \delta(t - nT) \xrightarrow{TF} X(f) = \frac{1}{T} \sum_{n=-\infty}^{+\infty} \delta(f - nf_0)$

$$2. h(t) = \begin{cases} 2 + \cos(2\pi t) & |t| \leq 1/2 \\ 0 & \text{ailleurs} \end{cases}$$

- Déterminer  $H(f)$ , la transformée de Fourier de  $h(t)$
- Esquisser  $H(f)$  pour l'intervalle  $0 \leq f \leq 3$  et Donner la valeur numérique de  $H(f)$  pour  $f = 2.5$

Solution  $H(f) = 2 \operatorname{sinc}(f) + \frac{1}{2} \operatorname{sinc}(f - 1) + \frac{1}{2} \operatorname{sinc}(f + 1)$

Principales propriétés de la TF

- o Linéarité :  $ax_1(t) + bx_2(t) \xrightarrow{TF} aX_1(f) + bX_2(f)$
- o Décalage temporel :  $x(t - t_0) \xrightarrow{TF} X(f)e^{-2\pi j f t_0}$
- o Décalage fréquentiel :  $x(t)e^{2\pi j f_0 t} \xrightarrow{TF} X(f - f_0)$  (MA)
- o Dualité temps-fréq :  $x(t) \xrightarrow{TF} X(f) \Rightarrow X(t) \xrightarrow{TF} x(-f)$
- o Changement d'échelle :  $x(at) \xrightarrow{TF} \frac{1}{|a|} X(f/a)$
- o Dérivation :  $\frac{d^n x(t)}{dt^n} \xrightarrow{TF} (2\pi j f)^n X(f)$
- o Inversion et conjugaison :  $x(-t) \xrightarrow{TF} X(-f)$   
 $x^*(t) \xrightarrow{TF} X^*(-f)$
- o Convolution :  $x(t) * h(t) \xrightarrow{TF} X(f)H(f)$

TF au sens des distributions

Pour les signaux à puissance moyenne finie (Dirac, Echelon, signaux périodiques, etc.), on peut définir une TF au sens des distributions.

- o Dirac :  $\delta(t - t_0) \xrightarrow{TF} e^{-2\pi j f t_0} \Rightarrow \delta(t) \xrightarrow{TF} 1$
- o Echelon et signe :  $U(t) \xrightarrow{TF} \frac{1}{2\pi j f} + \frac{1}{2} \delta(f) \quad \text{Sgn}(f) = \frac{1}{\pi j f}$
- o Périodiques :  $\sum_{n=-\infty}^{+\infty} C_n \exp(2\pi j n f_0 t) \xrightarrow{TF} \sum_{n=-\infty}^{+\infty} C_n \delta(f - n f_0)$
- o Peigne de Dirac :  $\sum_{n=-\infty}^{+\infty} \delta(t - nT) \xrightarrow{TF} X(f) = \frac{1}{T} \sum_{n=-\infty}^{+\infty} \delta(f - n f_0)$
- o  $\cos(2\pi f_0 t) \xrightarrow{TF} \frac{1}{2} \delta(f - f_0) + \frac{1}{2} \delta(f + f_0)$
- o  $\sin(2\pi f_0 t) \xrightarrow{TF} \frac{1}{2j} \delta(f - f_0) - \frac{1}{2j} \delta(f + f_0)$

Espace temporel	Espace fréquentiel
	 Re{X(f)} est une fonction paire Im{X(f)} est une fonction impaire
$x(t) = ax_1(t) + bx_2(t)$	$X(f) = aX_1(f) + bX_2(f)$
	 $e^{-j2\pi a f} X(f)$
$\frac{dx(t)}{dt}$	$j2\pi f X(f)$
$x(t) = \cos(2\pi f_0 t)$	
$x(t) \cos(2\pi f_0 t)$	$X(f - f_0) + X(f + f_0)$
$x(t) = T_0 \sum_{k=-\infty}^{+\infty} \delta(t - kT_0)$	$X(f) = \sum_{n=-\infty}^{+\infty} \delta(f - n f_0)$
$y(t) = \int_{u=-\infty}^{u=+\infty} h(u)x(t-u)du$ $= x(t) * h(t)$	$Y(f) = H(f)X(f)$
$y(t) = h(t)x(t)$	$Y(f) = H(f) * X(f)$

Série 0

1. Représenter les signaux suivants pour les cas continus et discrets

$$\prod_N(n-1), n.U(n), (n-2).U(n-3), (-n+3)U(n-2)U(-n+3), e^{-an}.U(n-1)$$

2. Donner l'expression du signal échelon U(n) en fonction du signal signe Sgn(n).

3. Les signaux suivants sont-ils à énergie finie, à puissance moyenne finie, ou ni l'un, ni l'autre ? Calculer, dans chaque cas, l'énergie totale et la puissance moyenne totale (a>0).

- Arect(n/N+1)      Asin(2πf<sub>0</sub>n)      Asin(2πf<sub>0</sub>n).U(n)      U(n)
- n.U(n)      Ae<sup>-anu(n)</sup>      Ae<sup>-an</sup>

4. Calculer l'autocorrélation des signaux suivants pour les cas continus et discrets

- Arect(n/N)      Asin(2πf<sub>0</sub>n)      5 δ<sub>N</sub>(n)      Bcos(2πf<sub>0</sub>n)

5. Calculer la sortie y(n) lorsque : x(n) = δ(n-n<sub>0</sub>) + δ(n-n<sub>1</sub>) et h(n) = e<sup>-an</sup>

6. Soit le signal échelon f(n) = E<sub>0</sub> U(n), d'amplitude E<sub>0</sub>.

- Représenter graphiquement et calculer le produit de convolution de f(n) par lui-même (auto-convolution).
- Faire de même pour le cas continu

7. Soit le signal x(n) = e<sup>-an</sup> U(n) transmis à travers le système h(n) dont la réponse impulsionnelle est h(n) est donnée par h(0) = h(1) = h(2) = 1/3.

- Calculer l'énergie et la puissance du signal d'entrée
- Calculer et tracer l'autocorrélation de h(n)
- Déterminer le signal y(n) résultant de la convolution numérique x(n)\*h(n).

8. Etudier la linéarité, la causalité, l'invariance et la stabilité des systèmes définis par les équations aux différences finies:

	Linéarité	Invariance	Causalité	Stabilité
y(n) = 2 .n. x(n-1)				
y(n+1) = 2 x(n+1) <sup>2</sup>				
y(n) =  x(n-1) - 0.5x(n+1)				
y(n) = 1/e <sup>n</sup> . x(n)				

Solutions

2. U(n) = 1/2(sgn(n)+1)

3. E = A<sup>2</sup>.(N+1) P<sub>m</sub> = 0, E = ∞ P<sub>m</sub> = A<sup>2</sup>/2, E = ∞ P<sub>m</sub> = A<sup>2</sup>/4, E = ∞ P<sub>m</sub> = 1/2  
 E = ∞ P<sub>m</sub> = ∞, E = A<sup>2</sup>/(1-e<sup>-2a</sup>) P<sub>m</sub> = 0, E = ∞ P<sub>m</sub> = ∞, E = 2A<sup>2</sup>N(1+N) P<sub>m</sub> = 0

4. A<sup>2</sup>N.Λ<sub>N</sub>(k)      A<sup>2</sup>/2.cos(2πf<sub>0</sub>k)      25δ<sub>N</sub>(k)      B<sup>2</sup>/2.cos(2πf<sub>0</sub>k)

5. x(n) = e<sup>-a(n-n<sub>0</sub>)</sup> + e<sup>-a(n-n<sub>1</sub>)</sup> 6. f(n)\*f(n) = E<sub>0</sub><sup>2</sup>.(n+1) pour n ≥ 0 et 0 ailleurs

7. E = 1/(1-e<sup>-2a</sup>) P = 0 R<sub>h</sub>(0) = 1/3, R<sub>h</sub>(±1) = 2/9, R<sub>h</sub>(±2) = 1/9 y(n) = 1/3 (x(n)+x(n-1)+x(n-2))

8. O-N-O-N      N-O-O-N      N-O-N-O      O-N-O-N

**Exercices supplémentaires**

1. Donner l'expression du signal  $x(n) = \text{Arect}[(n-n_0)/(N+1)] = A \prod_{N+1}(n-n_0)$  à l'aide du signal signe seulement.

Justifier graphiquement la solution trouvée (N supposé pair).

2. Soient  $x_1(n) = e^{-a.n} .U(n)$   $x_2(n) = e^{-b.n} .U(n)$  Calculer  $x_1(n) * x_2(n)$  avec  $(a, b) \in \mathfrak{R}^+$  et  $a > b$

3. Calculer et esquisser graphiquement pour les cas  $n_0 < n_1$  et  $n_0 > n_1$  le produit de convolution  $z(n) = x(n)*y(n)$  pour les cas suivants :

$X(n) = A[\delta(n+n_0) + \delta(n-n_0)]$  et  $Y(n) = B \delta(n) + \frac{1}{2}B[\delta(n+n_1) + \delta(n-n_1)]$

4. La fonction triangulaire est définie de la manière suivante:  $E^2 N . \Lambda_N(n) = \begin{cases} E^2 (n+1+N) & -N \leq n \leq 0 \\ -E^2 (n-1-N) & 0 \leq n \leq N \\ 0 & \text{ailleurs} \end{cases}$

- Vérifier analytiquement et graphiquement la relation  $E^2 N . \Lambda_N(n) = E . \Pi_{N+1}(n) * E . \Pi_{N+1}(n)$ ,

- En déduire l'auto-corrélation du signal et son énergie (devoir à rendre)

5. Soient  $x(n)$  et  $h(n)$  deux signaux numériques provenant respectivement de l'échantillonnage d'un signal  $x$  et de la réponse impulsionnelle  $h$  d'un système :

$x(n) = \{0, 0, 0, 0.5, 1.5, 0.5, 1.5, 0, 0, 0\}$  et  $h(n) = \{0.5, 0.5\}$

- Calculer l'énergie de chaque signal

- Calculer et tracer l'autocorrélation de  $h(n)$

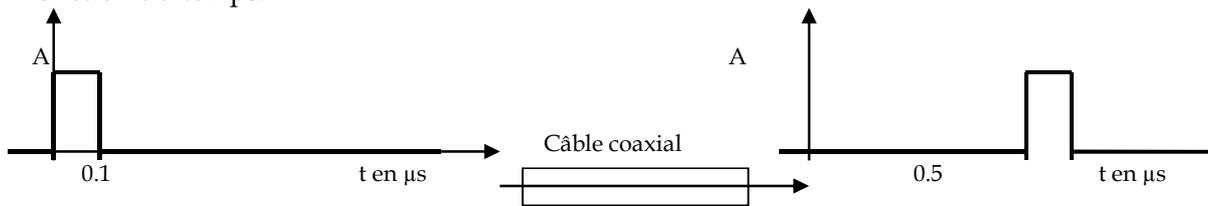
- Calculer la séquence  $y$  résultant de la convolution numérique  $x(n)*h(n)$ .

- Interpréter ces résultats du point de vue des plages de fréquences éliminées et conservées.

- Quel est le signal d'entrée qui permettrait de connaître le signal  $h(n)$  ?

- Proposer un signal  $h(n)$  permettant de réaliser un filtrage passe haut du signal  $x(n)$

6. On désire déterminer la vitesse de propagation d'une impulsion dans un câble coaxial sans perte. L'intensité est alors une fonction de la longueur de câble traversé 'l' et du temps 't'. Les mesures relevées donnent l'évolution de cette intensité pour  $l=0$  correspondant au signal en entrée du câble et pour une longueur  $l = 100m$  en fonction du temps.



- Ecrire les 2 signaux sous forme de 2 portes de mêmes amplitude et largeur

- Calculer la fonction de corrélation entre les 2 signaux.

- Montrer que la détection du maximum de la fonction de corrélation permet de calculer la vitesse de propagation dans le câble. Calculer cette vitesse.

**Solutions**

1.  $x(n) = A/2 [\text{sgn}(n-(n_0-N/2)) - \text{sgn}(n-(n_0+N/2+1))]$  2.  $x(n) = \begin{cases} 0 & n < 0 \\ e^{-bn} \frac{1-e^{(b-a)n+1}}{1-e^{(b-a)}} & n > 0 \end{cases}$

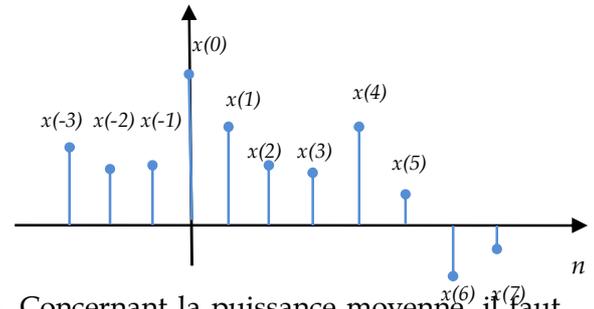
3.  $X(n)*Y_1(n) = AB[\delta(n+n_0) + \delta(n-n_0)] + AB/2[\delta(n+n_0+n_1) + \delta(n-n_0-n_1) + \delta(n-n_0+n_1) + \delta(n-n_0-n_1)]$

5. Voir interro1 S1 2016/2017                      6. Voir examen S1 2017/2018

**TP n°0 : Analyse temporelle des SLID (Sous Matlab)  
Convolution, Energie, Puissance et Corrélation**

**I. Rappels**

Un signal discret  $s(n)$  est une suite de  $N$  échantillons régulièrement espacés de  $T_e$  secondes:  $x(0), x(T_e), x(2T_e), \dots, x((N-1)T_e)$  où  $F_e=1/T_e$  est la fréquence d'échantillonnage. Le tracé graphique d'un signal discrétisé en temps peut s'effectuer simplement à l'aide de la fonction stem sous matlab.



- L'énergie d'un signal  $x(n)$  est fournie sous matlab par **sum(x.^2)**. Concernant la puissance moyenne, il faut diviser l'énergie par le nombre d'éléments de  $x(n)$ .
- Pour la corrélation et la convolution, on utilisera, respectivement, les fonctions **xcorr** et **conv**. A noter que la convolution ou la corrélation de  $x$  et  $h$  de durée respective  $N$  et  $M$  est un signal  $y(n)$  de durée  $(N+M-1)$
- La fonction **b=m+ s\*randn(N,1)** permet de générer un vecteur bruit  $b$  de distribution pseudo normale (Gaussienne) de taille  $N$  de moyenne  $m$  et de variance  $s^2$  dont la puissance est  $P_s = m^2 + s^2$ .

**II. Exemples à tester avant le TP**

**1.** Le programme suivant permet de générer un Dirac en 0 :  $\delta(n) = 1$  pour  $n=0$  et vaut 0 ailleurs

```
clc ; clear all ; close all ;
t=-10:10;
x=[zeros(1,10),1,zeros(1,10)];
stem(t,x);
axis([-10 10 -0.5 1.5]);
title('Dirac');
xlabel('n');ylabel('Amplitude');
```

**2.** Le programme suivant permet de générer un échelon  $U(n)=1$  pour  $n \geq 0$  et 0 pour  $n < 0$

```
clc ; clear all ; close all ;
t=-20:20;
x=[zeros(1,20),ones(1,21)];
stem(t,x);
title('Echelon unite');
xlabel('n');ylabel('Amplitude');
```

**3.** Pour générer  $N=128$  échantillons d'une sinusoïde de fréquence  $f_0=1000$ , on peut procéder de la façon suivante, choisir une fréquence d'échantillonnage :  $F_e = 8000$  (le pas de temps  $T_e=1/F_e$ ) Créer le vecteur des temps :  $t = (0:N-1)T_e$ . Calculer les échantillons:  $x = \cos(2*\pi*t*f_0)$  ; Puis, regarder le résultat : `plot(x)` ou `plot(t,x)`.

Ce qui nous donne :

```
clc ; clear all ; close all ;
N=128; fo=1000; Fe=8000; Te=1/Fe;
t=(0:N-1)*Te; x=cos(2*pi*fo*t);
plot(t,x) ;
figure; plot(x);
figure; stem(t,x);
```

### III. Programmes à réaliser

1. Le programme suivant permet de créer une porte de largeur 2s, centrée en 3 s, d'amplitude 4, échantillonnée avec  $T_e=0.1s$  avec  $N=50$  et de calculer son auto-corrélation et son énergie.

```
clc;clear all; close all;
Te=0.1; N=50; A=4;
t=(0:1:N-1)*Te;
porte=A*[zeros(1,15),ones(1,20),zeros(1,15)];
subplot(2,2,1);plot(t,porte);
subplot(2,2,2);stem(t,porte);
y1= xcorr(porte); tt=(-N+1:1:N-1)*Te;
subplot(2,2,3);plot(tt,y1);
Energie1=sum(porte.^2)
```

- Quelle est la différence entre  $\text{plot}(x)$  et  $\text{plot}(t,x)$ ?
- Quelle est la différence entre  $\text{plot}$  et  $\text{stem}$ ?
- Utiliser le workspace pour visualiser la taille et le contenu des vecteurs  $t$ ,  $tt$ ,  $\text{porte}$  et  $y$ . Commenter
- Calculer l'auto-corrélation théorique et comparer avec  $y1$ .
- Retrouve-t-on les propriétés de l'auto-corrélation?
- Calculer l'énergie théorique et comparer avec Energie 1

2. Commenter le programme suivant :

```
Clear all; close all; clc;
N=500; x=zeros(N,1);x(1:10)=1;
figure; plot(x);axis([0 N 0 1.2]);legend('signal émis');
y=circshift(x,50);
y=y+0.4*randn(N,1);
figure; plot(y);legend('signal reçu: bruité et retardé de 50s')
z=xcorr(y,x);z=z(N:end);
figure; plot(z);
```

- Induire un autre retard et observer.
- Changer la puissance du bruit et commenter.
- Donner un exemple d'application de ce programme.

3. Soit le programme ci-dessous

```
clearall; close all; clc;
N=300; T=3;Te=1;
X=rand(N,1); t_x=(0:N-1)*Te;
h=(1/T)*ones(T,1);t_h=(0:T-1)*Te;
subplot(3,1,1); plot(t_x,X); subplot(3,1,2); stem(t_h,h,'r');
Y=conv(X,h); t_y=(1-T:N-1)*Te;
subplot(3,1,3); plot(t_y,Y); axis([0 N min(Y) max(Y)]);
```

- Commenter le programme. Changer la valeur de T et observer.
- Comment nomme-t-on le signal h? Quel est son rôle?

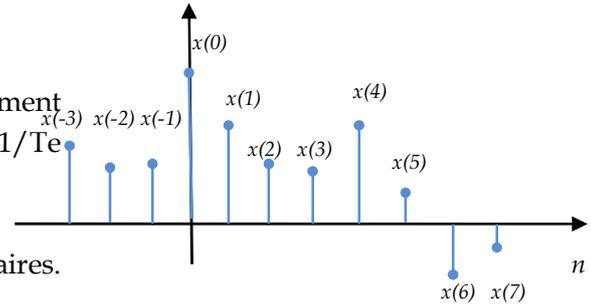
4. Générer et visualiser le signal  $x$  composé de 50 échantillons d'une sinusoïde de fréquence  $f_0 = 0.1$  avec  $f_e = 10.f_0$

- Calculer et afficher son auto-corrélation et comparer avec l'auto-corrélation théorique.
- Retrouver les caractéristiques du signal (puissance et fréquence) à partir de l'auto-corrélation.
- Générer le signal  $z=x+b$  ou  $b$  est un bruit avec ( $m=0$ ,  $s=0.5$ ). Calculer l'auto-corrélation du bruit et commenter.
- Calculer et visualiser l'auto-corrélation de  $z$  pour  $s=0.5$ ,  $s=1$  puis  $s=2$  en commentant.
- Retrouver dans chacun des cas précédents la fréquence du signal à partir de l'auto-corrélation de  $z$ .

**TP n°0 : Analyse temporelle des SLID (sous Python)  
Convolution, Energie, Puissance et Corrélation**

**I. Rappels**

Un signal discret  $s(n)$  est une suite de  $N$  échantillons régulièrement espacés de  $T_e$  secondes:  $x(0), x(T_e), x(2T_e), \dots, x((N-1)T_e)$  où  $F_e = 1/T_e$  est la fréquence d'échantillonnage.



Au début du programme python, importer les bibliothèques nécessaires.  
Pour alléger l'écriture leur donner un nom plus court:

```
import numpy as np
import matplotlib.pyplot as plt
```

**NumPy** est une bibliothèque destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.

**SciPy** regroupe un ensemble de bibliothèques Python à usage scientifique avec un environnement très similaire à Matlab. Elle comporte des modules pour l'optimisation, l'algèbre linéaire, les statistiques, le traitement du signal ou encore le traitement d'images. **SciPy** utilise les tableaux et matrices du module **NumPy**.

**II. Exemples à tester avant le TP**

1. Le programme suivant permet de générer un Dirac en 0 :  $\delta(n) = 1$  pour  $n=0$  et vaut 0 ailleurs

```
import numpy as np; import matplotlib.pyplot as plt
N=32;
x = np.zeros(N); x[0]=1;
plt.figure(1)
plt.stem(x)
plt.title('Un Dirac. '); plt.grid(True); plt.xlabel('n'); plt.ylabel('Amplitude'); plt.show()
```

2. Rajouter les lignes suivante pour générer un échelon  $U(n)=1$  pour  $n \geq 0$  et 0 pour  $n < 0$

```
x=np.zeros(N); y = np.ones(N); z=np.concatenate((x,y));
plt.figure(2); plt.stem(z)
plt.title('Un Echelon'); plt.grid(True); plt.xlabel('n'); plt.ylabel('Amplitude'); plt.show()
```

3. Générer une sinusoïde de fréquence  $f_0=1000$  et une fréquence d'échantillonnage :  $F_e = 1200$  ( $T_e=1/F_e$ )

```
import numpy as np; import matplotlib.pyplot as plt
N = 128; f0=1000; fe=1200.; Te=1/fe
t = np.linspace(0.0, (N-1)*Te, N) ; x = np.cos(2.0*np.pi*f0*t)
plt.figure(1)
plt.subplot(211); plt.plot(t,x); plt.grid(True); plt.xlabel('Seconde(s)');
plt.ylabel('Amplitude')
plt.subplot(212); plt.stem(t, x); plt.grid(True); plt.xlabel('Seconde(s)');
plt.ylabel('Amplitude')
plt.show()
```

**III. Programmes à réaliser**

1. Le programme suivant permet de créer une porte de largeur 1s, centrée en 2,5 s, d'amplitude 4, échantillonnée avec  $T_e=0.1s$  avec  $N=50$  et de calculer son auto-corrélation et son énergie.

```
import numpy as np; import matplotlib.pyplot as plt
N = 50; A=4; Te=0.1; t = np.linspace(0, N*Te, N)
```

```
x = A*np.concatenate((np.zeros(20),np.ones(10),np.zeros(20)))
plt.figure(1)
plt.subplot(221); plt.plot(x); plt.grid(True); plt.xlabel('S'); plt.ylabel('Amp')
plt.subplot(222); plt.plot(t, x); plt.grid(True); plt.xlabel('S'); plt.ylabel('Amp')
plt.subplot(223); plt.stem(t, x); plt.grid(True); plt.xlabel('S'); plt.ylabel('Amp')
plt.show()
tt=np.linspace((1-N)*Te, (N-1)*Te, 2*N-1); Rx=np.correlate(x,x,mode='full');
plt.subplot(224); plt.title('Autocorr');plt.plot(tt, Rx); plt.grid(True); plt.xlabel('S');
plt.ylabel('Amp')
Energie=sum(abs(x)**2)
```

- Quelle est la différence entre plt.plot(x) et plt.plot(t,x)? et la différence entre plt.plot et plt.stem?
- Utiliser l'explorateur de variables pour visualiser la taille et le contenu des vecteurs t, tt, x et Rx.
- Calculer l'auto-corrélation théorique et comparer avec Rx.
- Retrouve-t-on les propriétés de l'auto-corrélation?
- Calculer l'énergie théorique et comparer avec Energie.

## 2. Commenter le programme suivant :

```
import numpy as np; import matplotlib.pyplot as plt
N = 500; sigma = 0.5 ; mu = 0;
x = np.zeros(N); x[0:9]=1; y = np.roll(x,50)+ sigma * np.random.randn(N) + mu
Ryx = np.correlate(y,x,mode='full');Ryx=Ryx[N-1:2*N-1]
plt.figure(1); plt.subplot(131); plt.plot(x); plt.title('signal émis')
plt.subplot(132); plt.plot(y); plt.title('signal reçu')
plt.subplot(133); plt.plot(Ryx); plt.title('Intercorrélation entre signal émis et signal reçu');
plt.show()
```

- Induire un autre retard et observer.
- Changer la puissance du bruit et commenter.
- Donner un exemple d'application de ce programme.

## 3. Soit le programme ci-dessous

```
import numpy as np;import matplotlib.pyplot as plt
N = 128; f0=500; fe=10000.; Te=1/fe; T=3; sigma = 0.3 ;
t = np.linspace(0.0, (N-1)*Te, N); x = np.cos(2.0*np.pi*f0*t)+ sigma * np.random.randn(N) ;
t_h = np.linspace(0.0, (T-1)*Te, T); h = 1.0/T*np.ones(T)
t_y = np.linspace(0.0, (N+T-2)*Te, N+T-1); y = np.convolve(x, h, mode='full')
plt.figure(1);
plt.subplot(311); plt.plot(t,x); plt.grid(True); plt.title('Signal d entrée x(n)')
plt.subplot(312); plt.stem(t_h, h); plt.grid(True); plt.title('Le filtre h(n)')
plt.subplot(313); plt.plot(t_y,y); plt.grid(True); plt.title('Signal de sortie y(n)')
plt.show()
```

- Commenter le programme. Changer la valeur de T et observer.
- Comment nomme-t-on le signal h? Quel est son rôle?

## 4. Générer et visualiser le signal $x$ composé de 64 échantillons d'une sinusoïde de fréquence $f_0 = 0.1$ avec $f_e = 20.f_0$

- Calculer et afficher son auto-corrélation et comparer avec l'auto-corrélation théorique.
- Retrouver les caractéristiques du signal (puissance et fréquence) à partir de l'auto-corrélation.
- Générer le signal  $z=x+b$  ou  $b$  est un bruit avec ( $m=0, s=0.5$ ). Calculer l'auto-corrélation du bruit et commenter.
- Calculer et visualiser l'auto-corrélation de  $z$  pour  $s=0.5, s=1$  puis  $s=2$  en commentant.
- Retrouver dans chacun des cas précédents la fréquence du signal à partir de l'auto-corrélation de  $z$ .

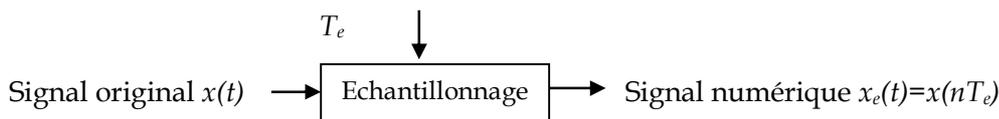
## I. Analyse fréquentielle des SLID

Dans la réalité, les signaux n'ont pas toujours une forme simple en raison de la nature de l'information qu'ils portent. Dans de tels cas, la représentation du signal en fonction de la fréquence est peut s'avérer très utile. Pour cela, on fait appel à la transformée de Fourier. Elle a pour but de mettre en évidence des caractéristiques du signal non évidentes dans la représentation temporelle : les propriétés fréquentielles (spectrales). L'utilisation de cette description fréquentielle permet de caractériser simplement les filtres linéaires, et faciliter leur étude.

Dans le but de calculer la transformée de Fourier  $X(f)$  d'un signal  $x(t)$  à l'aide d'un ordinateur, celui-ci n'ayant qu'un nombre limité de mots de taille finie, on est amené à discrétiser le signal (échantillonnage), à tronquer temporellement ce signal et à discrétiser sa transformée de Fourier [1].

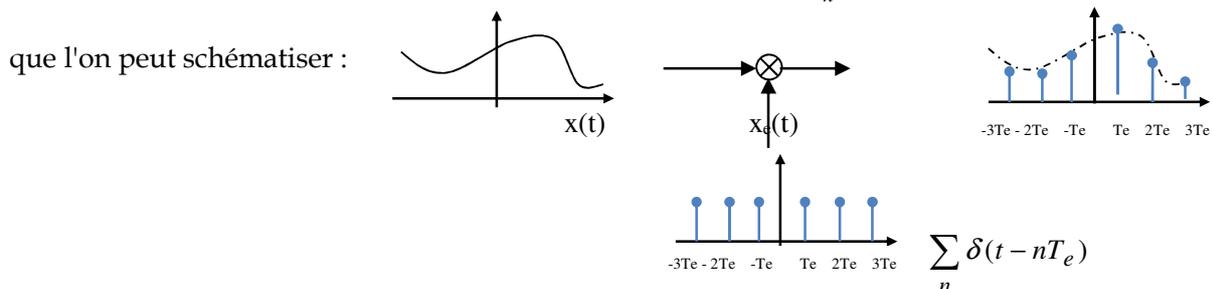
### 1. Quelques rappels sur l'échantillonnage idéal

L'échantillonnage est un élément important en traitement numérique de signaux. Il constitue la première opération à effectuer lors d'une conversion analogique à numérique (A/N). Il est conditionné, principalement, par deux contraintes majeures : Ne pas détériorer le signal (conserver l'information utile) tout en limitant l'espace mémoire nécessaire au stockage. On s'attachera dans une chaîne d'acquisition à minimiser cette valeur tout en ne détériorant pas le signal [3]. Ainsi, pour transformer un signal analogique en un signal numérique, on va donc prélever régulièrement des échantillons du signal analogique pour le rendre discret et permettre, ainsi, sa numérisation.



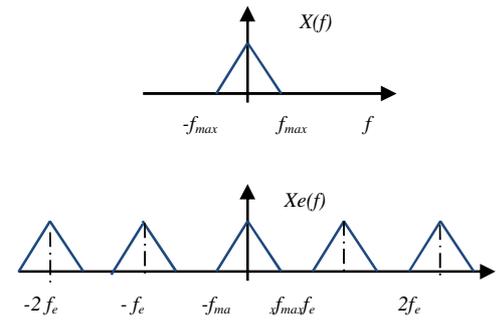
Soit  $x(t)$  un signal analogique de transformée de Fourier  $X(f)$ . Echantillonner le signal  $x(t)$  consiste à choisir une fréquence  $f_e$  et de construire un nouveau signal avec les  $x(nT_e)$  avec  $n$  un entier et  $T_e=1/f_e$ .

On peut écrire le signal échantillonné  $x_e(t)$  sous la forme :  $x_e(t) = \sum_n x(nT_e)\delta(t-nT_e)$



$$x_e(t) = x(t) \cdot \sum_n \delta(t - nT_e), \quad \Rightarrow X_e(f) = f_e \sum_n X(f - nf_e)$$

Cette expression montre que le spectre  $X_e(f)$  est périodique de période  $f_e$  et qu'il est la somme des répliques (copies) du spectre original  $X(f)$  décalées de  $nf_e$ . L'échantillonnage dans le domaine temporel se traduit par une "périodisation" de période  $f_e$  dans le domaine fréquentiel.



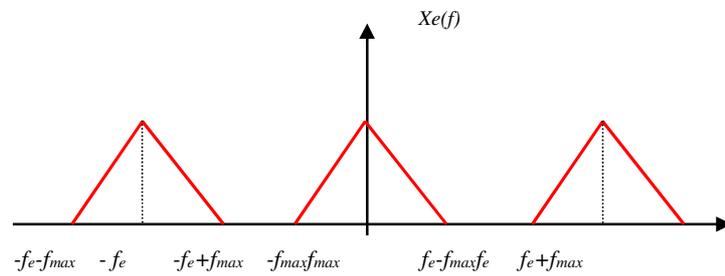
**Théorème de Shannon**

On considère que  $x(t)$  est un signal réel dont le spectre est borné en fréquence, de fréquence maximale  $f_{max}$  soit  $\forall |f| > f_{max} \quad X(f) = 0$

Deux cas de configurations :

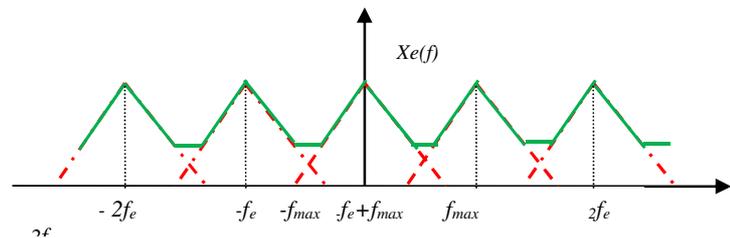
-  $f_e > 2 f_{max}$

Les motifs étant disjoints, on peut extraire  $X(f)$  grâce à un filtre passe-bas idéal et donc reconstituer parfaitement le signal  $x(t)$  à partir des  $x_e(t)$ .



-  $f_e < 2 f_{max}$

Les motifs élémentaires de  $|X_e(f)|$  se recouvrent (Repliement de spectres). Il n'est plus possible de récupérer le spectre  $X(f)$  par un filtrage approprié.



Remarques : Si le support du spectre  $X(f)$  n'est pas borné (s'étale sur l'axe réel) il y a un repliement du spectre des échantillons (aliasing), on ne peut pas isoler le spectre original à partir de celui des échantillons. Dans la pratique, on ne peut pas se contenter de prendre une fréquence d'échantillonnage égale à la fréquence de Nyquist ( $f_e/2 > f_{max}$ ), mais en prendre une supérieure car on ne peut réaliser un filtre passe-bas idéal avec une fréquence de coupure très nette. Par exemple, pour numériser la parole dans le réseau téléphonique, on utilise une fréquence d'échantillonnage 8kHz alors que le spectre de la voix est compris entre 300Hz et 3400Hz.

**Filtre anti-repliement** Les signaux étudiés en réalité sont rarement à support fréquentiel borné, c'est-à-dire que  $f_{max} = \infty$ . C'est par exemple le cas d'un signal rectangulaire périodique dont les raies fréquentielles s'étendent à l'infini ou encore un signal bruité. Ceci implique que quelle que soit la fréquence d'échantillonnage il y aura repliement de spectre puisque  $f_e > 2 f_{max} = \infty$  est une condition impossible à réaliser. Pour remédier à ce problème, on utilise à l'entrée d'un système numérique un filtre passe-bas appelé filtre anti-repliement ou anti-aliasing. Ce filtre est analogique, idéalisé il doit avoir un gain de 1 sur une bande de fréquence  $F_e$ , centrée en zéro. Son rôle va être de limiter le contenu spectral du signal à la partie utile. Il va participer aussi à limiter l'influence du bruit [3].

Exemple d'application

On échantillonne un signal sinusoïdal de fréquence 200Hz avec une fréquence d'échantillonnage  $f_e = 500\text{Hz}$  puis avec  $f_e = 300\text{Hz}$ . Quel signal obtient-on lors d'une reconstruction parfaite dans les deux cas ?

Solution

$$f_e=500\text{Hz} \Rightarrow x_r(t) = \cos(2\pi f_0 t) \text{ avec } f_0=200\text{Hz}$$

$$f_e=300\text{Hz} \Rightarrow x_r(t) = \cos(2\pi f_0 t) \text{ avec } f_0=100\text{Hz}$$

**2. Transformée de Fourier à temps discret (TFTD)**

Lorsque le signal à traiter n'est plus analogique mais numérique, la relation de la TF devient :

$$TF\{x(t)\} = X(f) = \int_{-\infty}^{+\infty} x(t) e^{-2\pi j f t} dt$$

$$\Rightarrow X_e(f) = \int_{-\infty}^{+\infty} x_e(t) e^{-2\pi j f t} dt = \int_{-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} x(nT_e) \delta(t - nT_e) e^{-2\pi j f t} dt = \sum_{n=-\infty}^{+\infty} x(nT_e) e^{-2\pi j f nT_e}$$

L'échantillonnage périodise le spectre du signal avec une période de répétition  $f_e$  ainsi  $X_e(f) = X_e(f + f_e)$ , par ailleurs, l'amplitude est multiplié par un facteur  $f_e$ . Sachant que tout signal périodique peut être décomposé en séries de Fourier, on a :

$$X_e(f) = \sum_{n=-\infty}^{+\infty} C_n e^{-2\pi j n T_e f} \quad \text{Avec} \quad x(nT_e) = C_n = \frac{1}{f_e} \int_{-f_e/2}^{f_e/2} X_e(f) e^{2\pi j n T_e f} df$$

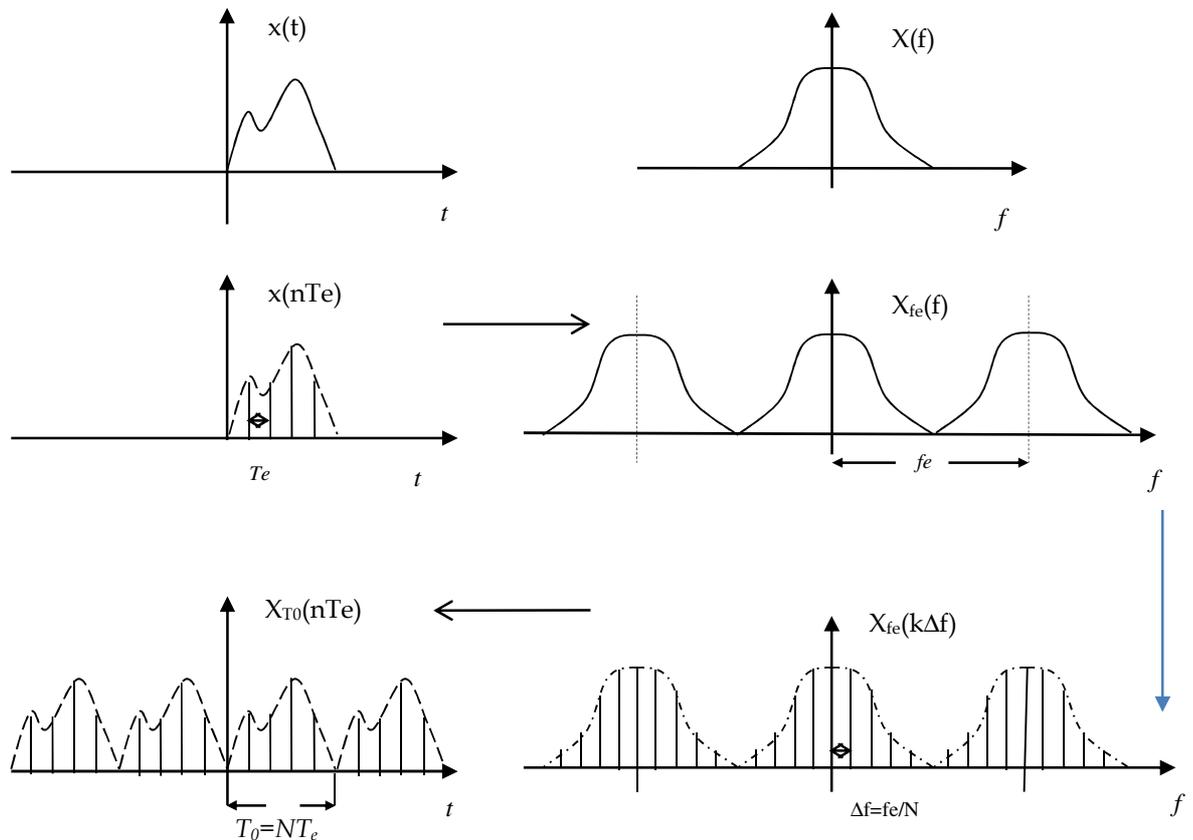
Cette transformée de Fourier appliquée aux signaux discrets est donc une fonction à **fréquence continue**, périodique de période  $f_e$ . Il est d'usage de la représenter sur un intervalle de longueur  $f_e$ , de  $-f_e/2$  à  $+f_e/2$ .

Cependant, si on veut calculer la TF d'un signal discret à l'aide d'un ordinateur, on se retrouve confronté aux problèmes suivants : Le calcul de la TF nécessite une infinité de points de mesures  $x(n)$  (pas toujours possible dans la pratique : contraintes temps réel, etc.). En outre, le calculateur ne peut calculer une TFTD car sa réponse fréquentielle est forcément discrète = un nombre fini de points fréquentiel alors que  $f$  varie continûment [5]. La solution est de limiter la durée de  $x(n)$  i.e. considérer un nombre fini  $N$  de points temporels et de discrétiser la fréquence (considérer un nombre fini  $L$  de points fréquentiels) d'où la TFD.

**3. Transformée de Fourier discrète (TFD)**

Cette transformée, popularisée par son calcul rapide (TFR ou FFT : Fast Fourier Transform), fait correspondre une suite de  $N$  valeurs à une autre de suite de  $N$  valeurs numériques également.

On considère un signal numérique  $s(n)$  défini par  $N$  échantillons temporels, obtenus par échantillonnage avec la période  $T_e$ . La numérisation du signal  $X(f)$  passe par l'échantillonnage de  $X(f)$ . On divise l'intervalle  $f_e$  par  $N$ , ainsi  $X(f)$  est échantillonné à la cadence  $\Delta f = f_e/N = 1/NT_e$ . Ce dernier résultat entraîne une périodicité du signal temporel de  $T_0 = 1/\Delta f = NT_e$ .



Sachant que  $f_e = N \cdot \Delta f$ ,  $f$  devient  $k \Delta f$ , les formules précédentes donnent:

$$X_e(k f_e / N) = \sum_{n=0}^{N-1} x(n) e^{-2\pi j n T_e k f_e / N} = \sum_{n=0}^{N-1} x(n) e^{-2\pi j n k / N} = X_e(k)$$

Et  $x(n) = \frac{1}{f_e} \sum_{k=-N/2}^{N/2-1} X_e(k) e^{2\pi j n k / N} \cdot \Delta f = \frac{1}{N} \sum_{k=-N/2}^{N/2-1} X_e(k) e^{2\pi j n k / N}$

$$\Rightarrow \begin{cases} X(k) = \sum_{n=0}^{N-1} x(n) e^{-2\pi j n k / N} \\ x(n) = \frac{1}{N} \sum_{k=-N/2}^{N/2-1} X(k) e^{2\pi j n k / N} \end{cases}$$

Ce qui nous permet d'obtenir la TFD et la TFD inverse :

$$\begin{cases} X(k) = \sum_{n=0}^{N-1} x(n) e^{-2\pi j n k / N} \\ x(n) = \frac{1}{N} \sum_{k=-N/2}^{N/2-1} X(k) e^{2\pi j n k / N} \end{cases}$$

On remarque aisément que l'on perd toute notion de temps relatif aux échantillons. Nous obtenons au final une relation entre une suite indexée par une variable entière  $n$  et une suite indexée par  $k$ . Les  $N$  termes  $X(k)$  correspondent à une approximation (à un facteur multiplicatif  $T_e$  près) de la transformée de Fourier de ce signal aux  $N$  points de fréquence  $k \Delta f$ , avec  $k$  entre 0 et  $N - 1$ , c'est-à-dire  $f$  entre 0 et  $f_e$ .

Exemple

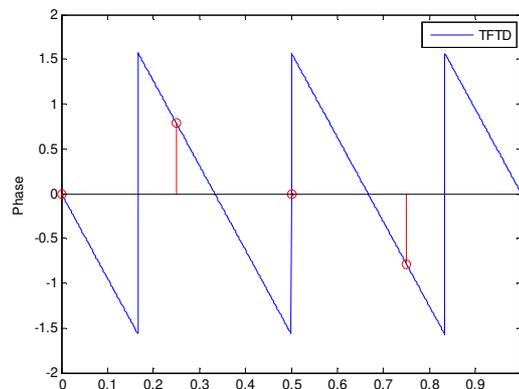
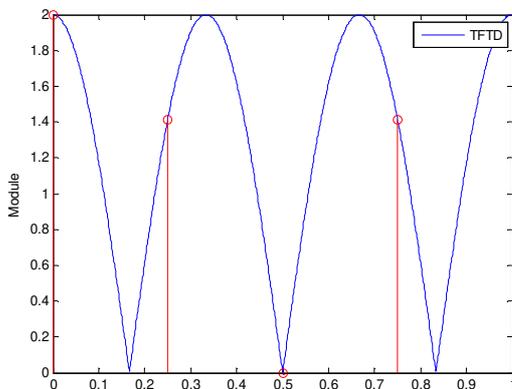
Soit le signal  $x(n) = 1$  pour  $n = 0$  et  $n = 3$  et 0 ailleurs.

- Calculons d'abord la TFTD, ce qui nous donne  $X(f) = \sum_{n=-\infty}^{+\infty} x(nT_e)e^{-2\pi j f n T_e} = 1 + e^{-6\pi j f T_e} = 2 \cos(3\pi f T_e) e^{-3\pi j f T_e}$

- Calculons maintenant la TFD sur  $N=4$  échantillons (4 échantillons de la TFD à partir de 4 échantillons du signal)

$$X(k) = \sum_{n=0}^3 x(n)e^{-2\pi j n k / 4} = \begin{cases} X(0) = \sum_{n=0}^3 x(n)e^0 = x(0) + x(1) + x(2) + x(3) = 2 \\ X(1) = \sum_{n=0}^3 x(n)e^{-\pi j n / 2} = x(0) + x(1)e^{-\pi j / 2} + x(2)e^{-\pi j} + x(3)e^{-3\pi j / 2} = 2 \cos(3\pi / 4) e^{-3\pi j / 4} \\ X(2) = \sum_{n=0}^3 x(n)e^{-\pi j n} = 2 \cos(3\pi / 2) e^{-3\pi j / 2} \\ X(3) = \sum_{n=0}^3 x(n)e^{-3\pi j n / 2} = 2 \cos(9\pi / 4) e^{-9\pi j / 4} \end{cases}$$

Modules des 4 échantillons de  $X(k)$ :  $2, \sqrt{2}, 0, \sqrt{2}$

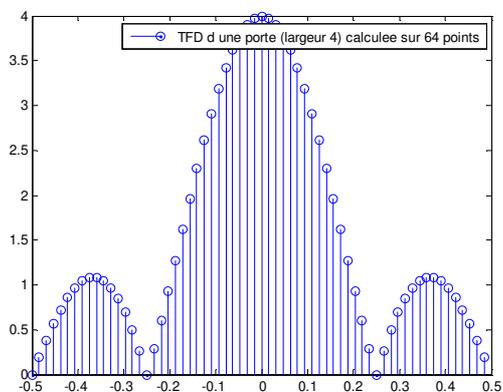
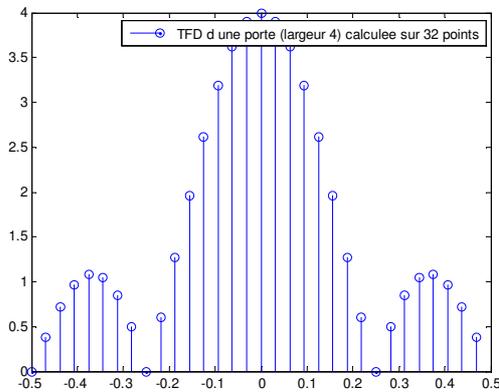
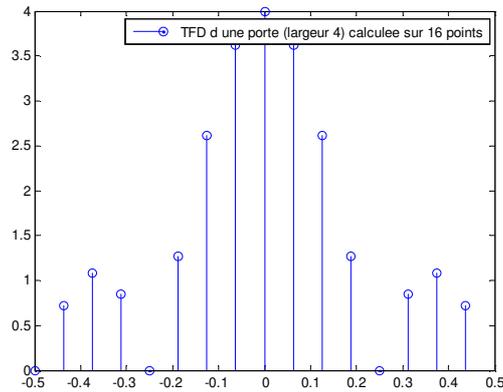
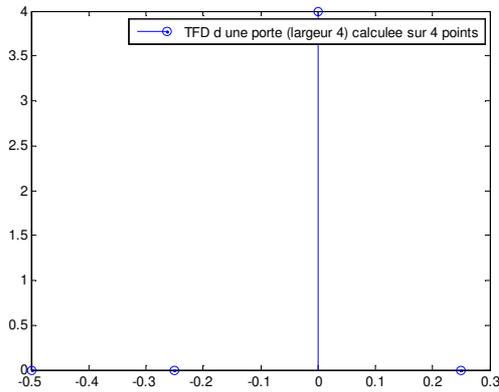


On peut observer que les quatre échantillons de la TFD (en rouge) se superposent à la courbe de la TFTD (en bleu). On confirme que la TFD n'est que l'échantillonnage de la TFTD limitée à  $N$ . On note en outre, que la précision fréquentielle est de  $\Delta f = f_q / N$ . Pour améliorer cette précision, il faudrait diminuer le pas en fréquence.

Remarque  $x(n)$  est périodique de période  $N$  et  $X(k)$  est aussi périodique de période  $N$  :

$$x(n + N) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j \frac{2\pi(n+N)k}{N}} = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j \frac{2\pi n k}{N}} e^{j \frac{2\pi N k}{N}} = x(n)$$

Sachant que  $x(n)$  et  $X(k)$  sont calculés sur le même nombre de points  $N$ , on peut augmenter la précision, par la technique du zéro-padding : on calcule la TFD sur un nombre  $NF$  pouvant être bien plus grand que le nombre de points  $N$  disponible du signal ( $NF \gg N$ ). La figure suivante en donne un exemple (Voir TP n°1).



Autres propriétés : Toutes les propriétés se déduisent des propriétés de la transformée continue en se rappelant que chaque signal manipulé, de durée finie, doit être considéré comme une période d'un signal périodique, et cela en temps et en fréquence. La conséquence en est que la translation d'un signal ( lors d'une convolution ou de corrélation) se traduit par un décalage circulaire [6]

o Linéarité :  $ax_1(n) + bx_2(n) \xrightarrow{TFD} aX_1(k) + bX_2(k)$

o Décalage temporel :  $x(n - m) \xrightarrow{TFD} X(k)e^{-\frac{2\pi j k m}{N}}$

o Décalage fréquentiel :  $x(n)e^{\frac{2\pi j k_0 n}{N}} \xrightarrow{TFD} X(k - k_0)$

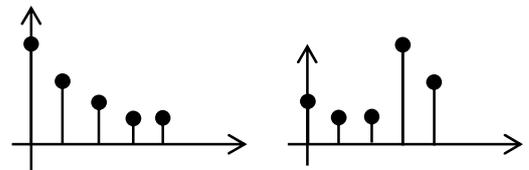
o Dualité temps-fréquence :  $x(n) \xrightarrow{TFD} X(k) \Rightarrow X(n) \xrightarrow{TF} N.x(-k)$

o Changement d'échelle :  $x(an) \xrightarrow{TFD} \frac{1}{|a|} X(k/a)$

o Convolution périodique :  $x(n) * h(n) \xrightarrow{TFD} X(k).H(k)$

o  $\delta(n) \xrightarrow{TFD} 1$        $\delta(n - m) \xrightarrow{TFD} e^{-2\pi j m k / N}$

o Par ailleurs, l'énergie se conserve:  $\sum_0^{N-1} |x(n)|^2 = \frac{1}{N} \sum_0^{N-1} |X(k)|^2$



4. Calcul de la FFT ou TFR

La TFD est restée un outil peu utilisé jusqu'à l'apparition d'algorithmes « rapides » permettant son calcul. Le plus connu est du à Cooley et Tuckey et date de 1965. Le calcul direct de la TFD sur N points nécessite N<sup>2</sup> opérations complexes. L'algorithme proposé réduit à Nlog2(N) le nombre d'opérations. Ainsi, pour N = 1024, le temps de calcul de l'algorithme rapide peut être 100 fois plus court que le calcul utilisant la formule de définition de la TFD. Pour en donner un exemple, prenons le cas de N=8, il faut calculer :

$$S_k = \sum_{n=0}^{N-1} s_n e^{-j \frac{2\pi nk}{N}} = \sum_{n=0}^7 s_n e^{-j \frac{2\pi nk}{8}}$$

Soit :  $W_N = \exp\left(-j \frac{2\pi}{N}\right)$  alors  $S_k = \sum_{n=0}^7 s_n W_N^{nk}$

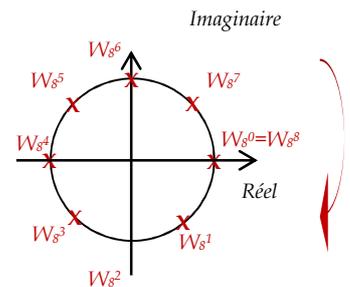
Pour N=8, explicitons la relation précédente :

$$\begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \end{bmatrix} = \begin{bmatrix} W_8^0 & W_8^0 \\ W_8^0 & W_8^1 & W_8^2 & W_8^3 & W_8^4 & W_8^5 & W_8^6 & W_8^7 \\ W_8^0 & W_8^2 & W_8^4 & W_8^6 & W_8^8 & W_8^{10} & W_8^{12} & W_8^{14} \\ W_8^0 & W_8^3 & W_8^6 & W_8^9 & W_8^{12} & W_8^{15} & W_8^{18} & W_8^{21} \\ W_8^0 & W_8^4 & W_8^8 & W_8^{12} & W_8^{16} & W_8^{20} & W_8^{24} & W_8^{28} \\ W_8^0 & W_8^5 & W_8^{10} & W_8^{15} & W_8^{20} & W_8^{25} & W_8^{30} & W_8^{35} \\ W_8^0 & W_8^6 & W_8^{12} & W_8^{18} & W_8^{24} & W_8^{30} & W_8^{36} & W_8^{42} \\ W_8^0 & W_8^7 & W_8^{14} & W_8^{21} & W_8^{28} & W_8^{35} & W_8^{42} & W_8^{49} \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{bmatrix}$$

Les facteurs  $W_n$  présentent un certain nombre de propriétés dont certaines sont mises à profit dans l'algorithme :

$$W_N^{nN} = 1, W_N^{N/2} = -1, W_N^{n+N} = W_N^n$$

$$\begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \end{bmatrix} = \begin{bmatrix} W_8^0 & W_8^0 \\ W_8^0 & W_8^1 & W_8^2 & W_8^3 & W_8^4 & W_8^5 & W_8^6 & W_8^7 \\ W_8^0 & W_8^2 & W_8^4 & W_8^6 & W_8^8 & W_8^{10} & W_8^{12} & W_8^{14} \\ W_8^0 & W_8^3 & W_8^6 & W_8^9 & W_8^{12} & W_8^{15} & W_8^{18} & W_8^{21} \\ W_8^0 & W_8^4 & W_8^8 & W_8^{12} & W_8^{16} & W_8^{20} & W_8^{24} & W_8^{28} \\ W_8^0 & W_8^5 & W_8^{10} & W_8^{15} & W_8^{20} & W_8^{25} & W_8^{30} & W_8^{35} \\ W_8^0 & W_8^6 & W_8^{12} & W_8^{18} & W_8^{24} & W_8^{30} & W_8^{36} & W_8^{42} \\ W_8^0 & W_8^7 & W_8^{14} & W_8^{21} & W_8^{28} & W_8^{35} & W_8^{42} & W_8^{49} \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{bmatrix}$$



L'algorithme suppose que N est pair : posons N=2 P. Introduisons les 2 sous-suites de  $s_n$  en fonction de la parité de n.

$$u_n = \{s_{2n}\}_{n=0, \dots, P-1}$$

$$v_n = \{s_{2n+1}\}_{n=0, \dots, P-1}$$

$$S_k = \sum_{n=0}^{N-1} s_n W_N^{nk} = \sum_{i=0}^{P-1} u_i W_{2P}^{2ik} + \sum_{i=0}^{P-1} v_i W_{2P}^{(2i+1)k}$$

$$S_k = \sum_{i=0}^{P-1} u_i W_P^{ik} + W_N^k \sum_{i=0}^{P-1} v_i W_P^{ik}$$

$$S_k = U_k + W_N^k V_k$$

On obtient ainsi :

Par ailleurs,

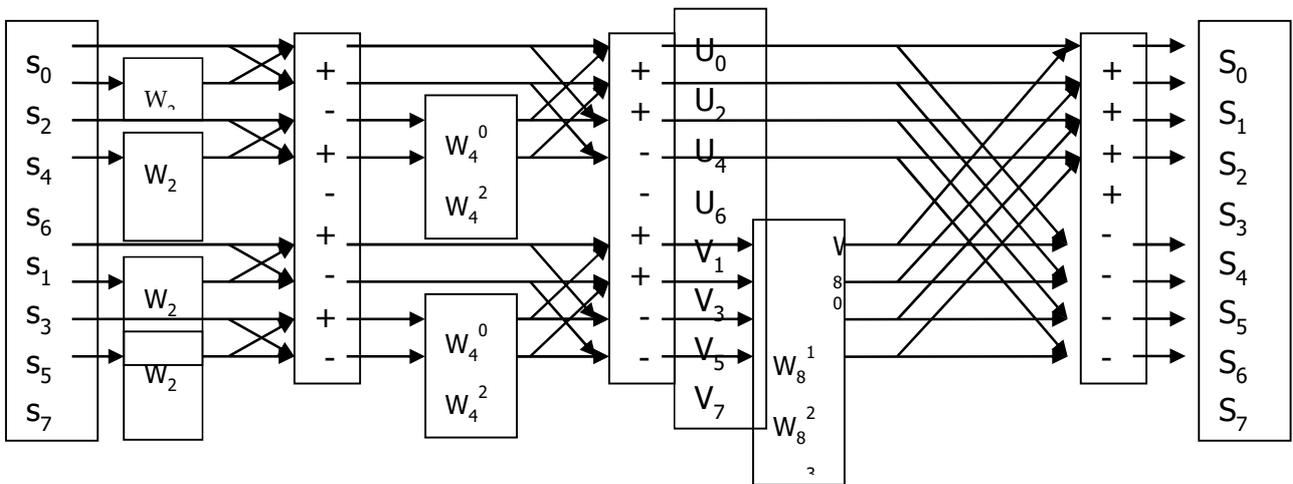
$$S_{(k+P)} = \sum_{n=0}^{N-1} s_n W_N^{(k+P)n} = \sum_{i=0}^{P-1} u_i W_{2P}^{2i(k+P)} + \sum_{i=0}^{P-1} v_i W_{2P}^{(2i+1)(k+P)}$$

$$S_{(k+P)} = \sum_{i=0}^{P-1} u_i W_P^{ik} W_P^{iP} + W_N^k W_N^P \sum_{i=0}^{P-1} v_i W_P^{ik} W_P^{iP}$$

$$S_{(k+P)} = \sum_{i=0}^{P-1} u_i W_P^{ik} \cdot 1 + W_N^k (-1) \sum_{i=0}^{P-1} v_i W_P^{ik} \cdot 1$$

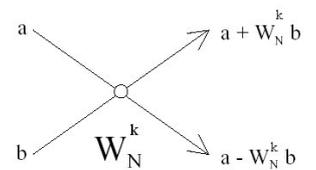
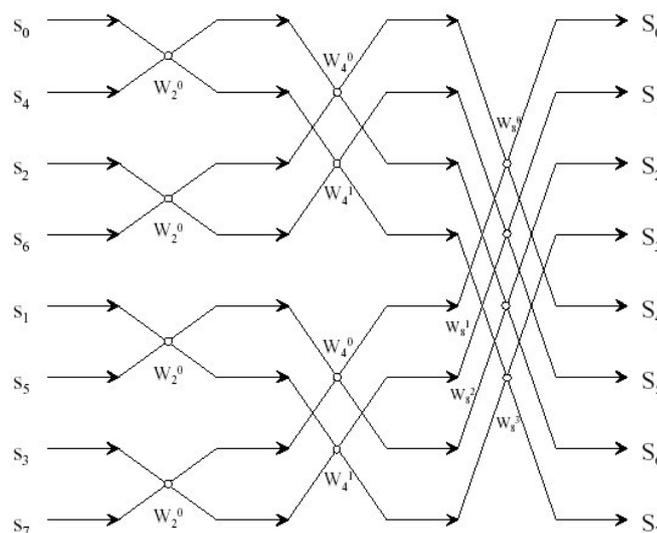
$$S_{(k+P)} = U_k - W_N^k V_k$$

Le calcul de la FFT revient donc à calculer  $U_k$  et  $V_k$  qui sont les TFD sur  $P$  points des suites de termes de rang pair et impair. On s'aperçoit qu'il ne reste qu'à exprimer les  $U_k$  et  $V_k$ .



Or, ce sont des TFD sur  $P$  points, qui peuvent subir le même schéma que précédemment à condition que  $P$  soit pair. On peut réitérer le processus à chaque sous-niveau si  $N$  est une puissance de 2 (Dans le cas contraire, on rajoute autant de 0 que nécessaire pour obtenir une puissance de 2).

Avec cette représentation, l'algorithme FFT se représente par :



L'algorithme papillon de FFT dit aussi à entrelacement temporel peut s'écrire sous forme matricielle. On obtient :

$$\begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \end{bmatrix} = \begin{bmatrix} 1 & W_8^0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -W_8^0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & W_8^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -W_8^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & W_8^1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -W_8^1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & W_8^3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -W_8^3 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & W_4^0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & W_4^0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -W_4^0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -W_4^0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & W_4^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & W_4^2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -W_4^2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -W_4^2 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 & W_2^0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & W_2^0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & W_2^0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & W_2^0 \\ 1 & 0 & 0 & 0 & -W_2^0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -W_2^0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -W_2^0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -W_2^0 \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{bmatrix}$$

En opérant de la même façon sur la TFD<sup>-1</sup>, on arrive à réduire considérablement le coût.

### 5. TFD des signaux de longueur illimitée

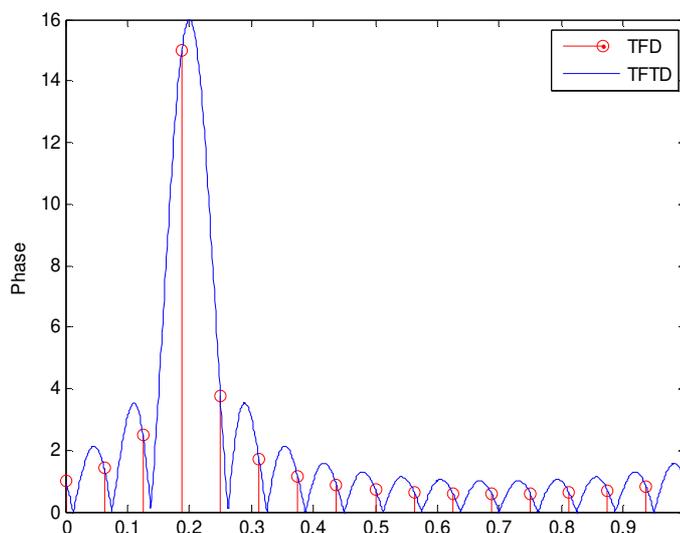
Le nombre d'éléments d'une séquence transformée par la TFD est implicitement limité, la fenêtre intrinsèque à la transformée discrète de Fourier est donc la fenêtre rectangulaire de durée  $T_0 = NT_e$ .

$$\left| TFTD \left\{ x(n) \cdot \text{Rect}(n) \right\}_{T_0 = NT_e} \right| \approx |X(f) * N \text{sin}c(fT_0)|$$

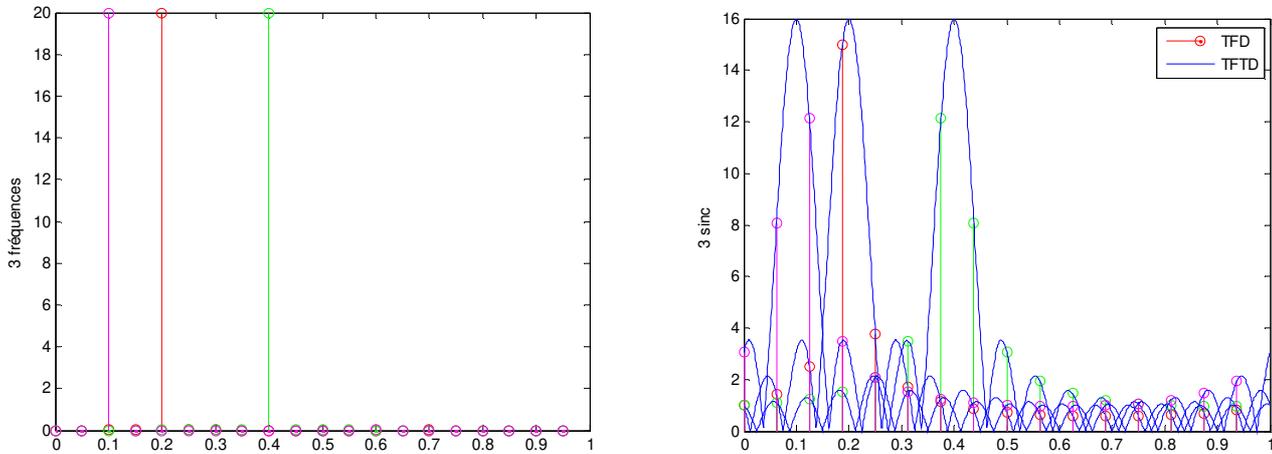
La troncation du signal échantillonné par une fenêtre de largeur  $T_0$  a pour effet de convoluer le spectre avec un sinus cardinal qui s'annule tous les  $1/T_0$  avec  $T_0 = NT_e$  soit tous les  $f_e/N$ .

#### Exemple :

En rouge est illustré  $X(k)$  le module de la TFD de  $\{x(n) = e^{2\pi j f_0 n}\}$ , pour  $n = \{0, \dots, N - 1\}$ , avec  $N = 16$  et  $f_0 = 0,2$ . En bleu  $X(f)$ , le module de la TTFD de  $x(n)$ . L'allure de  $X(f)$  fait apparaître un lobe principal de largeur  $2/N$  (ou  $2f_e/N$ ) autour de la fréquence  $f_0$  et des lobes secondaires de largeur  $1/N$  (ou  $f_e/N$ ).



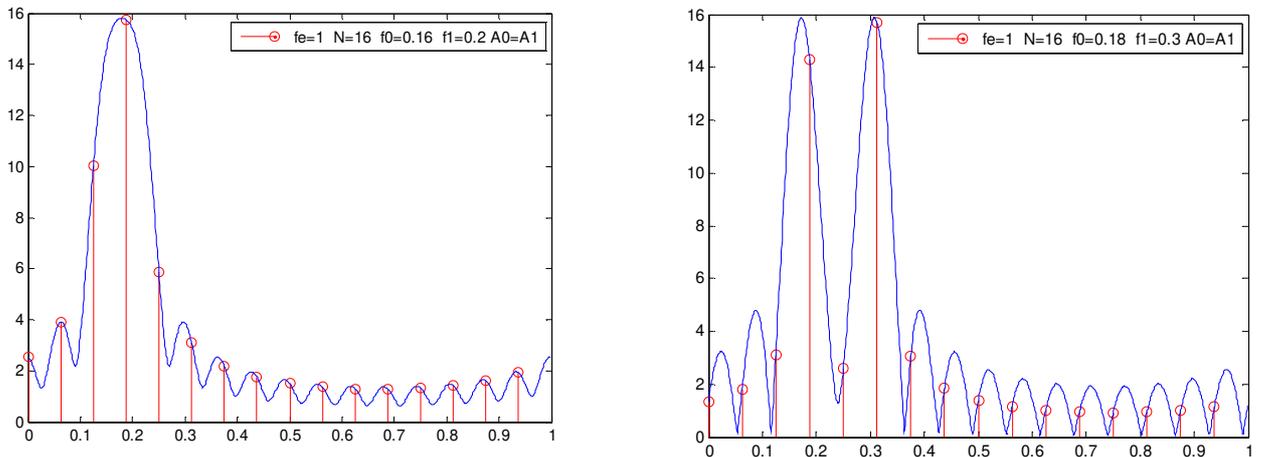
Dans le cas général, le spectre, obtenu par transformée de Fourier discrète, est donc un ensemble de fonctions sinc( $T_0 f$ ) centrées sur les fréquences qui composent le signal théorique initial.



Exemple:

Considérons l'exemple suivant:  $x(n) = A_0 e^{2\pi j f_0 n} + A_1 e^{2\pi j f_1 n}$  pour  $n (0:N-1)$  où  $f_0$  et  $f_1 \in [-f_e/2, f_e/2]$

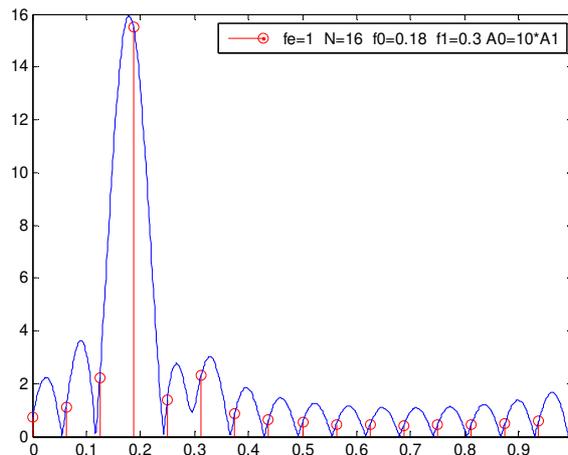
Le tracé du spectre par TFD montre que si l'écart en valeur absolue entre  $f_0$  et  $f_1$  est supérieur à  $f_e/N$ , il sera possible de distinguer les deux fréquences sur le tracé. Cette résolution en fréquence est liée au nombre de points  $N$  du signal (voir les 2 figure ci-dessous)



On considère maintenant l'exemple suivant (figure ci-dessus, à droite):

$$s(n) = A_0 e^{2\pi j f_0 n} + A_1 e^{2\pi j f_1 n} \quad \text{pour } n (0:N-1) \quad \text{où } A_0 > A_1$$

Un masquage du lobe principal de la composante en  $f_1$  peut survenir en raison des ondulations présentes dans le spectre de  $A_0 \exp(2\pi j f_0 n)$ . Une « fréquence » d'amplitude faible au voisinage d'une d'amplitude plus élevée sera masquée par le premier lobe secondaire.



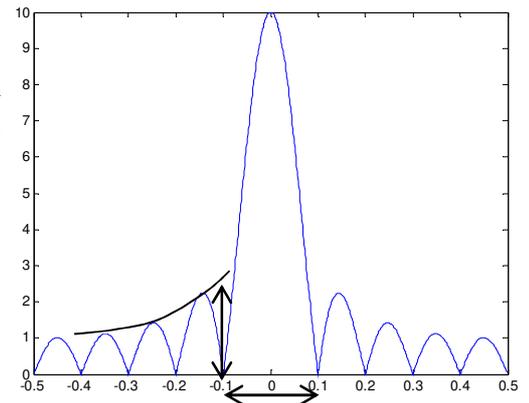
Ce phénomène se révélera gênant lorsque le spectre est composé de nombreuses raies, proches les unes des autres. L'importance des lobes secondaires peut être réduite par l'emploi d'autres fenêtres.

Mais cela se fera au détriment de la séparation de « fréquences » très voisines mais d'amplitude semblables car les 2 raies seront confondues dans un lobe principal plus large (la fenêtre rectangulaire possède le lobe principal le plus étroit).

6. Fenêtres de pondération

Lors de l'analyse spectrale d'un signal de longue durée, nous n'avons accès, en pratique, qu'à une portion limitée de ce signal. Le spectre obtenu correspond donc au spectre du signal à analyser auquel une *fenêtre* a été préalablement multipliée [7]. Pour ne pas altérer le spectre original, il faudrait que  $W(f)$  (spectre de la fenêtre) se rapproche le plus possible d'une distribution de Dirac. La distribution de Dirac étant l'élément neutre du produit de convolution. Il y a deux éléments importants pour se rapprocher de la distribution de Dirac. La finesse du lobe principale et la hauteur des lobes secondaires.

En effet, plus la largeur du pic principal est fine, plus la résolution est grande, c'est-à-dire que l'on peut séparer des raies proches. Et plus les lobes secondaires sont élevés plus on dégrade la forme du spectre et la détection d'un signal d'amplitude faible en présence d'un signal d'amplitude élevée sera ardue [8].

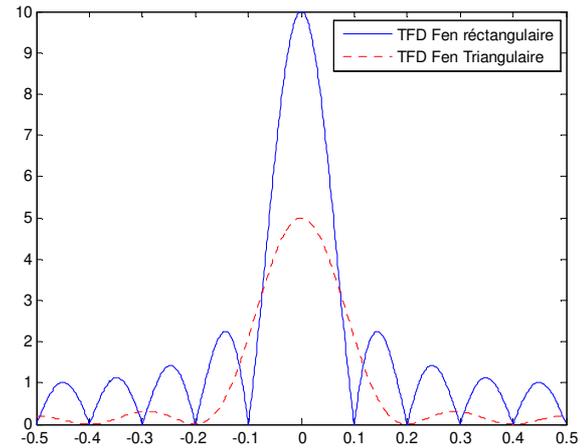


- Fenêtre Rectangulaire

Pour la fenêtre rectangulaire, on voit que la finesse du lobe principale ( $2 f_e/N$ ), peut être réglée par le nombre d'échantillons  $N$ . Ainsi, plus on observe le signal longtemps, plus la résolution du spectre augmente ce qui paraît logique. Par contre,  $\lambda_R$  varie très peu en fonction de  $N$  (-13dB) donc toujours une distorsion de spectre.

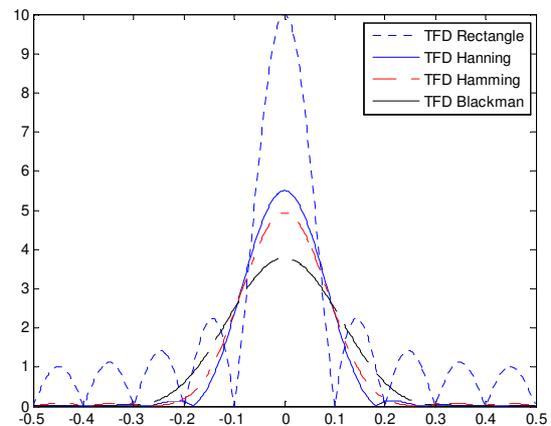
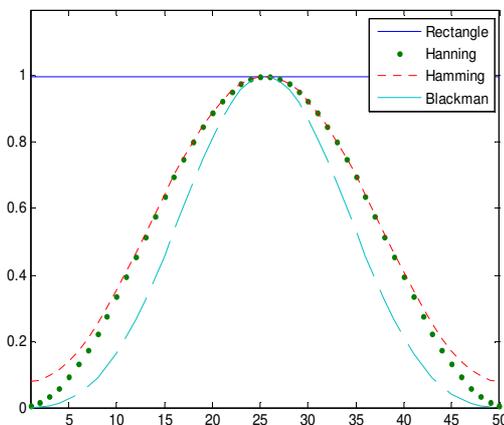
- Fenêtre Triangulaire

Pour obtenir la transformée de Fourier de la fenêtre triangulaire de largeur N, rappelons que la convolution de deux signaux rectangulaires donne un signal triangulaire. Ainsi, on peut exprimer cette fenêtre sous la forme d'une convolution de deux rectangles de largeur  $N/2$ . On observe une atténuation des lobes secondaires (-24dB) par rapport à la fenêtre rectangulaire[8]. Malheureusement, ceci se fait au prix de l'élargissement du pic central ( $4f_e/N$ ).



- Autres fenêtres

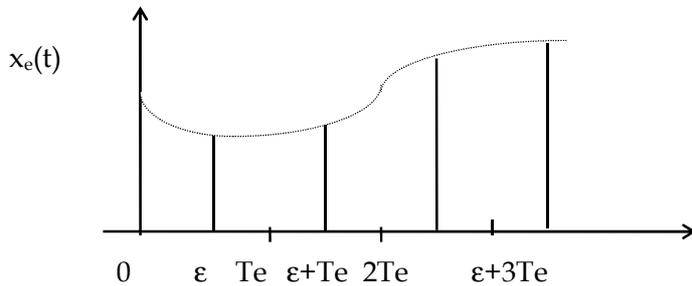
Fenêtres		Largeur de Transition : $\Delta f$	Lobe Principale/Secondaire
Rectangle	$w_{Rect}(n) = \begin{cases} 1 & \text{pour }  n  \leq \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases}$	$2 f_e / N$	-13 db
Triangulaire	$w_{Trian}(n) = \begin{cases} 1 - 2 n /(N-1) & \text{pour }  n  \leq \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases}$	$4 f_e / N$	- 25 db
Hanning	$w_{Han}(n) = \begin{cases} 0.5 + 0.5 \cos(\frac{2\pi n}{N-1}) & \text{pour }  n  \leq \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases}$	$4 f_e / N$	- 31 db
Hamming	$w_{Ham}(n) = \begin{cases} 0.54 + 0.46 \cos(\frac{2\pi n}{N-1}) & \text{pour }  n  \leq \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases}$	$4 f_e / N$	- 41 db
Blackman	$w_{Black}(n) = \begin{cases} 0.42 + 0.5 \cos(\frac{2\pi n}{N-1}) + 0.08 \cos(\frac{4\pi n}{N-1}) & \text{pour }  n  \leq \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases}$	$6 f_e / N$	- 57 db



Dans un problème d'analyse spectrale, on utilise généralement plusieurs fenêtres l'une après l'autre afin d'obtenir un bon compromis résolution/déformation.

Série n°1

1. On considère un signal de parole de durée 1mn et ayant une bande passante de 10 kHz échantillonne comme suit :

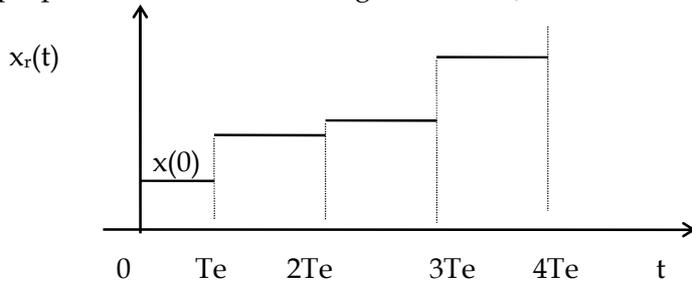


- Calculer le nombre minimal d'échantillons nécessaires pour représenter ce signal.
- Exprimer le signal échantillonné  $x_e(t)$  où  $\epsilon$  est un retard puis calculer sa TF

On suppose que l'échantillonnage se fait sans retard ( $\epsilon = 0$ )

- Exprimer le signal échantillonné et montrer que l'on peut reconstituer théoriquement le signal  $x(t)$  à partir des échantillons  $x(nTe)$ .

On se propose de reconstituer le signal de la façon suivante, exprimer  $x_r(t)$ .



2. Calculer la transformée de Fourier à temps discret (TFTD) de  $x(n)=\delta(n) + 6\delta(n -1) + 3\delta(n -2)$

3. Soit  $x(n)=1/N$  pour  $0 \leq n \leq N-1$  et 0 ailleurs.

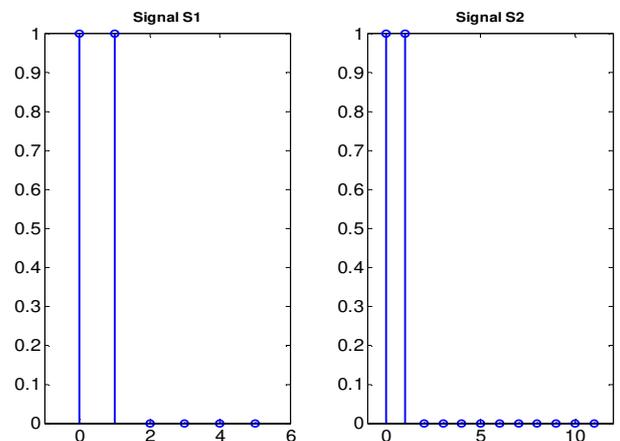
- Calculer et tracer la TFTD
- Calculer sa TFD  $X(k)$  pour N quelconque puis pour  $N=4$  et tracer la.

4. On a le signal suivant  $x(n)=e^{-an}U(n)$ , avec  $U(n)$  l'échelon unité, échantillonné à  $Te = 1$ .

- Calculer la TFTD
- Calculer la TFD pour N et comparer si  $N \rightarrow \infty$

5. La figure ci-contre représente 2 signaux discrets. Le signal S1 est obtenu avec un échantillonnage à la fréquence  $f_e=1200\text{Hz}$ .

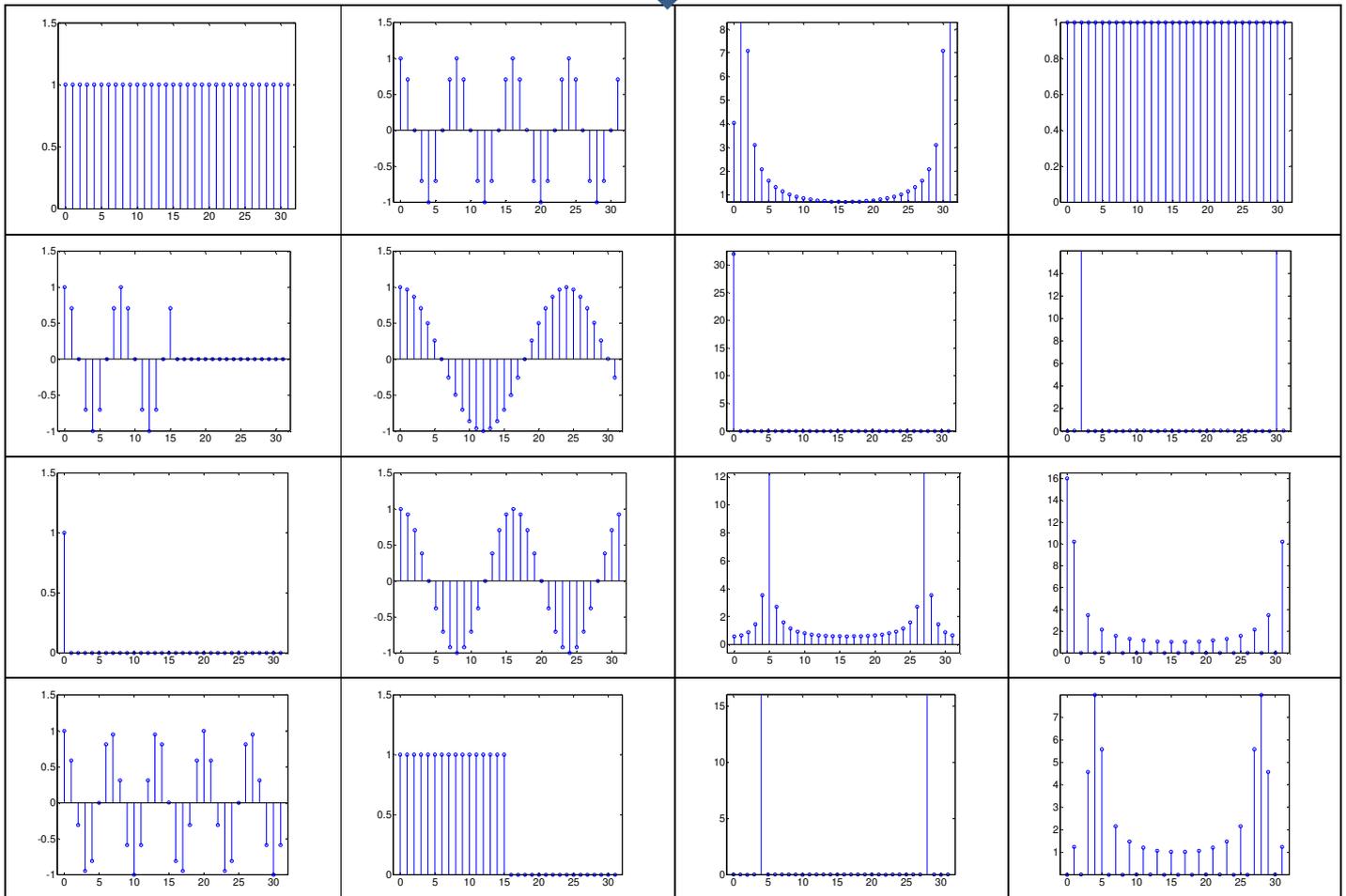
- Calculer l'énergie du signal.
- Déterminer et tracer l'auto-corrélation du signal S1.
- On décale le signal S1 de 2 vers la droite, sans faire de calcul, tracer son auto-corrélation.
- Déterminer la TFTD de S1 et tracer son module.
- Calculer la TFD pour tout N. Tracer la TFD pour  $N=6$ .
- Le signal S2 est obtenu à partir du signal S1, comment ?
- Tracer approximativement la TFD de S2 ( $N=12$ ).



6. Calculer la TFD du signal  $s(n)=n$  pour  $N=2, 3, 4, 8$ .

7. Etant donné les signaux  $s(n)=\{1,-2,3,2\}$  et  $v(n)=\{-2,1,2,3\}$ , déterminer  $w(n)=s(n)*v(n)$  par  
 - la méthode directe  
 - la méthode de la TFD

8. Etablir la correspondance entre chaque signal (gauche) et sa TFD (droite)  
 $x(n)$  ↓  $X(k)$



9. Un signal analogique est échantillonné à une fréquence  $f_e=7500$  Hz et  $N$  échantillons sont collectés.

- Quelle est la résolution fréquentielle pour  $N=1250$ ?
- Déterminer  $N$  permettant d'atteindre une résolution fréquentielle de 4.5 Hz.

10. Un signal analogique est échantillonné à une fréquence  $f_e= 500$ Hz et  $N=980$  échantillons sont collectés. On veut connaître la valeur du spectre à 120 Hz.

- Quel indice  $k$  de la TFD est le plus proche de 120Hz?
- Quel est le nombre minimum de 0 à rajouter obtenir une valeur de la TFD à 120 Hz exactement?
- Donner alors la valeur de l'indice  $k$  correspondant.

11. Soit  $x(n) = \sin(2\pi f_1 n) + 0.5 \sin(2\pi f_2 n)$  avec  $0 \leq n \leq 127, f_1 = 0.223, f_2 = 0.240$

- On veut employer un fenêtrage, calculer la largeur du lobe principale de chaque fenêtre
- Déterminer alors le fenêtrage permettant de visualiser les 2 sinusoides.

12. Soit  $x(n) = A_0 e^{2\pi j f_0 n} + A_1 e^{2\pi j f_1 n}$  pour  $n (0:N-1)$  où  $f_0=0.25$ , et  $f_1=0.70$  et  $f_e=2$   $N=20$ .
- Donner l'allure de la TTFD et de la TFD.
  - Quel est le but du fenêtrage ?
  - Quelle est la meilleure fenêtre à utiliser pour ce signal ? Justifier.

**Solutions**

1.  $D=60s$  et  $f_{max}=5kHz \Rightarrow N_{min}=D/T_{emin}=6.10^5$

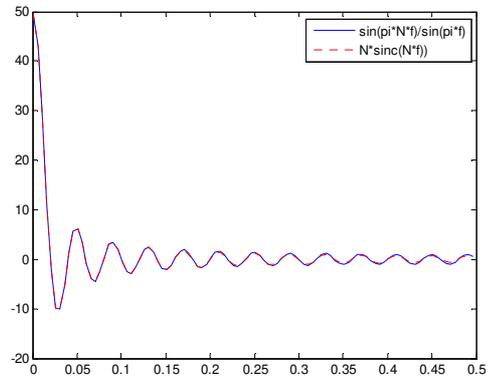
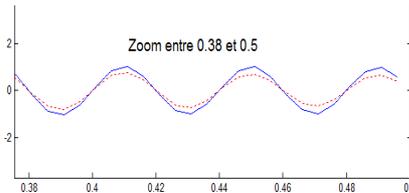
$$x_e(t) = \sum_n x(nT_e - \xi) \delta(t - nT_e - \xi) \Rightarrow X_e(f) = f_e \sum_n e^{-2\pi j f \xi} X(f - n f_e) \xi=0 \Rightarrow x_e(t) = \sum_n x(nT_e) \delta(t - nT_e)$$

Reconstruction théorique  $\Rightarrow X_R(f) = X_e(f) \frac{1}{f_e} \pi(f) \Rightarrow x_R(t) = \sum_n x(nT_e) * \text{sinc}(f_e(t - nT_e))$

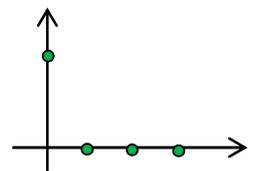
Bloqueur d'ordre 0  $\Rightarrow x_R(t) = \sum_n x(nT_e) \pi_{T_e}(t - nT_e - T_e/2)$

2.  $X(f) = 1 + 6.e^{-2\pi j f T_e} + 3.e^{-4\pi j f T_e}$

3.  $X(f) = \frac{1}{N} \frac{\sin(\pi N f T_e)}{\sin(\pi f T_e)} e^{-\pi j (N-1) f T_e} \approx \text{sinc}(N f T_e) e^{-\pi j (N-1) f T_e}$



$$X(k) = \frac{1}{N} \frac{\sin(\pi k)}{\sin(\pi k / N)} e^{-\pi j k (N-1) / N} \approx \text{sinc}(k) e^{-\pi j k (N-1) / N} \quad X(k) = \frac{1}{4} \frac{\sin(\pi k)}{\sin(\pi k / 4)} e^{-3 \pi j k / 4}$$



4. TTFD  $X(f) = \frac{1}{1 - e^{-a - 2\pi j f T_e}}$       TFD  $X(k) = \frac{1 - e^{-aN}}{1 - e^{-a - 2\pi j k / N}}$

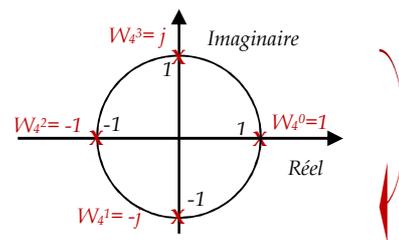
5.  $E=3$ ,  $R_{S1}(0)=2$ ,  $R_{S1}(1)=R_{S1}(-1)=1$ , identique,  $S_{S1}(f)=2\cos(\pi f T_e) e^{-3\pi j f T_e}$ ,  $S_{S1}(k)=2\cos(\pi k / N) e^{-\pi j k / N}$

$S_{S1}=[2, 1.73, 1, 0, 1, 1.73]$ , zéro padding, Rajouter des points entre la TFD de  $N=6$ .

6.  $N=2$ ,  $X(k)=[1, -1]$        $N=3$ ,  $X(k)=[3, \frac{1}{2}(-3 + \sqrt{3}j), \frac{1}{2}(-3 - \sqrt{3}j)]$        $N=4$ ,  $X(k)=[6, 2j - 2, -2, -2 - 2j]$

7. Convolution circulaire  $w(n) = s(n) * v(n) = [0, 18, 0, -2]$

$$\begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^4 & W_4^6 \\ W_4^0 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix} = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^4 & W_4^6 \\ W_4^0 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$



$S_k=[4, 4j-2, 4, -2-4j]$        $V_k=[4, -4+2j, -4, -4-2j]$

$W_k=[16, -20j, -16, 20j] \Rightarrow w(n)=[0, 18, 0, -2]$

Convolution Normale  $\longrightarrow$

Convolution Circulaire  $\longrightarrow$

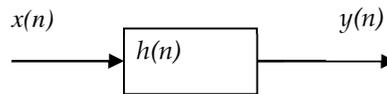
$$\begin{array}{r}
 \begin{array}{cccc}
 1 & -2 & 3 & 2 \\
 \times & -2 & 1 & 2 & 3 \\
 \hline
 -2 & 4 & -6 & -4 \\
 & 1 & -2 & 3 & 2 \\
 & & 2 & -4 & 6 & 4 \\
 + & & & & 3 & -6 & 9 & 6 \\
 \hline
 -2 & 5 & -6 & -2 & 2 & 13 & 6 \\
 + & 2 & 13 & 6 & & & & \\
 \hline
 0 & 18 & 0 & -2 & & & & 
 \end{array}
 \end{array}$$

- 8. 1-3, 2-7, 3-8, 4-1, 5-2, 6-4, 7-5, 8-6
- 9.  $\Delta f=6$   $N=1667$       10.  $k=235$  ,Rajouter 20 zéros et prendre  $k=240$
- 11. Rect=0.0156 Tri, Han, Ham= 0.03125 Black=0.0469  $f_1-f_2=0.017 > L\Delta f/2 \Rightarrow$  Hamming
- 12. 2 sinc l'un en  $f_0$  et l'autre en  $f_1$ 's'annulant tous les  $f_c/N=0.1$ , Atténuation des lobes secondaires, Blackman.

**Exercices supplémentaires**

- 1. Calculer la TFD du signal  $x(n)=1$  pour  $0 \leq n \leq 6$  et  $x(n)=0$  pour  $7 \leq n \leq 29$  puis représenter la.
- 2. Calculer la TFD de la suite  $x=[2,3,-1,1]$  sur  $N=4$  pts
- 3. Calculer la transformée de Fourier discrète (TFD) de la suite  $x(n)$  formée de  $N = 8$  points ( $n \in [0,7]$ ),obtenue en échantillonnant à la fréquence  $f_c = 16$  Hz le signal  $s(t) = 2\sin(8\pi t)+8\cos(4\pi t)$

4. On considère le système suivant :

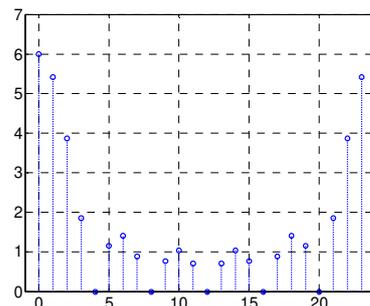


On suppose que  $H(0)=H(1)=1$  et  $H(2)=H(3)=0$ .

- Déterminer et tracer  $h(n)$  pour  $0 \leq n \leq 3$
- On pose  $H(4)=H(5)=\dots = H(N-1)=0$ . Donner le nom de cette technique.
- Quelle est le but de cette opération ? Donner, alors, l'allure approximative de  $h(n)$  pour  $N$  assez grand.

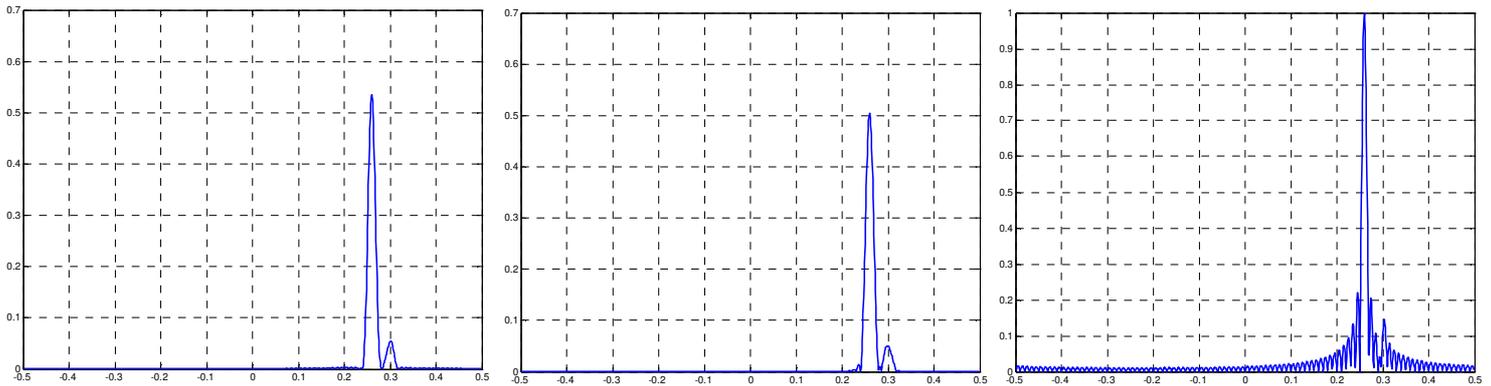
5. Le signal suivant représente la TFD  $X(k)$  d'un signal  $x(n)$  calculée sur 24 points

- Représenter la TFD  $X(f)$  du signal  $x(n)$ .
- Représenter sa TFD sur 6 points.
- Sans calcul, identifier  $x(n)$  :

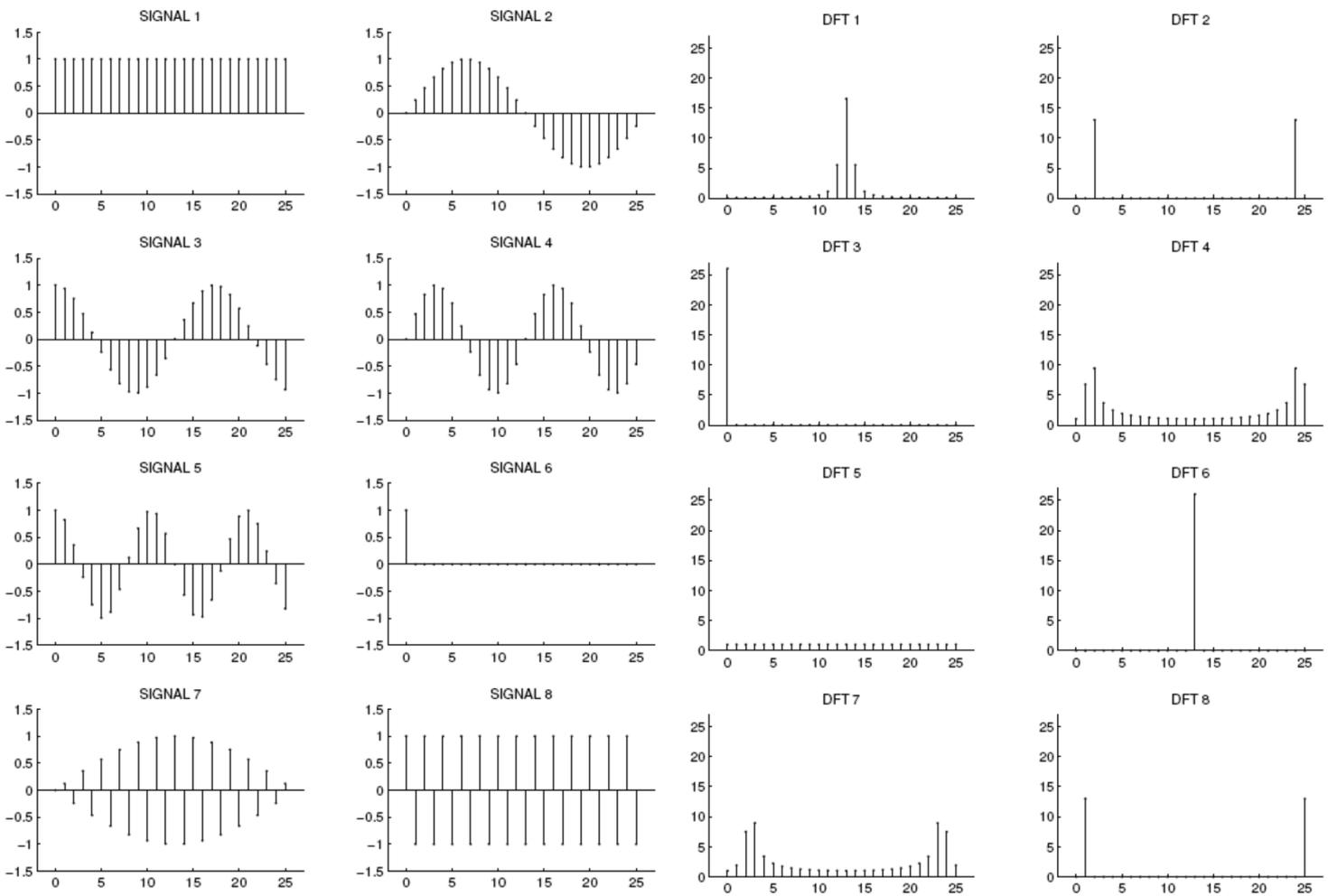


6. On souhaite calculer la TFD du signal suivant :  $e^{2\pi j f_0 n} + 0.1 e^{2\pi j f_1 n}$  avec  $f_0=0.26$ ,  $f_1=0.3$ ,  $N=100$  ( $f_c=1$ ) et  $NF=1024$ . A cette fin, on teste différentes fenêtres (Rectangulaire, Hamming et Hanning). Les TFD obtenues sont illustrées ci-dessous.

- Identifier les fenêtres utilisées (en justifiant vos réponses)
- Quelle est la meilleure fenêtre pour ce signal ? Justifier

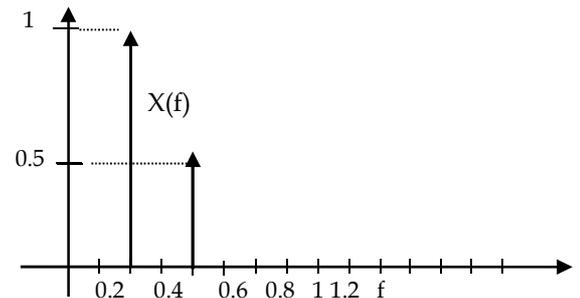


7. On donne les signaux et les TFD suivantes, associer chaque signal à sa TFD



8. L'analyse par la TFD d'un signal  $x(n)$  illimité a donné le signal ci-contre

1. Tracer sa TFD (on supposera que  $T_e=1$  et  $N=10$ ).
2. Tracer sa TFD si on utilise un fenêtrage de Hamming.
3. Quelle fenêtre choisir ? (Justifier) : \_\_\_\_\_



9. Pour chacun des cas suivants, on désire choisir le fenêtrage adéquat permettant d'analyser le contenu fréquentiel ( $f_e=1, N=20$ ):

-  $x(n)=2 \exp(2\pi j f_0 n) + \exp(2\pi j f_1 n)$   $f_0=0.3 f_1= 0.36$

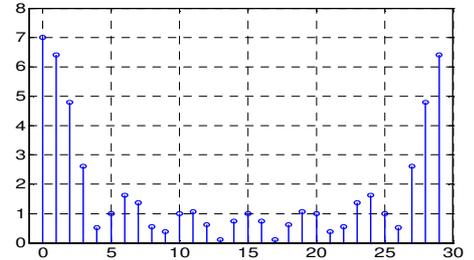
-  $x(n)=\exp(2\pi j f_0 n) + \exp(2\pi j f_1 n)$   $f_0=0.3 f_1= 0.46$

-  $x(n)=2 \exp(2\pi j f_0 n) + 0.2 \exp(2\pi j f_1 n)$   $f_0=0.3 f_1= 0.42$

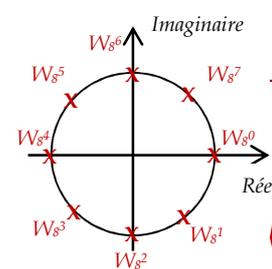
Solutions:

1.  $X(0)=7$   $X(k \neq 0) = \frac{\sin(\pi k 7/30)}{\sin(\pi k/30)} e^{-6\pi j k/30}$

2.  $X=[5, 3-2j, -3, 3+2j]$



3. 
$$\begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \end{bmatrix} = \begin{bmatrix} W_8^0 & S_0 \\ W_8^0 & W_8^1 & W_8^2 & W_8^3 & W_8^4 & W_8^5 & W_8^6 & W_8^7 & S_1 \\ W_8^0 & W_8^2 & W_8^4 & W_8^6 & W_8^8 & W_8^{10} & W_8^{12} & W_8^{14} & S_2 \\ W_8^0 & W_8^3 & W_8^6 & W_8^9 & W_8^{12} & W_8^{15} & W_8^{18} & W_8^{21} & S_3 \\ W_8^0 & W_8^4 & W_8^8 & W_8^{12} & W_8^{16} & W_8^{20} & W_8^{24} & W_8^{28} & S_4 \\ W_8^0 & W_8^5 & W_8^{10} & W_8^{15} & W_8^{20} & W_8^{25} & W_8^{30} & W_8^{35} & S_5 \\ W_8^0 & W_8^6 & W_8^{12} & W_8^{18} & W_8^{24} & W_8^{30} & W_8^{36} & W_8^{42} & S_6 \\ W_8^0 & W_8^7 & W_8^{14} & W_8^{21} & W_8^{28} & W_8^{35} & W_8^{42} & W_8^{49} & S_7 \end{bmatrix} = \begin{bmatrix} W_8^0 & S_0 \\ W_8^0 & W_8^1 & W_8^2 & W_8^3 & W_8^4 & W_8^5 & W_8^6 & W_8^7 & S_1 \\ W_8^0 & W_8^2 & W_8^4 & W_8^6 & W_8^8 & W_8^{10} & W_8^{12} & W_8^{14} & S_2 \\ W_8^0 & W_8^3 & W_8^6 & W_8^9 & W_8^{12} & W_8^{15} & W_8^{18} & W_8^{21} & S_3 \\ W_8^0 & W_8^4 & W_8^8 & W_8^{12} & W_8^{16} & W_8^{20} & W_8^{24} & W_8^{28} & S_4 \\ W_8^0 & W_8^5 & W_8^{10} & W_8^{15} & W_8^{20} & W_8^{25} & W_8^{30} & W_8^{35} & S_5 \\ W_8^0 & W_8^6 & W_8^{12} & W_8^{18} & W_8^{24} & W_8^{30} & W_8^{36} & W_8^{42} & S_6 \\ W_8^0 & W_8^7 & W_8^{14} & W_8^{21} & W_8^{28} & W_8^{35} & W_8^{42} & W_8^{49} & S_7 \end{bmatrix}$$



$s_n = [8 \quad 2+4\sqrt{2} \quad 0 \quad -2-4\sqrt{2} \quad -8 \quad 2-4\sqrt{2} \quad 0 \quad -2+4\sqrt{2}] W_8^k = \left[ 1 \quad \frac{\sqrt{2}}{2}(1-j) \quad -j \quad \frac{\sqrt{2}}{2}(-1-j) \quad -1 \quad \frac{\sqrt{2}}{2}(-1+j) \quad j \quad \frac{\sqrt{2}}{2}(1+j) \right]$

- 4. Voir examen 14/15
- 5. Voir interro 15/16
- 6. Hamming, Hanning, Rectangulaire, Meilleure : Hamming
- 7. (1,3), (2,8), (3,4), (4,2), (5,7), (6,5), (7,1) et (8,6)
- 8. Voir examen 15/16
- 9. Voir examen 16/17

TP n°1 : TFD et fenêtrage (avec Matlab)

Rappels

1. Transformée de Fourier Discrète

La TFD d'ordre N d'un signal numérique  $s(kT_e)$ ,  $k=0\dots N-1$  est définie par :

$$X(k) = X\left(\frac{k}{N}f_e\right) = \sum_{n=0}^{N-1} x(nT_e)e^{-2\pi jkn/N}, k = -N/2 \dots N/2$$

Sa transformée inverse est donnée par:

$$x(n) = x(nT_e) = \frac{1}{N} \sum_{k=-N/2}^{N/2-1} X(kf_e/N)e^{2\pi jkn/N}, n = 0 \dots N-1$$

Matlab fournit la fonction  $y=\text{fft}(x)$  pour calculer la transformée de Fourier complexe d'un vecteur de longueur N.

- La fonction  $\text{fft}(x)$  (respectivement  $\text{ifft}(X)$ ) permettra de calculer la TFD (respectivement la TFD<sup>-1</sup>) sur les N points, soit de 0 à N-1 correspondant à l'intervalle  $[0, f_e]$  (respectivement  $[0, NT_e]$ ). Le spectre obtenu sera un vecteur complexe que l'on pourra représenter en coordonnées cartésiennes (partie réelle et imaginaire fonction **real** et **imag**), ou en coordonnées polaires (module et phase, fonction matlab **abs** et **angle** ou **unwrap**).

- Pour une représentation entre  $-f_e/2$  et  $f_e/2$  de la TFD, il suffira d'écrire  $X=\text{fftshift}(X)$

- Pour augmenter la précision, il suffit d'écrire  $\text{fft}(x,NF)$  où  $NF > N$ . La transformée inverse est donnée alors par  $\text{ifft}(x,NF)$ .

2. Exemple d'utilisation de la FFT sous matlab

```

clc; clear all
N=100; fo=1000; Fe=8000; Te=1/Fe;
n=0:1:N-1;
x=sin(2*pi*fo*n*Te);
t=Te*[0:1:N-1];
subplot(2,2,1);plot(t,x,'gx:'); grid;
y=fft(x);
axe_FF=Fe*(0:1/N:1-(1/N));
subplot(2,2,2);plot(axe_FF,abs(y/N),'b.:'); grid;
title('Transformée de Fourier Discrète par FFT du signal entre 0 et fe')
xlabel('fréquence'); ylabel('Module du spectre');
y_dec=fftshift(y);
axe_freq=Fe*(-1/2:1/N:1/2-(1/N));
subplot(2,2,3); plot(axe_freq,abs(y_dec/N),'b.:'); grid;
title('Transformée de Fourier Discrète par FFT du signal entre -fe/2 et fe/2')
xlabel('fréquence'); ylabel('Module du spectre');
%FFT inverse %
FFT_INV=ifft(y);
subplot(2,2,4);plot(t,FFT_INV,'bx:'); grid;
title('Transformée de Fourier Discrète inverse du signal N=100')
xlabel('temps'); ylabel('Amplitude');
    
```

3. Programmes à réaliser

I. Reprendre l'exemple vu en cours  $x(n) = [1 \ 0 \ 0 \ 1]$ , calculer, afficher et visualiser sa TFD pour  $NF=4$  puis  $NF=32$  et commenter.

## II. TFD d'une porte

```
clc; clear all; close all;
N=6; Fe=1 ; Te=1/Fe;
n=Te*(1:N); x=ones(1,N);
subplot(2,1,1); stem(n,abs(x),'bx:'); grid;
NF=6; y=fft(x,NF); y_dec=fftshift(y);
subplot(2,1,2); axe_freq=Fe*(-1/2:1/NF:1/2-(1/NF));
stem(axe_freq,abs(y_dec/N),'r.:'); grid;
```

- Expliquer par un dessin le rôle de fftshift.
- Calculer la TFD théorique et comparer avec celle obtenue par ce programme.
- Prendre NF = 12 puis 24 puis encore 128 et commenter.

- Vérifier que l'énergie se conserve 
$$\sum_0^{N-1} |x(n)|^2 = \frac{1}{N} \sum_0^{N-1} |X(k)|^2$$

- Illustrer la propriété de décalage fréquentiel de la TF en calculant (théoriquement) et en représentant le module de la TFD de  $x(n) \times e^{2\pi j f_0 n}$  avec  $f_0=0.2$  Hz.
- Reprendre le signal original et mesurer la largeur  $\Delta f$  du lobe principal à partir de sa TF. Sa taille est-elle fonction de N ou NF?
- Afficher le module de la transformée de Fourier en décibels en écrivant `stem(axe_freq, 20*log10(abs(y_dec/N)),'r.:');` grid; puis vérifier que l'on a bien environ 13 dB entre le max du lobe principal et le max du lobe secondaire pour tout N (prendre N=10 et N=40).

## III. TFD d'un signal illimité et Effet du fenêtrage

- Déterminer la TTFD théorique de  $s(n) = A_0 e^{2\pi j f_0 n}$
- Que se passe-t-il si l'on calcule cette TTFD sur N ?

On calcule maintenant la TFD comme illustré par le programme suivant :

```
clc; clear all; close all;
N=50; NF=1024; Fe=8000; Te=1/Fe; f0=1680;
n=(0:N-1)*Te; x=exp(2*pi*j*f0*n);
y=fft(x,NF); yy=fftshift(y/N); axe_f=Fe*(-1/2:1/NF:1/2-(1/NF));
plot(axe_f,abs(yy)); grid; title('TFD Fenêtre Rectangulaire'); hold on;
```

- Pourquoi parle-t-on de fenêtre rectangulaire ? A-t-on vraiment multiplié par une fenêtre ?
- Que se passe-t-il si l'on remplace f0 par 1600?
- On remet f0 à 1680 et on rajoute au programme les lignes suivantes

```
fen=hanning(N);
xx=x.*fen; y=fft(xx,NF); yy=fftshift(y/N);
plot(axe_f,abs(yy),'r'); grid; title('TFD Fenêtre Hanning');
```

- Calculer la largeur du lobe principale pour chaque fenêtre en fonction de  $f_0$  et N.
- Comparer les fenêtres (amplitude principale, largeur des lobes, amplitudes des lobes).
- Tester d'autres fenêtres (Hamming, Blackman)
- Modifier ce programme pour calculer la TFD de la somme de 2 exponentielles de fréquences  $f_0=1680$  Hz et  $f_1=1800$  de même amplitude 1. Comparer alors les résolutions fréquentielles théoriques et pratiques des différentes fenêtres
- Refaire le même travail avec  $f_0=1680$  Hz et  $f_1=2100$  avec des amplitudes différentes ( $A_0=1$  et  $A_1=0.2$ ) et commenter.

Remarque Deux raies d'un spectre sont considérées comme séparables, si le maximum de l'une correspond au premier minimum nul de l'autre soit  $|f_1 - f_2| > \text{Largeur du lobe principal} / 2 = L \Delta f / 2$  (Voir tableau page 31).

TP n°1 : TFD et fenêtrage (en python)

Rappels

1. Transformée de Fourier Discrète

La TFD d'ordre N d'un signal numérique s(kTe), k=0...N-1 est définie par :

$$X(k) = X\left(\frac{k}{N}f_e\right) = \sum_{n=0}^{N-1} x(nT_e)e^{-2\pi jkn/N}, k = -N/2.....N/2$$

Sa transformée inverse est donnée par:

$$x(n) = x(nT_e) = \frac{1}{N} \sum_{k=-N/2}^{N/2-1} X(kf_e/N)e^{2\pi jkn/N}, n = 0.....N-1$$

Python fournit dans **scipy.fftpack** les fonctions nécessaires pour calculer la transformée de Fourier complexe d'un vecteur de longueur N.

- La fonction **scipy.fftpack.fft(x)** ou **numpy.fft.fft(x)** permettra de calculer la TFD sur les N points, soit de 0 à N-1 correspondant à l'intervalle [0, fe] (respectivement [0, NT<sub>e</sub>]). Le spectre obtenu sera un vecteur complexe que l'on pourra représenter en coordonnées polaires (module et phase, fonction **numpy.abs** et **numpy.angle** ou **numpy.unwrap**).

- Pour une représentation entre -fe/2 et fe/2 de la TFD, il suffira d'écrire **X=np.fft.fftshift(X)**

- Pour augmenter la précision, il suffit d'écrire **numpy.fft.fft(x,NF)** où NF>N. La transformée inverse (la TFD<sup>-1</sup>) est donnée alors par **numpy.fft.ifft(x,NF)**.

Quelques astuces

1. Pour des figures séparées, aller Outils ->Préférences->Console IPython->Graphise->Sortie: Automatique. Fermer spyder et ré-ouvrir.
2. Pour effacer toutes les variables avant exécution Préférences->Exécuter->Supprimer toutes les variables avant exécution
3. Ctl+L efface la console

2. Exemple d'utilisation de la FFT sous Spyder

```
import numpy as np
import matplotlib.pyplot as plt
N = 100; f0=1000; fe=8000; Te=1/fe
t = np.linspace(0.0, N*Te, N)
x = np.sin(2.0*np.pi*f0*t)
y = np.fft.fft(x)
y = np.fft.fftshift(y)
freq = np.arange(-N/2, N/2)*fe/N
plt.figure(1)
plt.subplot(311)
plt.plot(freq, np.abs(y))
plt.title('Module de la T.F.D. '); plt.grid(True); plt.xlabel('Fréquence (Hz) ');
plt.ylabel('Amplitude')
plt.subplot(312)
plt.plot(freq, np.angle(y))
plt.title('Phase de la T.F.D. '); plt.grid(True); plt.xlabel('Fréquence (Hz) ');
plt.ylabel('Amplitude')
plt.subplot(313)
plt.plot(freq, 20*np.log10(np.abs(y)/N))
plt.title('Module de la T.F.D. en db '); plt.grid(True); plt.xlabel('Fréquence (Hz) ');
plt.ylabel('Amplitude')
plt.show()
```

### 3. Programmes à réaliser

I. Reprendre l'exemple vu en cours  $x(n) = [1 \ 0 \ 0 \ 1]$ , calculer, afficher et visualiser sa TFD pour  $NF=4$  puis  $NF=32$  et commenter.

```
import numpy as np
import matplotlib.pyplot as plt
NF = 4; Te = 1.0; fe=1.0/Te;
x = np.array([1,0,0,1])
y = scipy.fftpack.fft(x,NF)
y = np.fft.fftshift(y)
freq = np.arange(-NF/2,NF/2)*fe/NF
plt.figure(2)
plt.subplot(211)
plt.stem(freq, np.abs(y),use_line_collection=True)
plt.title('Module de la T.F.D. '); plt.grid(True); plt.xlabel('Fréquence
(Hz) ');plt.ylabel('Amplitude')
plt.subplot(212)
plt.stem(freq, np.angle(y),use_line_collection=True)
plt.title('Phase de la T.F.D. '); plt.grid(True); plt.xlabel('Fréquence (Hz) ');
plt.ylabel('Amplitude')
plt.show()
```

### II. TFD d'une porte (à compléter)

```
import numpy as np
import matplotlib.pyplot as plt
from math import pow
N=6; Te = 1.0; fe=1.0/Te; NF=64; f0=0.2
x=np.ones(N)

E=0.0
for i in x:
    E+=pow(i, 2)

t = np.linspace(0.0, N*Te, N)
z = np.exp(2*np.pi*1j*f0*t)
```

- Expliquer par un dessin le rôle de `fftshift`.
- Calculer la TFD théorique et comparer avec celle obtenue par ce programme.
- Prendre  $NF = 12$  puis  $24$  puis encore  $128$  et commenter.
- Vérifier que l'énergie se conserve  $\sum_0^{N-1} |x(n)|^2 = \frac{1}{N} \sum_0^{N-1} |X(k)|^2$
- Illustrer la propriété de décalage fréquentiel de la TF en calculant (théoriquement) et en représentant le module de la TFD de  $x(n) \times e^{2\pi j f_0 n}$  avec  $f_0=0.2$  Hz.
- Reprendre le signal original et mesurer la largeur  $\Delta f$  du lobe principal à partir de sa TF. Sa taille est-elle fonction de  $N$  ou  $NF$ ?
- Afficher le module de la transformée de Fourier en décibels en écrivant puis vérifier que l'on a bien environ  $13$  dB entre le max du lobe principal et le max du lobe secondaire pour tout  $N$  (prendre  $N=10$  et  $N=40$ ).

**III. TFD d'un signal illimité et Effet du fenêtrage (à compléter)**

```
import numpy as np
import matplotlib.pyplot as plt

N = 50; f0=1680; fe=8000; Te=1/fe; NF=1024
t = np.linspace(0.0, N*Te, N)
x = np.exp(2*np.pi*1j*f0*t)
#help(scipy.signal.hamming)
window = sp.windows.hann(N);
```

- Déterminer la TFTD théorique de  $s(n) = A_0 e^{2\pi j f_0 n}$
- Que se passe-t-il si l'on calcule cette TFTD sur N ?
- Pourquoi parle-t-on de fenêtre rectangulaire ? A-t-on vraiment multiplié par une fenêtre ?
- Que se passe-t-il si l'on remplace  $f_0$  par 1600?
- On remet  $f_0$  à 1680 et on teste différentes fenêtres
- Calculer la largeur du lobe principale pour chaque fenêtre en fonction de  $f_e$  et N.
- Comparer les fenêtres (amplitude principale, largeur des lobes, amplitudes des lobes).
- Tester d'autres fenêtres (Hamming, Blackman)
- Modifier ce programme pour calculer la TFD de la somme de 2 exponentielles de fréquences  $f_0=1680$  Hz et  $f_1=1800$  de même amplitude 1. Comparer alors les résolutions fréquentielles théoriques et pratiques des différentes fenêtres
- Refaire le même travail avec  $f_0=1680$  Hz et  $f_1=2100$  avec des amplitudes différentes ( $A_0=1$  et  $A_1=0.2$ ) et commenter.

Remarque Deux raies d'un spectre sont considérées comme séparables, si le maximum de l'une correspond au premier minimum nul de l'autre soit  $|f_1 - f_2| > \text{Largeur du lobe principal} / 2 = L\Delta f / 2$  (Voir tableau page 31) .

## II. Analyse des filtres numériques par la TZ

La transformée de Fourier est un outil précieux d'analyse et de traitement des signaux. Cependant, dans certains problèmes (comme le filtrage numérique), les limites de la TF sont vite atteintes. La transformée en Z, qui s'applique aux signaux discrets, généralise la TF et permet de dépasser ces limites [10]. Elle est tout-à-fait analogue à la transformée de Laplace, mais plus facile à utiliser. Ce type de transformée permet de décrire aisément les signaux à temps discret et la réponse des systèmes linéaires invariants soumis à des entrées diverses. C'est un outil qui permet de calculer la réponse impulsionnelle d'un système linéaire invariant décrit par une équation aux différences finies. Elle permet l'interprétation directe des caractéristiques des signaux et des filtres dans le domaine des fréquences [9].

### 1. Transformée en Z

• Définition : La TZ est la généralisation de la TFTD ( $X(f) = \sum_{n=-\infty}^{+\infty} x(nT_e) e^{-2\pi j f n T_e}$ ).

Soit un signal discret  $x(n)$ . Sa TZ est définie par:

$$X(z) = \sum_{n=-\infty}^{+\infty} x(n) \cdot z^{-n} \quad \text{où } z \text{ est une variable complexe définie partout où cette série converge.}$$

En effet, comme cette transformation est une série infinie, elle n'existera que pour les valeurs de  $z$  pour lesquelles cette série converge.

#### Exemples :

-  $x(n) = \delta(n) \Rightarrow X(z) = 1,$

-  $x(n) = \delta(n-k) \Rightarrow X(z) = z^{-k}, \quad \text{si } k > 0 \text{ RDC} = \mathbb{C} - \{0\} \quad \text{si } k < 0 \text{ RDC} = \mathbb{C} - \{\infty\}$

-  $x(n) = (1, 2, 3, 5, 0, 2)$  on peut écrire  $x(n) = \delta(n) + 2 \cdot \delta(n-1) + 3 \cdot \delta(n-2) + 5 \cdot \delta(n-3) + 2 \cdot \delta(n-5)$

$X(z) = 1 + 2z^{-1} + 3z^{-2} + 5z^{-3} + 2z^{-5} \text{ RDC} = \mathbb{C} - \{0\}$

La série des puissances introduite dans l'équation de définition de la TZ ne converge que pour un sous-ensemble du plan complexe. Ce sous-ensemble est appelé région de convergence (RDC) ou domaine de convergence. Une région de convergence correspond à l'ensemble des valeurs de  $z$  telles que  $X(z)$  soit définie et à valeurs finies. Spécifier le domaine de convergence de la transformée est tout aussi important que la transformée elle-même [9].

• Condition d'existence : La transformée existe si la série converge. Pour cela, on utilise le critère de Cauchy

$$\lim_{n \rightarrow \infty} |U_n|^{1/n} < 1 \text{ sur la convergence des séries géométriques } S = \sum_{n=0}^{+\infty} U_n$$

L'ensemble des valeurs de la variable complexe z pour lesquelles la série converge est appelée Région De Convergence (RDC):

$$RDC = \left\{ z \in \mathbb{C} / \sum_{n=-\infty}^{+\infty} |x(n) \cdot z^{-n}| < +\infty \right\}$$

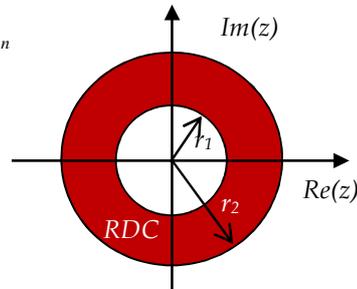
Exemple : TZ{u(n)}

$$U(z) = \sum_{n=0}^{+\infty} z^{-n} = \lim_{N \rightarrow \infty} \sum_{n=0}^{N-1} z^{-n} = \lim_{N \rightarrow \infty} \frac{1 - z^{-N}}{1 - z^{-1}}, \text{ la limite est finie si } |z^{-1}| < 1 \Leftrightarrow |z| > 1 \Rightarrow U(z) = \frac{1}{1 - z^{-1}} \text{ pour } |z| > 1$$

De façon générale, on montre que la RDC est un anneau de convergence centré sur l'origine défini par :

$$r_1 < |z| < r_2 \quad \text{avec } r_1 = \lim_{n \rightarrow +\infty} |x(n)|^{1/n} \text{ et } r_2 = \lim_{n \rightarrow +\infty} |x(-n)|^{-1/n}$$

où r1 peut être réduit à 0 et r2 peut être égal à l'∞.

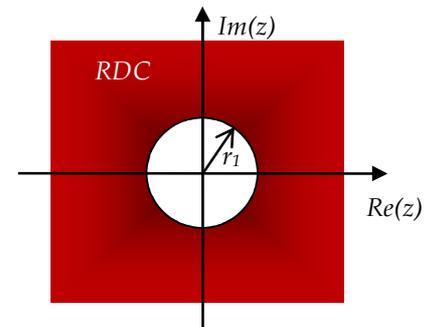
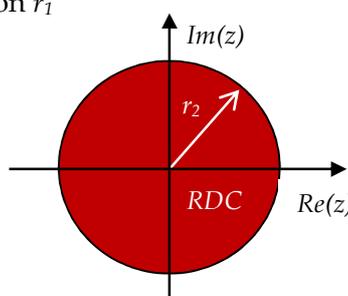


- x(n)=0 pour n < n0 ⇒ r2 = +∞,

RDC = région extérieure au cercle de rayon r1

- x(n)=0 pour n > n0 ⇒ r1 = 0

RDC = disque de rayon r2



⇒ système anti-causal : RDC cercle. système causal : RDC extérieure au cercle.

Exemples

- Soit a > 0,  $x(n) = \begin{cases} a^n & \text{si } n \geq 0 \\ 0 & \text{si non} \end{cases} \Rightarrow X(z) = \sum_{n=0}^{+\infty} a^n z^{-n} = \frac{z}{z - a}$ , convergente pour  $|z| > a$ .

- Soit b > 0,  $y(n) = \begin{cases} 0 & \text{si } n \geq 0 \\ -b^n & \text{si } n < 0 \end{cases} \Rightarrow Y(z) = \sum_{n<0} -b^n z^{-n} = \frac{z}{z - b}$ , convergente pour  $|z| < b$ .

- Soient  $a > 0, b > 0$ ,  $w(z) = \begin{cases} a^n & \text{si } n \geq 0 \\ b^n & \text{si } n < 0 \end{cases} \Rightarrow W(z) = \frac{z}{z-a} - \frac{z}{z-b}$ , convergente pour  $b > |z| > a$ .

**Remarque :** La TZ de  $a^n$  pour  $n \in ]-\infty, +\infty[$  n'existe pas.

## 2. Propriétés de la TZ

Les propriétés qui sont les plus utilisées sont résumées comme suit :

Si on définit :  $x(n) \xrightarrow{TZ} X(z)$ ,  $x_1(n) \xrightarrow{TZ} X_1(z)$  et  $x_2(n) \xrightarrow{TZ} X_2(z)$

- Linéarité :  $a.x_1(n) + b.x_2(n) \xrightarrow{TZ} a.X_1(z) + b.X_2(z)$  RDC intersection des deux RDC

- Théorème du retard :  $x(n-k) \xrightarrow{TZ} z^{-k}.X(z)$  RDC : identique

- Théorème de l'avance :  $x(n+k) \xrightarrow{TZ} z^k.X(z) - \sum_{n=0}^{k-1} x(n)z^{k-n}$  RDC : identique

- Multiplication par  $a^n$  :  $a^n x(n) \xrightarrow{TZ} X\left(\frac{z}{a}\right)$  RDC :  $a.r_1 < |z| < a.r_2$

- Retournement du temps :  $x(-n) \xrightarrow{TZ} X(z^{-1})$  RDC :  $1/r_2 < |z| < 1/r_1$

- Convolution :  $x_1(n) * x_2(n) \xrightarrow{TZ} X_1(z).X_2(z)$  et RDC : identique

- Théorème de dérivation :  $n.x(n) \xrightarrow{TZ} -z \cdot \frac{dX(z)}{dz}$  RDC : identique

- Théorème de la valeur initiale : si  $x(n)=0$  pour  $n < 0$  alors  $x(0) = \lim_{z \rightarrow \infty} X(z)$

- Théorème de la valeur finale :  $\lim_{n \rightarrow +\infty} x(n) = \lim_{z \rightarrow 1} (z-1)X(z)$

## Exemples

1.  $x(n) = \cos(w_0 n)U(n) = \frac{1}{2}(e^{jw_0 n} + e^{-jw_0 n})U(n) \Rightarrow$

$X(z) = \frac{1}{2} \left( \frac{1}{1 - e^{jw_0} z^{-1}} + \frac{1}{1 - e^{-jw_0} z^{-1}} \right) = \frac{1 - \cos(w_0)z^{-1}}{1 - 2\cos(w_0)z^{-1} + z^{-2}}$  avec  $|z| > 1$

2. Calculer la transformée en z des fonctions discrètes suivantes. Vérifier que les théorèmes de la valeur initiale et finale s'appliquent :  $x(n) = 0,8^n u(n)$  et  $y(n) = n0,8^n u(n)$ .

$$X(z) = \frac{z}{z-0.8} \quad Y(z) = -z \frac{d\left(\frac{z}{z-0.8}\right)}{dz} = \frac{0.8z}{(z-0.8)^2}$$

$$x(0) = \lim_{z \rightarrow \infty} \left| \frac{z}{z-0.8} \right| = 1 \quad y(0) = \lim_{z \rightarrow \infty} \left| \frac{0.8z}{(z-0.8)^2} \right| = 0$$

$$x(\infty) = \lim_{z \rightarrow \infty} \left| \frac{z(z-1)}{z-0.8} \right| = 0 \quad y(\infty) = \lim_{z \rightarrow \infty} \left| \frac{0.8z(z-1)}{(z-0.8)^2} \right| = 0$$

Quelques TZ

$x(n)$	$X(z)$	Région de convergence
$\delta(n)$	1	$\forall z$
$U(n)$	$\frac{1}{1-z^{-1}}$	$ z  > 1$
$a^n U(n)$	$\frac{1}{1-az^{-1}}$	$ z  >  a $
$n a^n U(n)$	$\frac{az^{-1}}{(1-az^{-1})^2}$	$ z  >  a $
$-a^n U(-n-1)$	$\frac{1}{1-az^{-1}}$	$ z  <  a $
$\cos(\omega_0 n T_e) U(n)$	$\frac{1-z^{-1} \cos(\omega_0 T_e)}{1-2z^{-1} \cos(\omega_0 T_e) + z^{-2}}$	$ z  > 1$
$\sin(\omega_0 n T_e) U(n)$	$\frac{z^{-1} \sin(\omega_0 T_e)}{1-2z^{-1} \cos(\omega_0 T_e) + z^{-2}}$	$ z  > 1$
$a^n \cos(\omega_0 n T_e) U(n)$	$\frac{1-az^{-1} \cos(\omega_0 T_e)}{1-2az^{-1} \cos(\omega_0 T_e) + a^2 z^{-2}}$	$ z  >  a $
$a^n \sin(\omega_0 n T_e) U(n)$	$\frac{az^{-1} \sin(\omega_0 T_e)}{1-2az^{-1} \cos(\omega_0 T_e) + a^2 z^{-2}}$	$ z  >  a $

3. TZ rationnelles (correspondant aux SLID)

Les systèmes linéaires invariants décrits par une équation aux différences finies possèdent une transformée en Z rationnelle c'est ainsi que celles-ci vont s'écrire comme le rapport de deux polynômes en  $z^{-1}$ .

$$\sum_{i=0}^M a_i y(n-i) = \sum_{i=0}^N b_i x(n-i) \xrightarrow{TZ} \sum_{i=0}^M a_i \cdot z^{-i} Y(z) = \sum_{i=0}^N b_i \cdot z^{-i} X(z)$$

On peut caractériser un système LI par  $h(n)$  ou par la transformée en Z ( $H(z)$ ) de sa réponse impulsionnelle  $h(n)$ , encore appelée fonction de transfert du système.

$$\Rightarrow H(z) = \frac{\sum_{i=0}^N b_i \cdot z^{-i}}{\sum_{i=0}^M a_i \cdot z^{-i}} = \frac{b_0}{a_0} z^{M-N} \frac{N(z)}{D(z)} = \frac{b_0}{a_0} z^{M-N} \frac{\prod_{i=1}^N (z - z_i)}{\prod_{i=1}^M (z - p_i)} = K z^{M-N} \frac{\prod_{i=1}^N (z - z_i)}{\prod_{i=1}^M (z - p_i)}$$

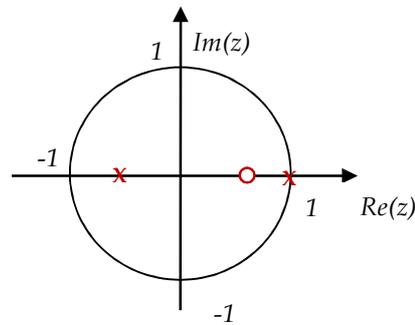
On appelle zéros, les valeurs de  $z$  pour lesquelles  $H(z)=0$  et on appelle pôles, les valeurs de  $z$  pour lesquelles  $H(z)$  est infini (annule le dénominateur). C'est ainsi que  $H(z)$  possède  $N$  zéros ( $z_i$ ),  $M$  pôles ( $p_i$ ). Si  $M > N$ , elle possède  $(M-N)$  zéros en 0, sinon  $(N-M)$  pôles en 0.

Ainsi, la position de ses pôles et de ses zéros (+le facteur d'amplitude  $K=b_0/a_0$ ) va nous fournir une description complète de  $H(z)$  (par conséquent de  $h(n)$  et  $H(f)$ ) donc du comportement du système.  $H(z)$  peut donc être représentée sous la forme d'un cercle modélisant la position des pôle set des zéros dans le plan complexe.

Exemple

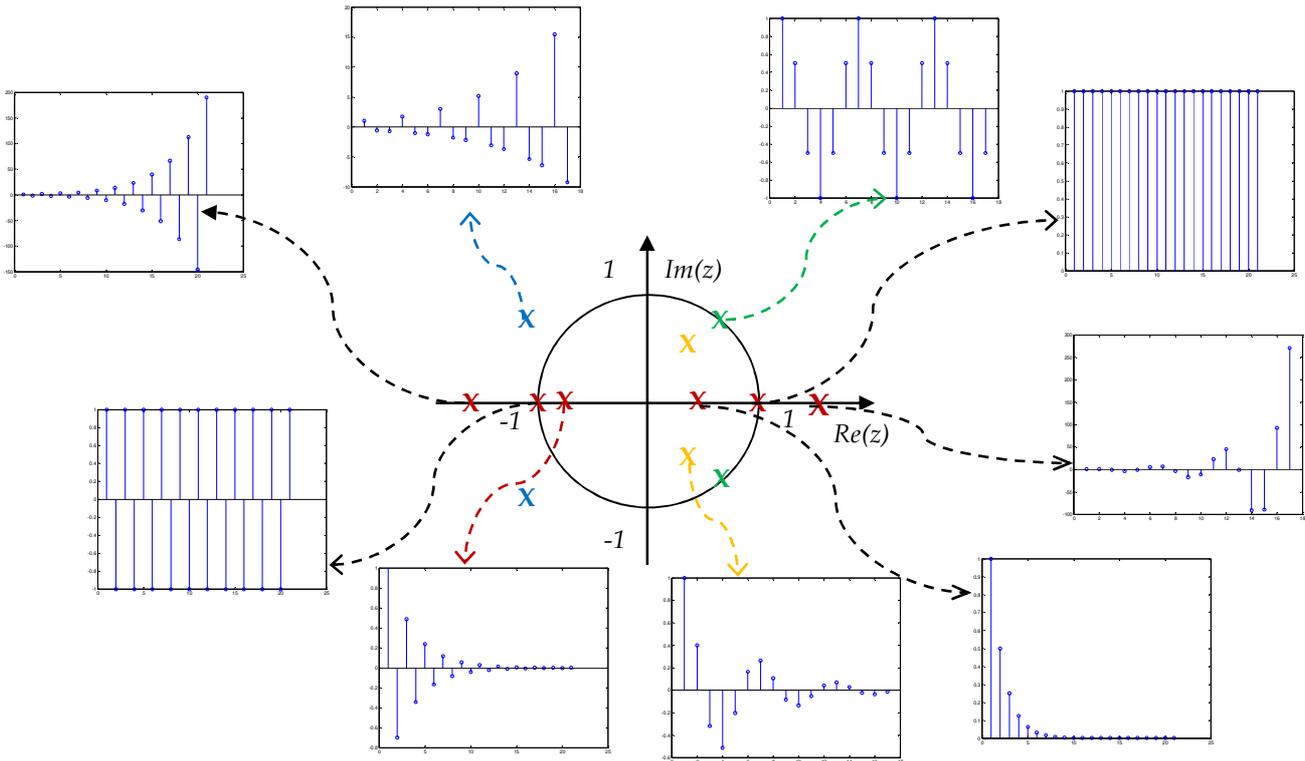
$$H(z) = \frac{3z - 2}{(z - 1)(z + 0.5)}$$

Un zéro en  $2/3$  et deux pôles  $p_1 = -0.5$  et  $p_2 = 1$



Remarques

- Dans la plupart des systèmes, les  $a_i$  et le  $b_i$  sont réels  $\Rightarrow$  les pôles et les zéros sont soit réels soit des paires de complexes conjugués.
- Rappelons que le rayon d'un système causal se trouve à l'extérieur d'un cercle. Par ailleurs, s'il est stable :  $\sum_n |h(n)| < \infty$ , puisque  $H(z) = \sum_{n=-\infty}^{+\infty} h(n) \cdot z^{-n}$ , il suffit donc que  $z=1$  fasse partie de la RDC.
- Pour un système causal et stable, tous les pôles sont à l'intérieur du cercle unité ( $|p_i| < 1, \forall i$ ). Le domaine de convergence ne peut contenir de pôles puisque la TZ ne converge pas aux pôles. S'il est anti-causal, il sera stable si les pôles sont à l'extérieur du cercle unité.
- Si le filtre est non-récurrent  $H(z) = \sum_{i=0}^N b_i \cdot z^{-i}$ . Un filtre RIF a tous ses pôles à l'origine et sera donc toujours stable.



- A un pôle  $p_i$  simple ou multiple va correspondre une réponse impulsionnelle qui converge si  $|p_i| < 1$ . Elle divergera dans le cas contraire, soit si  $|p_i| > 1$ .
- Sachant qu'à chaque pôle complexe est associé un pôle conjugué cela donnera une réponse impulsionnelle  $h(n)$  oscillante (cosinus ou sinus) amorties si  $|p_{i=1,2}| < 1$  ou divergentes si  $|p_{i=1,2}| > 1$ .
- Dans un système à phase minimale, tous les zéros sont à l'intérieur du cercle unité ( $|z_i| < 1, \forall i$ ).

#### 4. Détermination de la réponse fréquentielle des filtres numériques

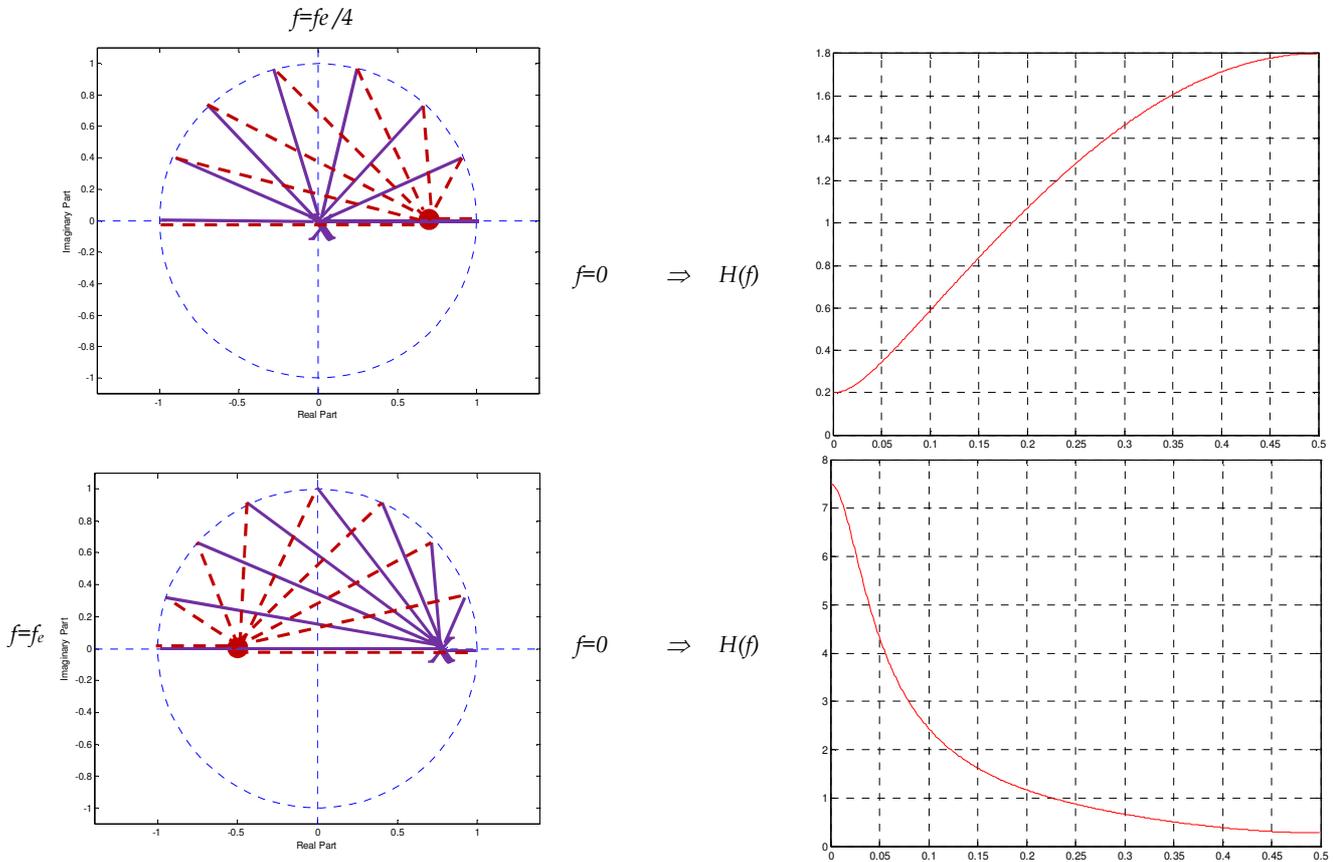
On suppose que le cercle unité ( $|z|=1$ )  $\in$  RDC de  $X(z)$ . On restreint le calcul de  $X(z)$  au cercle unité en posant  $z = e^{2\pi j f T_e}$ .

Lorsqu'un zéro est placé sur un point donné du plan en  $z$ , la réponse fréquentielle sera de 0 au point considéré. Un pôle quant à lui produira un pic au point correspondant. Plus les pôles ou les zéros sont proches du cercle unité, plus ils influencent la réponse en fréquence [11].

- un zéro ou un pôle à l'origine n'influent pas sur le module de la réponse fréquentielle.
- un zéro sur le cercle unité introduit une annulation du module pour la fréquence correspondant
- Un zéro au voisinage du cercle unité introduit une atténuation dans le module de la réponse en fréquence. Atténuation d'autant plus importante que le zéro est proche du cercle unité.
- Un pôle sur le cercle unité introduit une résonance infinie dans le module de la réponse en fréquence pour la fréquence correspondante.

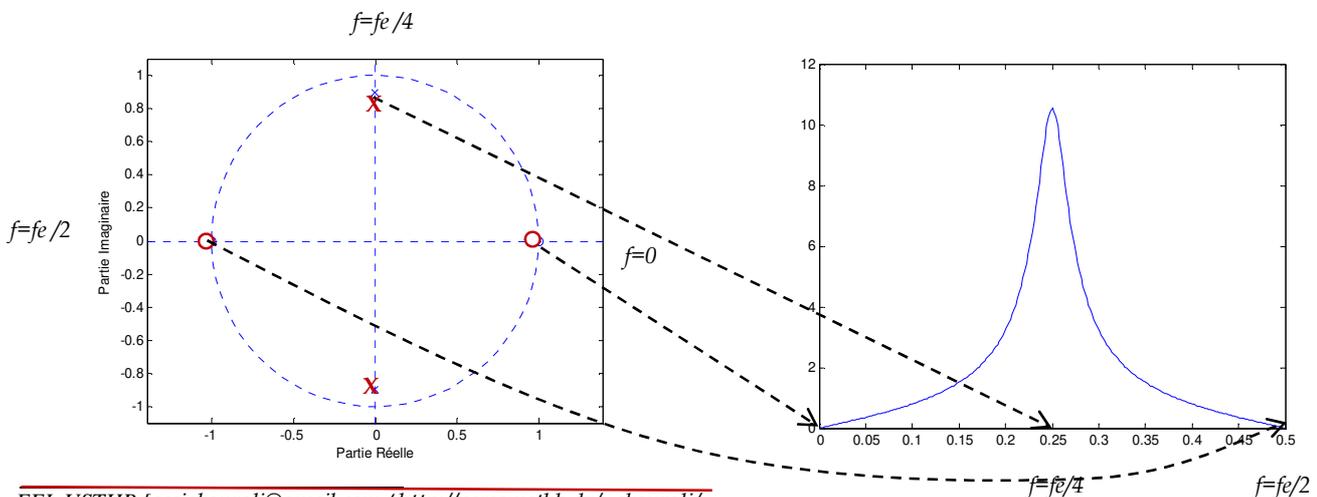
- Un pôle au voisinage du cercle unité introduit une résonance d'autant plus importante dans le module de la réponse en fréquence que le pôle est proche du cercle unité.

Exemples: 1 pôle en 0 et un zéro en 0.7. Pour obtenir l'allure de  $H(f)$ , on divise le vecteur du numérateur (en rouge) sur celui du dénominateur (en mauve). Et pour le suivant un pôle 0.8 et un zéro en -0.5

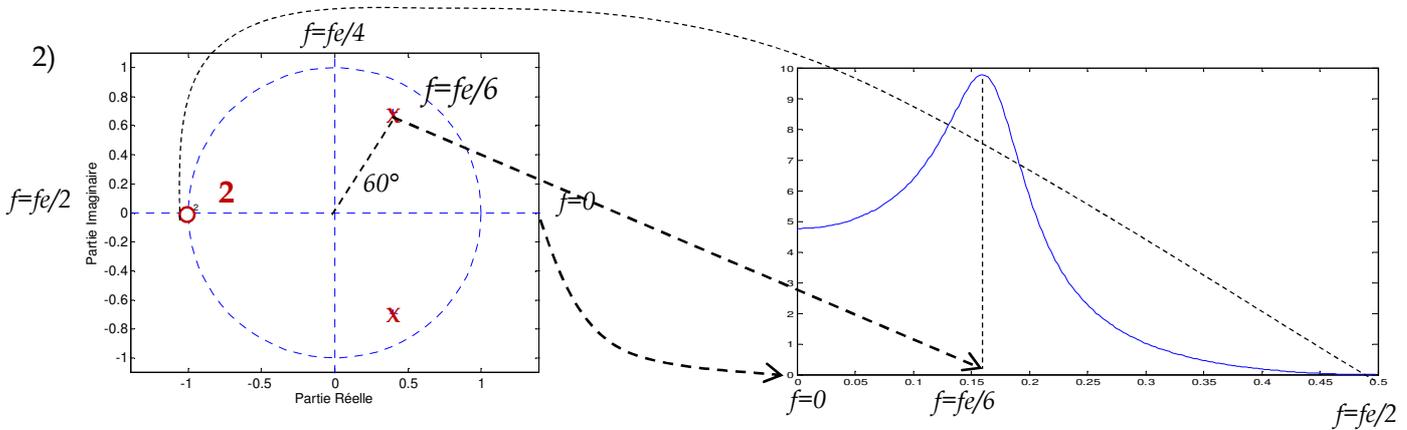


Autres Exemples

1) Sur la figure ci-dessous le cercle complet correspond à une fréquence d'échantillonnage  $f_e$ . Des pôles proches du cercle unité sont à l'origine de larges pics tandis que des zéros proches ou sur le cercle unité produisent des minima. Ce tracé nous permettra d'identifier la nature du filtre.



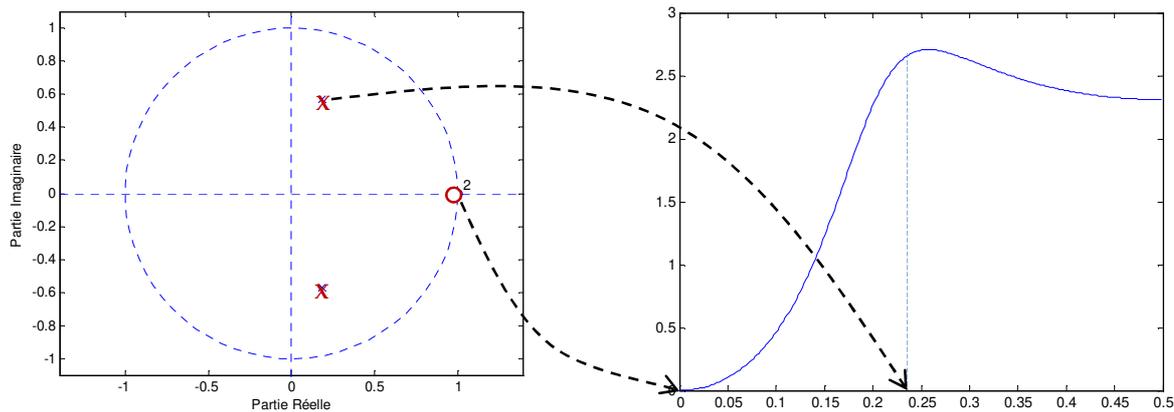
On peut aussi avoir une idée sur son comportement général : passe-bas, passe-haut ou passe-bande, connaître sa ou ses fréquences de coupure.



$$3) H(z) = \frac{z^2 - 2z + 1}{z^2 - 0.371z + 0.36} \Rightarrow \text{Zéros double en } z=-1, \text{ pôles } p_{1,2} = \pm 0.6e^{j72^\circ}$$

- Un zéro double en  $z = 1 \Rightarrow |H(f)| = 0$  pour  $f = 0$

- Des pôles proches du cercle unité  $\Rightarrow$  maxima.



Remarque : Puisque les coefficients du filtre sont réels, les pôles et zéros sont réels (sur l'axe des réels) ou paires de complexes conjugués.

### 5. Détermination de la réponse impulsionnelle des FN (TZ inverse)

La transformée en Z présente l'avantage d'être plus facilement inversible que la transformée de Fourier. Le passage de la TZ vers  $h(n)$  peut se faire par le biais de la transformées en Z de signaux élémentaires connus à condition qu'il soit possible d'écrire  $H(z)$  comme la combinaison de transformées élémentaires. Dans le cas contraire, on peut employer l'intégration sur un contour fermé en utilisant le calcul des résidus, ou le développement en puissance de  $z$  et de  $z^{-1}$ , ou encore le développement en fractions élémentaires [12].

1. La relation générale de la transformée en z inverse est donnée par l'équation donnée par l'intégrale de Cauchy :

$x(n) = \frac{1}{2\pi.j} \oint_C X(z).z^{n-1}.dz$ , où C est un contour fermé parcouru dans le sens inverse des aiguilles d'une montre contenant l'origine.

En pratique, on utilise le théorème des résidus:  $x(n) = \sum_{p_i \text{ poles de } z^{n-1} X(z)} \text{Res} [z^{n-1} X(z)]_{z=p_i}$

$$\text{Res}[z^{n-1} X(z)]_{z=p_i} = \frac{1}{(m-1)!} \frac{d^{m-1}}{dz^{m-1}} [(z-p_i)^m z^{n-1} X(z)]_{z=p_i}$$

Exemple :  $X(z) = \frac{z}{z-e^a} \Rightarrow p_i=e^a \Rightarrow \text{Res} [z^{n-1} X(z)]_{z=e^a} = [z^n]_{z=e^a} = e^{an}.u(n)$

2. Transformée inverse par division polynômiale : Il est possible de calculer la transformée en Z inverse selon les puissances croissantes de z<sup>-1</sup>(système causal) ou selon les puissances décroissantes de z (système anti-causal).

Exemples  $X(z) = \sum_n C_n z^{-n} \xrightarrow{TZ^{-1}} x(n) = C_n$

-  $y(n)=y(n-3)+x(n) \Rightarrow H(z) = \frac{1}{1-z^{-3}} = \sum_{k=0}^{\infty} (z^{-k})^3 = 1+z^{-3}+z^{-6}+\dots \Rightarrow h(n) = \sum_{k=0}^{\infty} \delta(n-3k)$

-  $X(z) = \frac{1}{1-az^{-1}}$  pour  $|z| > a$

Domaine de convergence extérieur à un cercle → signal causal → division pour avoir une série en z<sup>-1</sup>.

$$\begin{array}{r} 1 \\ -1+a.z^{-1} \\ 0+a.z^{-1} \\ -a.z^{-1}+a^2.z^{-2} \\ 0+a^2.z^{-2} \\ \dots \end{array} \quad \left| \begin{array}{l} 1-a.z^{-1} \\ \hline 1+a.z^{-1}+a^2.z^{-2}+\dots \\ \hline x[0] \quad x[1] \quad x[2] \\ \dots \end{array} \right.$$

On obtient :  $\frac{1}{1-az^{-1}} = 1+a.z^{-1}+a^2.z^{-2} \Rightarrow x(n) = a^n.u(n)$

-  $X(z) = \frac{1}{1-az^{-1}}$  pour  $|z| < a$

Région de convergence intérieure à un cercle → signal anti-causal → division pour avoir une série en z.

$$\begin{array}{r} z \\ -z+a^{-1}.z^2 \\ 0+a^{-1}.z^2 \\ -a^{-1}.z^2+a^{-2}.z^3 \\ 0+a^{-2}.z^3 \\ \dots \end{array} \quad \left| \begin{array}{l} -a+z \\ \hline -a^{-1}.z -a^{-2}.z^2 -a^{-3}.z^3 -\dots \\ \hline x[-1] \quad x[-\infty] \end{array} \right.$$

On obtient :  $\frac{1}{1-a.z^{-1}} = -a^{-1}.z^{-1} - a^{-2}.z^{-2} - \dots \Rightarrow x(n) = -a^n .u(-n - 1)$

Notons que la division peut se réaliser sans faire apparaître une expression analytique générale.

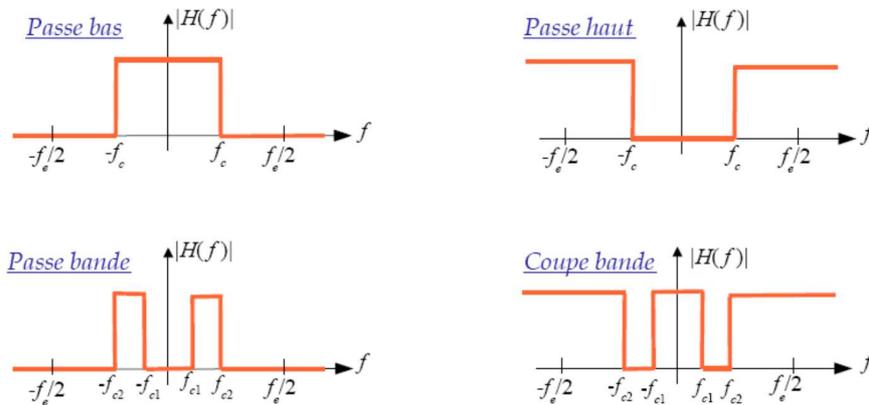
3. L'idée générale de cette approche consiste à trouver pour une fonction  $X(z)$  complexe un développement en fonctions en  $Z$  plus simples et pour lesquelles une transformée inverse est connue:

$$X(z) = \sum_i X_i(z) \xrightarrow{TZ^{-1}} x(n) = \sum_i x_i(n)$$

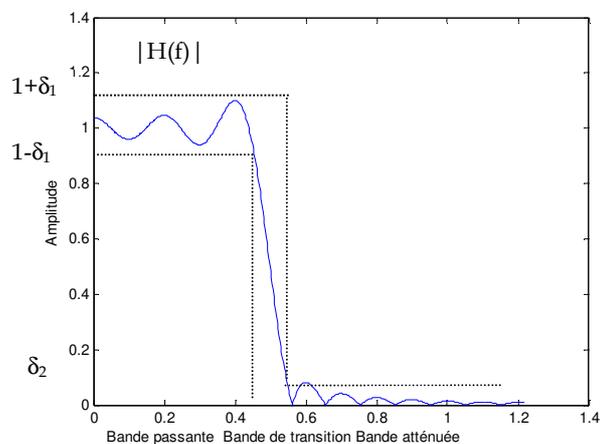
où les  $X_i(z)$  sont des fonctions dont les  $TZ^{-1}$  sont connues (Voir page 38).

### 6. Caractéristiques des filtres numériques

Un filtre numérique est constitué d'un groupement de circuits logiques astreints à un processus de calcul (ou algorithme) qui confère à ce filtre une fonction déterminée (passe-bas, passe-haut, passe-bande, réjecteur de bande, intégrateur  $[y(n)=(x(n)+x(n-1))/2]$ , différentiateur  $[y(n)=(x(n)-x(n-1))/2]$ , ...). Il faut souligner que certains filtres ne sont pas conçus pour arrêter une fréquence, mais pour modifier légèrement le gain à différentes fréquences, comme les égaliseurs. Ce sont tous des systèmes linéaires, discrets, invariants dans le temps et unidimensionnels. De plus, pour qu'ils soient physiquement réalisables, il faut qu'ils soient nécessairement causaux.



Les filtres représentés ci-dessus sont idéaux. Dans un cas réel il n'est pas possible d'obtenir une fréquence de coupure aussi raide. Le passage entre zones passantes et zones atténuées se fait par des zones dites "de transition"  $f_p-f_a$  dont la largeur va exprimer la sélectivité du filtre. Par ailleurs, les bandes passantes et atténuées ne sont également pas idéales, elles contiennent des ondulations dont l'amplitude est exprimée par les paramètres d'ondulation en bande passante  $\delta_1$  et bande atténuée  $\delta_2$  [9].



Remarque : Idéalement, il est souhaitable qu'un filtre possède une phase linéaire dans la bande passante. Une phase linéaire assurera un même déphasage pour toutes les fréquences (pas de distorsion). Les filtres FIR peuvent générer des filtres à phase linéaire à la condition que la réponse impulsionnelle soit symétrique.

Si un filtre est à phase linéaire, sa réponse fréquentielle est de la forme :

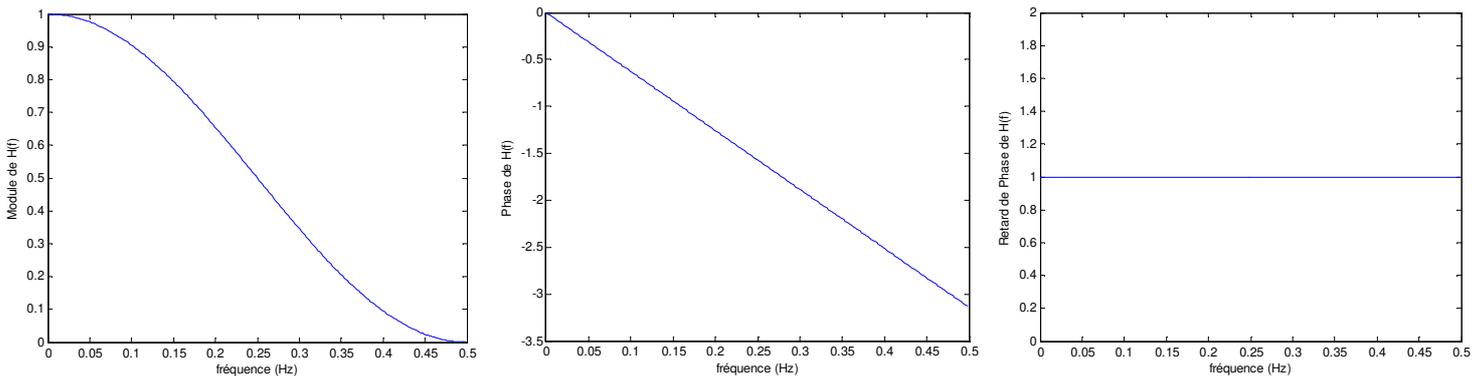
$$H(f) = R(f)e^{-j\phi(f)} \text{ avec } \phi(f) = \phi_0 + 2\pi f\tau$$

Et la dérivée de cette dernière par rapport à  $f$  fournit le 'retard de groupe', défini donc par :

$$\beta = -\frac{d\phi(f)}{df}$$

et qui correspond au retard subi par le signal après être passé par un filtre. Si la phase est linéaire (filtres RIF symétrique), le retard est constant et le signal à la sortie aura donc une distorsion minimale puisque l'effet de la phase sur le signal sera un simple décalage temporel (primordial dans un système audio)

Exemple  $y(n) = \frac{1}{4}(x(n) + 2x(n-1) + x(n-2)) \Rightarrow h(n) = \frac{1}{4}(\delta(n) + 2\delta(n-1) + \delta(n-2))$  et  $H(f) = e^{-2\pi jf} \cos^2(\pi f)$



**RII ou RIF ?**

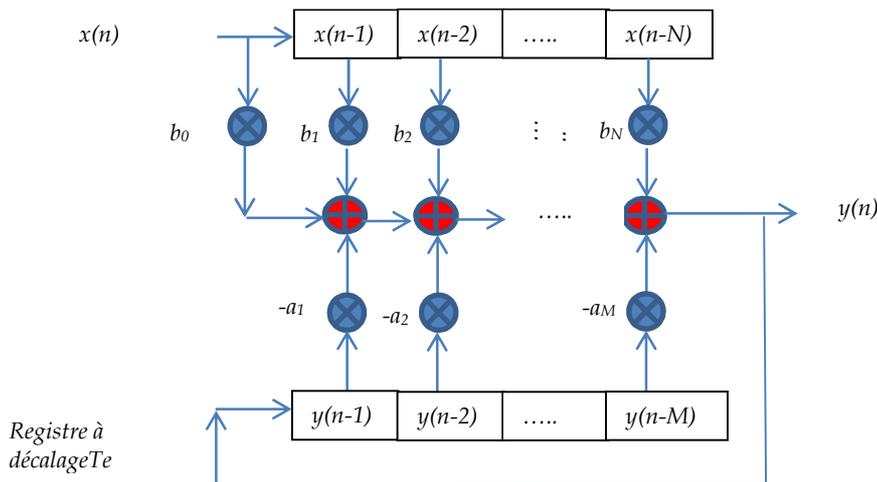
➤ Les filtres RII, on l'avantage qu'ils sont efficaces. Avec très peu de pôles et zéros on peut assurer la plupart des réponses fréquentielles dont on peut avoir besoin dans les applications audio. Cependant, le filtre étant récursif, les erreurs de précision numérique deviennent une question d'importance, car ils peuvent s'amplifier et devenir hors de contrôle, d'abord dans la forme de bruit, mais éventuellement dans la forme d'instabilité. La forme de la réponse impulsionnelle n'est pas facile à déterminer, non plus, car elle est définie indirectement par les pôles et zéros de  $H(f)$ .

➤ Par contre, les filtres RIF n'ont jamais des problèmes d'instabilité, car la sortie n'est qu'une somme finie d'échantillons de l'entrée. Cependant, quand la réponse impulsionnelle est longue, le numéro d'opérations peut devenir un facteur décisif quand il faut choisir entre RIF ou RII. Un autre avantage des RIF est le retard de groupe constant, qui permet d'avoir une distorsion de phase minimale sur le signal traité [13].

7. Structure des filtres numériques

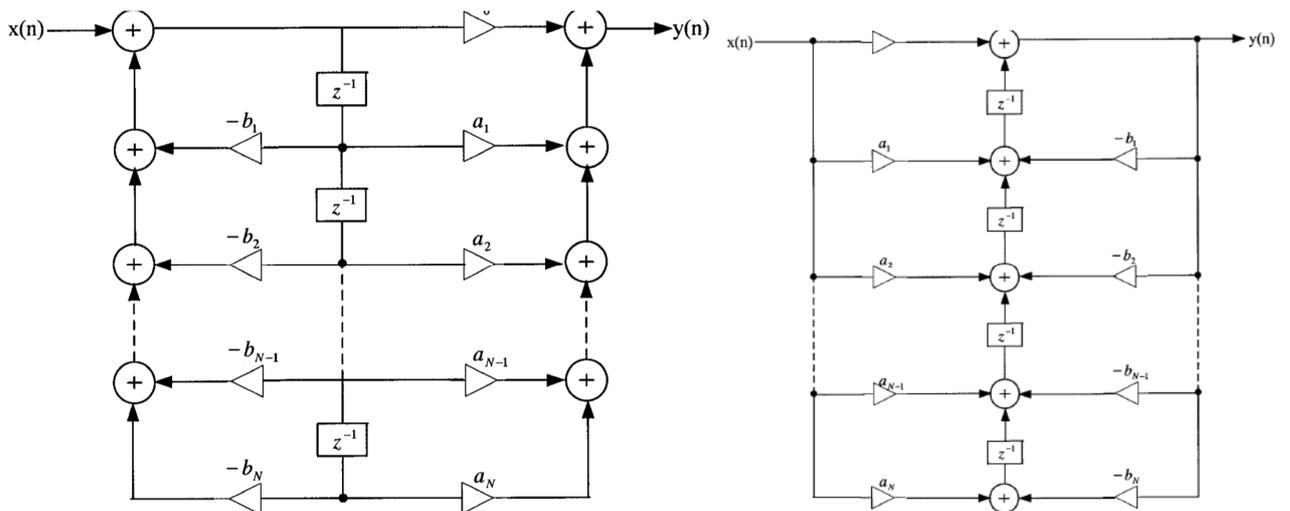
L'application d'un filtre numérique implique le calcul de la sortie  $y(n)$  à l'instant  $t=nT_e$  à partir des sorties et entrées précédentes plus la valeur courante de l'entrée.

$$\sum_{i=0}^M a_i y(n-i) = \sum_{i=0}^N b_i x(n-i) \quad \text{En prenant } a_0=1, \text{ on obtient}$$



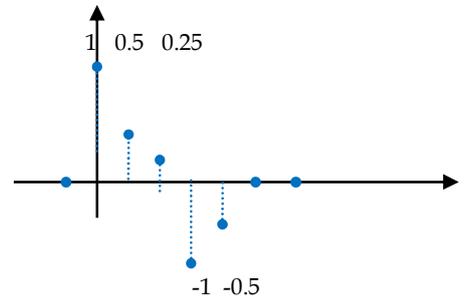
Un filtre numérique est généralement constitué des éléments suivants : un ou plusieurs organes de retard (ce sont des registres à décalage jouant le rôle de mémoires retardées), pilotés par une horloge de période; des opérateurs arithmétiques (additionneurs et multiplieurs); des registres fournissant les coefficients de pondération du filtre [18]. Ainsi, à chaque top d'horloge  $T_e$ , les valeurs des registres subissent un décalage permettant de calculer la nouvelle sortie. Cette structure peut être aussi réalisée par logiciel.

En employant la fonction de transfert  $H(z)$ , diverses structures peuvent être utilisées : structure directe (implémentation de l'équation aux différences), structure canonique (structure directe avec minimisation des composants) et structure en éléments simples.



Série n°2

1. Une séquence finie  $x(n)$  est définie par :  $x(n) = \begin{cases} \neq 0 & N_1 \leq n \leq N_2 \\ = 0 & \text{ailleurs} \end{cases}$



Déterminer le ROC suivant les différentes valeurs de N.

2. Donner la transformée en z de la fonction numérique discrète  $x(n)$  représentée par le graphique ci-contre.

3. Calculer la transformée en z,  $X(z)$ , et esquisser la carte des pôles et zéros ainsi que la ROC pour chacune des séquences suivantes :

$$x(n) = (0.5)^n u(n) + (0.25)^n u(n), y(n) = (0.25)^n u(n) + (0.5)^n u(-n-1), z(n) = (0.5)^n u(n) + (0.25)^n u(-n-1)$$

4. Calculer la Transformée en Z du signal  $x(n) = \text{rect}_N(n)$ ,  
 - en appliquant la définition de la TZ directement,  
 - en utilisant le signal échelon et le théorème du retard.

5. Soit  $H(z)$  la fonction de transfert d'un SLIT causal avec :  $H(z) = \frac{az-1}{z-a}$  avec  $a$  réel

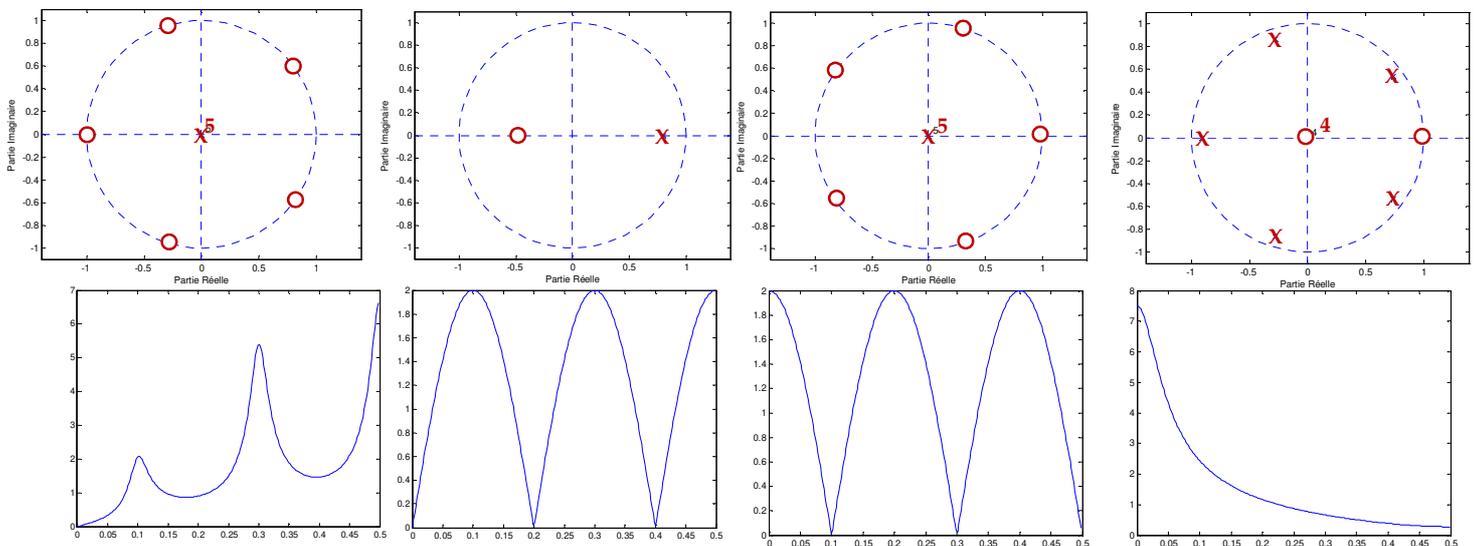
Déterminer les valeurs de  $a$  pour lesquelles  $H(z)$  correspond à un système stable. Prendre une valeur de  $a=0.5$ . Représenter alors les pôles et zéros de la fonction, la région de convergence. Donner et tracer  $|H(f)|$ .

6. Soit les SLIT décrits par les équations suivantes :

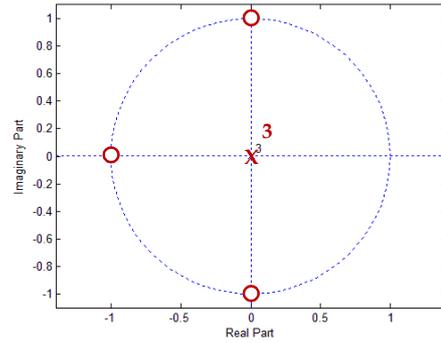
- $y(n) = 3y(n-1) - 2y(n-2) + x(n)$
- $y(n) = 0.3y(n-1) + 0.3y(n+1) - 0.3x(n)$

Pour chaque cas, déterminer la fonction de transfert du système. Etudier la stabilité et la causalité et calculer la réponse impulsionnelle.

7. Etablir les correspondances entre les diagrammes pôles zéros et les réponses en fréquence pour une fréquence d'échantillonnage  $f_e=1$  en justifiant vos choix :



8. On suppose que le tracé des pôles et des zéros de ce système est le suivant :
- Est-ce un filtre RIF ou RII ? (Justifier votre réponse)
  - Donner l'allure approximative de H(f)
  - Déterminer H(z) puis déterminer et tracer h(n)
  - A partir de h(n), étudier la stabilité, la causalité et l'invariance de ce filtre.
  - Calculer et tracer H(f) pour au moins 3 valeurs
  - Ce filtre possède-t-il un retard de groupe constant (justifier)
  - Déterminer sa réponse pour une entrée échelon x(n)=U(n).



9. Trouver la séquence y(n) qui a comme transformée en z :  $Y(z) = \frac{1}{6 - 5z^{-1} + z^{-2}}$ .

10. De quelle fonction x(n), la fonction:  $X(z) = \frac{\sqrt{2}/2 z^2}{z^2 - 0.8\sqrt{2}z + 0.64}$  est la transformée en Z

11. Trouver h(n) correspondant à la transformée en z suivante :  $H(z) = \frac{z^{-2} - 2z^{-1} + 1}{a^2 z^{-2} - 2az^{-1} + 1}$

**Solutions**

1. Si  $N_1 \geq 0$  RDC=C-{0}      Si  $N_2 \leq 0$  RDC=C-{∞}      Si  $N_1 \leq 0 \leq N_2$  RDC=C-{0, ∞}

2.  $X(z) = 1 + 0,5 z^{-1} + 0,25 z^{-2} - 1 z^{-3} - 0,5 z^{-4}$  RDC=C-{0}

3.  $X(z) = \frac{z(2z-0.75)}{(z-0.5)(z-0.25)} \quad |z| > 0.5$        $Y(z) = \frac{-0.25z}{(z-0.5)(z-0.25)} \quad 0.25 < |z| < 0.5$       N'existe pas

4.  $(1-z^{-N})/(1-z^{-1})$  5. Pôle z=a, stable si  $|a| < 1$ . Le module vaut toujours 1 (cellule passe-tout).

6.  $H(z)=z^2/(z^2-3z+2)$  causal pour  $|z| > 2$  mais instable avec  $x(n)=(2^{n+1}-1)U(n)$   
 $H(z)=0.375 (1/(3z-1)+1/(0.33z-1))$  causal pour  $|z| > 3$  instable avec  $x(n)=0.375(3^n - 0.33^n)U(n)$

8. Tous les pôles en 0 alors RIF,  $H(z)=z^{-3}+z^{-2}+z^{-1}+1$ ,  $h(n)=\delta(n-3)+\delta(n-2)+\delta(n-1)+1$ ,  
 $H(f)=2(\cos(3\pi fT_e) + \cos(\pi fT_e))e^{-3\pi j fT_e}$ , Retard de groupe cst,  $x(n)=U(n)$  alors  $y(n)= U(n-3)+U(n-2)+U(n-1)+U(n)$

9.  $y(n) = \frac{1}{2} \left(\frac{1}{2}\right)^n U(n) - \frac{1}{3} \left(\frac{1}{3}\right)^n U(n)$  10.  $x(n) = 0.8^n \cos\left[\frac{\pi}{4}(n-1)\right] U(n)$

11.  $x(n) = (n+1)a^n U(n+1) - 2na^{n-1} U(n) + (n-1)a^{n-2} U(n-1)$

**Exercices supplémentaires**

1. Soit un système linéaire invariant dans le temps (SLID) dont la réponse impulsionnelle h(n) est telle que :  
 $h(n)=1$  pour  $0 \leq n \leq 3$  et 0 ailleurs

Calculer la réponse y(n) à la suite x(n) définie par :

-  $x(n)=a^n$  pour  $0 \leq n \leq 5$ , avec  $a=0.7$  et  $x(n)=0$  ailleurs.

-  $x(n)=\cos(2\pi n/8)$  pour  $0 \leq n \leq 7$  et  $x(n)=0$  ailleurs.

Solution  $y(0) = 1$   $y(1) = 1+a = 1.7$   $y(2) = 1+a+a^2 = 2.19$   $y(3) = 1+a+a^2+a^3 = 2.533$   $y(4) = a+a^2+a^3+a^4 = 1.7731$   $y(5) = a^2+a^3+a^4+a^5 = 1.24117$   
 $y(6) = a^3+a^4+a^5 = 0.75117$   $y(7) = a^4+a^5 = 0.40817$   $y(8) = a^5 = 0.16807$

$y(0) = 1$   $y(1) = 1 + \cos(p/4) = 1.707$   $y(2) = 1.707$   $y(3) = 1$   $y(4) = -1$   $y(5) = -2.414$   $y(6) = -2.414$   $y(7) = -1$   $y(8) = 0$   $y(9) = 0.707$   $y(10) = 0.707$   
 2. Soit  $y(n) = x(n) + ax(n-1) + by(n-1)$ , l'équation aux différences d'un système discret causal.

- a) Trouvez  $h(n)$ , la réponse impulsionnelle de ce système ; pour quelles valeurs de  $a$  et  $b$  le système est-il stable
- b) Trouvez la réponse impulsionnelle du système formé par la mise en série de deux systèmes  $h(n)$ .
- c) Même question pour la mise en parallèle de deux systèmes  $h(n)$ .

**Réponses :** a)  $h(n) = b^n u(n) + ab^{n-1} u(n-1)$ ; stable pour  $a$  finie et  $|b| < 1$ .

b)  $h(n)*h(n) = (n+1)b^n u(n) + (2nab^{n-1} + (n-1)a^2 b^{n-2})u(n-1)$       c)  $2h(n) = 2b^n u(n) + 2ab^{n-1} u(n-1)$

3. On considère un système linéaire régi par l'équation aux différences suivante :

$$y(n) = (x(n+m) + x(n+m-1) + x(n+m-2))/3 \quad \text{où } m \text{ est un paramètre entier.}$$

- Montrer que ce système est linéaire invariant dans le temps. Etudier la causalité et la stabilité selon les valeurs de  $m$ .

- Calculer la fonction de transfert  $H(z)$  de ce système pour  $m=0$  et  $m=1$ .

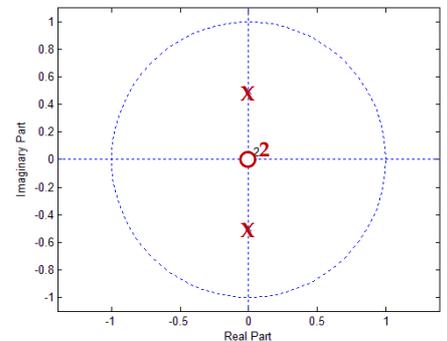
- En déduire la réponse fréquentielle.

**Réponses** Le système est causal si  $m < 1$  et toujours stable. Pour  $m=0, H(z) = 1/3(1+z^{-1}+z^{-2})$ ,

$H(f) = 0.33(1 + 2 \cos(2\pi fT_e))e^{-2j\pi fT_e}$  (filtre RIF passe-bas à phase linéaire). Pour  $m=1$ , de même filtre moyenneur

4. On suppose donné le tracé des pôles et des zéros du système suivant :

- Est-ce un filtre RIF ou RII ? (Justifier votre réponse)
- Donner l'allure approximative de  $H(f)$
- Déterminer  $H(z)$  puis déterminer l'équation de récurrence
- Déterminer et tracer  $h(n)$
- A partir de  $h(n)$ , étudier la stabilité, la causalité et l'invariance du filtre



5. Déterminer en utilisant la décomposition en éléments simples, la forme du signal  $x(n)$  dont la TZ est donnée par :

$$X(z) = \frac{z^2}{z^2 - 3z + 2} \quad \text{avec } |z| > 2 \quad \text{et} \quad X(z) = \frac{z^2}{z^2 - (a+1)z + a} \quad \text{avec } |a| < 1$$

**Réponses :**  $x(n) = (2^{n+1} - 1)u(n)$        $x(n) = 1/(1-a) + a^n a / (a-1) = (1-a^{n+1}) / (1-a)$

6. On considère la transformée :  $X(z) = \frac{z}{z - z_0} + \frac{z}{z - z_0^*}$  on pose  $z_0 = e^{(r+j\theta)}$ , déterminer  $x(n)$ .

**Réponse**  $x(n) = 2e^{rn} \cos(n\theta)U(n)$

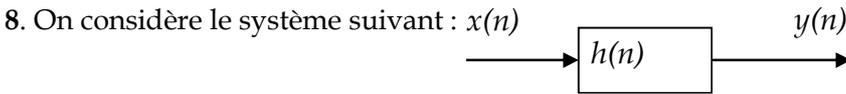
7. En utilisant la méthode des résidus puis celle de décompositions en éléments dans la TZ est connue, déterminer

$h(n)$  dont la transformée est :  $H(z) = \frac{z - z_0}{(z - p_0)(z - p_0^*)}$   $z_0$  : réel,  $p_0 = \rho e^{j\theta}$

**Réponse**  $h(n) = \frac{\rho^{n-1}}{\sin \theta} \left[ \left( 1 - \frac{z_0}{\rho} \cos \theta \right) \sin(n\theta) + \frac{z_0}{\rho} \sin \theta \cos(n\theta) \right] U(n-1)$

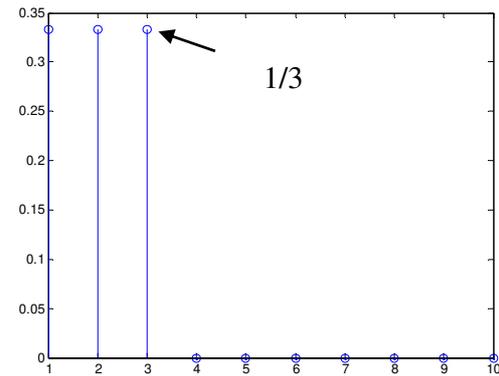
$$y(n) = x(n-1) - z_0 x(n-2) + 2\rho \cos \theta y(n-1) - \rho^2 y(n-2)$$

Certains exercices sont pris des références suivantes [14][15]



On suppose que  $h(n)$  est donné comme ci-contre.

- Calculer et tracer son auto corrélation et en déduire son énergie
- Tracer son auto corrélation si l'on suppose qu'il est périodique de période 9.
- Etudier la causalité
- Est ce un filtre RIF ou RII
- A partir de l'expression de  $h(n)$ , déduire le rôle de ce filtre :
- Déterminer l'équation aux récurrences du système :
- Déterminer les pôles et zéros de ce filtre puis donner leur tracé. En déduire un tracé approximative de  $|H(f)|$
- Calculer et tracer  $|H(f)|$  puis en déduire le tracé du module de la TFD pour  $N=6$ .



9. On considère que l'équation aux récurrences du système suivant est donnée comme suit :

$$y(n) = 0.9 y(n-1) - 0.81 y(n-2) + x(n) + 2 x(n-1) + x(n-2)$$

- Etudier la causalité et l'invariance de ce système
- Est-ce un filtre RIF ou RII ?
- Déterminer  $H(z)$  et donner le tracé des pôles et zéros, en déduire le rôle de ce filtre :
- Donner les allures approximatives de  $h(n)$  et  $|H(f)|$
- Déterminer  $h(n)$  et tracer la pour les 3 premières valeurs
- On suppose que  $f_e = 6$  kHz, quelle sera la sortie du filtre si l'on donne en entrée : un signal bruité par une sinusoïde de 500 Hz puis un signal composé de 2 sinusoïdes l'une de 500 Hz et l'autre de 3000 Hz

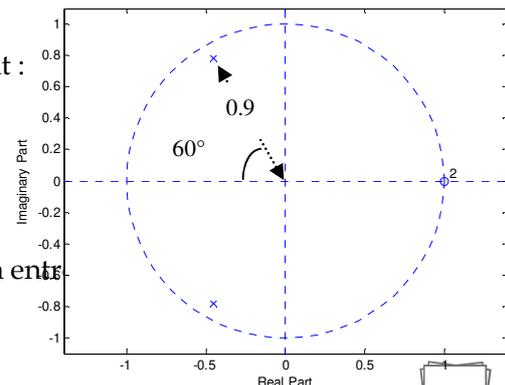
10. On considère le système LIT décrit par l'équation aux récurrences suivantes :

$$x(n) = \delta(n) + 2\delta(n-1) + 3\delta(n-2) + 4\delta(n-3) \text{ et } h(n) = 9\delta(n) + 7\delta(n-1) + 4\delta(n-2) + \delta(n-3)$$

- Ce système est-il invariant ? Justifier.
- Etudier la stabilité et la causalité de  $x(n)$ ,  $h(n)$  et  $y(n)$ .
- Les signaux  $x(n)$ ,  $h(n)$  et  $y(n)$  sont-ils à énergie finie (ou infinie) ? Justifier
- Déterminer  $y(n)$  directement et tracer le.
- Déterminer  $y(n)$  en passant par la TZ de  $x(n)$  et  $h(n)$
- On considère que  $x(n)$  et  $h(n)$  sont périodiques de période 4, déterminer  $y(n)$  de 2 façons.
- Quelle est lien entre la convolution et l'autocorrélation ?

11. On suppose que le tracé des pôles et des zéros d'un système est le suivant :

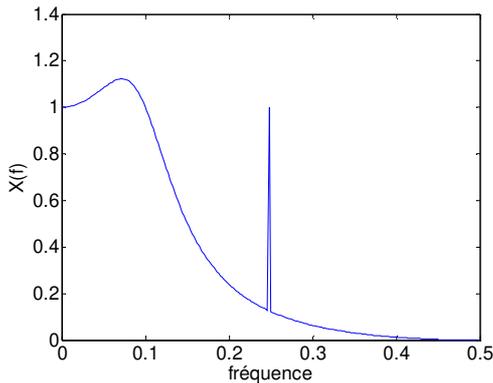
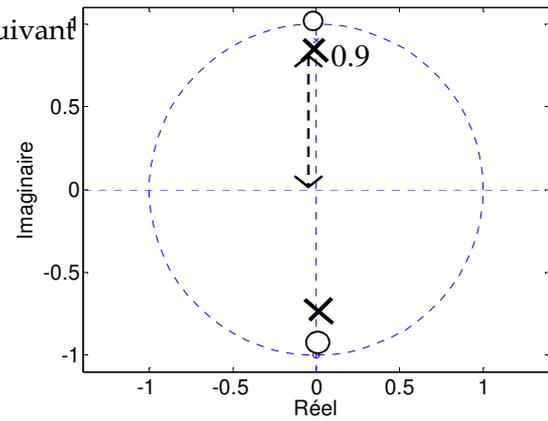
- Donner les allures approximatives de  $h(n)$  et  $|H(f)|$  puis en déduire le rôle du filtre
- Déterminer  $H(z)$  (On supposera un gain de 1 en  $z = -1$ )
- Déterminer les coefficients du filtre
- On suppose que  $f_e = 3$  kHz, quelle sera la sortie du filtre si l'on donne en entrée un signal bruité par une sinusoïde de 500 Hz :



un signal composé de 2 sinusoïdes l'une de 1100 Hz et l'autre de 1300Hz

12. On suppose le filtre représenté par le tracé des pôles et des zéros suivant

- Caractériser le filtre RII ou RIF
- Quel est son rôle ?
- Donner les pôles et les zéros :
- Donner les allures approximatives de  $h(n)$  et  $|H(f)|$
- Déterminer  $H(z)$  (gain de 1 à  $f=0$ )
- Déterminer les coefficients du filtre
- Esquisser la TF de la sortie  $Y(f)$  si la TF de l'entrée  $X(f)$  est la suivante:



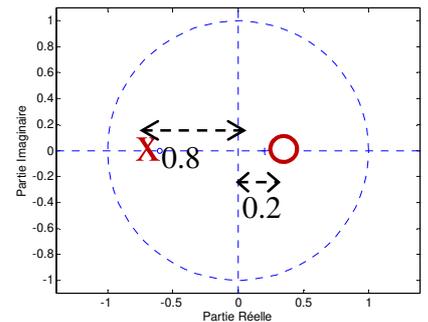
13. On suppose le filtre défini par l'équation aux récurrences suivante:

$$y(n) = 0.4y(n-1) - 0.16y(n-2) + 0.5x(n) + x(n-1) + 0.5x(n-2)$$

- Caractériser le filtre RII ou RIF
- Déterminer  $H(z)$  puis donner les pôles et les zéros
- Le retard de groupe est-il constant ?
- Donner les allures approximatives de  $h(n)$  et  $|H(f)|$ , en déduire le rôle de ce filtre.
- Quelle serait la sortie  $y(n)$  si  $x(n) = 3 + \exp(\pi j n / 2) + 3 \exp(\pi j n / 4)$ . Prendre  $f_e = 1$ .

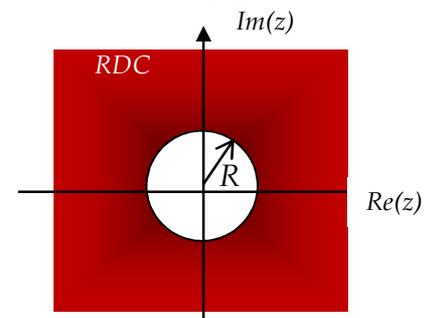
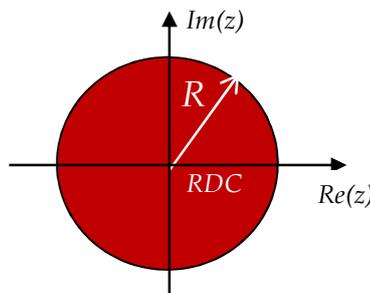
14. Soit le tracé des pôles et zéros suivant :

- Identifier le filtre (RII ou RIF) : (Justifier)
- Déterminer les pôles et les zéros
- Donner l'allure approximative de  $|H(f)|$  en précisant les valeurs sur les 2 axes
- En déduire le rôle du filtre :
- Est-il stable (Justifier) ?



15. Etudier la stabilité des systèmes suivants dont le RDC est donné comme suit :

- Pour  $R=2$
- Pour  $R=0.8$



**Solutions**

7 à 15 voir interros et examens années précédentes

**TP n°2: Analyse des filtres numériques par la TZ (sous Matlab)**

**Rappel :** Soit  $H(z)$  la transformée en  $z$  d'un filtre numérique donné dont la décomposition sous forme fraction rationnelle est donnée par :

$$H(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_N z^{-N}}{1 + a_1 z^{-1} + \dots + a_M z^{-M}}$$

Grâce à la seule connaissance du vecteur  $b$  et du vecteur  $a$ , on peut analyser tout filtre et :

- Déterminer les pôles et les zéros du filtre (et étudier sa stabilité)
- Déterminer la réponse impulsionnelle ou indicielle
- Déterminer la réponse fréquentielle et le retard de groupe (dérivée de la phase), etc.

**Quelques fonctions utiles[16]**

$s = \text{filter}(b, a, e)$  : filtre numériquement les données stockées dans le vecteur  $e$  avec le filtre décrit à la fois par le vecteur  $b$  (coefficients du numérateur de  $H(z)$ ) et le vecteur  $a$  (coefficients du dénominateur de  $H(z)$ ) pour une entrée  $e$ . Il faut normaliser l'équation de telle sorte que  $a_0 = 1$ .

Pour déterminer la réponse indicielle  $e$  sera un dirac, pour la réponse indicielle  $e$  sera un échelon.

$[H, f] = \text{freqz}(b, a, N, fe)$  : retourne  $N$  valeurs du gain complexe (Réponse fréquentielle TFD) du filtre numérique échantillonné à la fréquence  $fe$  (Hertz), décrit par  $b$  et  $a$ . Ces valeurs sont stockées dans  $H$  et calculées pour  $N$  fréquences mises dans  $f$ . Les fréquences sont equi-espacées sur l'intervalle  $[0, fe/2]$ .

$[b, a] = \text{invfreqz}(H, f, nb, na)$  : retourne le numérateur  $b$  (d'ordre  $nb$ ) et le dénominateur  $a$  (d'ordre  $na$ ) à partir de la réponse fréquentielle donnée par  $H$  et  $f$ .

$[h, n] = \text{impz}(b, a, N, fe)$  : retourne la réponse impulsionnelle du filtre numérique décrit par  $b$  et  $a$ . La réponse impulsionnelle est calculée en  $N$  instants stockés dans  $h$  et espacés de  $1/fe$ , les valeurs de réponse correspondante sont stockées dans  $h$ .

$[\text{tau}, f] = \text{grpdelay}(b, a, N, fe)$  : retourne le retard de groupe (dérivée de la phase) du filtre numérique décrit par  $b$  et  $a$ . La réponse impulsionnelle est calculée en  $N$  fréquences mises dans  $f$ .

$\text{zplane}(b, a)$  : permet de tracer les pôles et les zéros dans le plan complexe.

**I. Analyse d'un filtre RII**

Soit le filtre  $h(n)$  décrit par l'équation aux différences suivantes :

$$y(n) = 1.2 y(n-1) - 0.516 y(n-2) + 0.079 x(n) + 2 * 0.079 x(n-1) + 0.079 x(n-2)$$

- La première étape consiste à déterminer les vecteurs  $a$  et  $b$ . On calcule  $H(z)$  (coefficients en  $z^{-1}$ ) et on trouve : Numérateur :  $b = [0.079 \ 2 * 0.079 \ 0.079]$  et Dénominateur :  $a = [1 \ -1.2 \ 0.516]$
- Puis, par programme on peut : déterminer et tracer la réponse impulsionnelle, la réponse fréquentielle (module et phase), le retard de groupe, les pôles et les zéros, étudier la stabilité, la nature du filtre, etc.

```

clc ; clear all ; close all ;
b = [0.079 2*0.079 0.079]; %Numérateur
a = [1 -1.2 0.516]; %Dénominateur
figure (1); zplane(b,a);
N = 32; n=0:N-1; delta = [1; zeros(N-1,1)];
h = filter(b, a, delta); figure(2); stem(n,h);
echelon=ones(1,N); h_ind=filter(b,a,echelon);
figure(3); stem(n,h_ind);
L = 256; fe=1; [H, f] = freqz(b, a, L, fe);
module = abs(H); figure (4); plot(f,module);
% phase = angle(H); figure (5); plot(f,phase);
    
```

```
% [tau,f]=grpdelay(b,a,L,fe);figure (6); plot(f,tau);
% [num,den]=invfreqz(H,f, 2,2)
```

1. Calculer les pôles et zéros de ce filtre (à préparer), correspondent-ils à ceux de la figure 1 ?
2. A partir du tracé des pôles et des zéros, esquisser l'allure de  $h(n)$  et  $H(f)$  en justifiant vos réponses (à préparer). Confirmer avec les figures 2 et 4.
3. Etudier la stabilité du filtre (à partir du tracé des pôles et de  $h(n)$ ). Quel est le rôle de ce filtre?
4. Quelle valeur de  $b$  faut-il changer pour faire de ce filtre un passe-haut?
5. Modifier les valeurs de  $a$  pour avoir une réponse impulsionnelle divergente. Le filtre obtenu est-il stable?
6. Enlever les commentaires et comparer les figures 5 et 6. Quel lien les relie ?
7. Quel retard de groupe souhaite-t-on avoir dans la bande passante du filtre ?
8. Rétablir les valeurs par défaut et rajouter les lignes suivantes

```
nom_fich = uigetfile('*.wav', 'Selectionner le fichier son');
[x,fe]=wavread(nom_fich);
sound(x,fe); N=length(x);t=(0:N-1)/fe;
figure;subplot(2,1,1);plot(t,x);
legend('Son original');xlabel('Temps (s)');ylabel('Amplitude'); axis([0 N/fe -1 1.5]);
y = filter(b, a, x);sound(y,fe)
subplot(2,1,2);plot(t,y);
legend('Son filtré');xlabel('Temps (s)');ylabel('Amplitude'); axis([0 N/fe -1 1.5]);
```

9. Comparer les deux signaux en utilisant le zoom et commenter.
10. Prendre une petite portion du signal et observer sa TF avant et après filtrage en commentant.
11. Créer un signal composé de la somme de 2 sinusoides de fréquences 0.1 et 0.4 puis observer le signal avant et après filtrage.

## II. Analyse d'un filtre RIF

On considère la récurrence suivante :  $y(n)=0.5 x(n)-0.5 x(n-1)$

1. Déterminer  $h(n)$ , les pôles et zéros et esquisser  $H(f)$ . En déduire le rôle de  $H(f)$ , puis calculer le retard de groupe (en préparation).
2. Vérifier ces réponses par matlab.
3. Que peut-on dire sur la stabilité, la nature et le retard de groupe de ce filtre?
4. Si l'on remplace l'un des coefficients 0.5 par 1, que devient le retard de groupe?
5. Quelle serait la sortie d'un tel filtre si l'entrée était constante?
6. Rétablir les valeurs par défauts et rajouter les lignes concernant le fichier audio puis commenter.
7. Prendre la même portion du signal et observer sa TF avant et après filtrage en commentant.
8. Créer un signal composé de la somme de 2 sinusoides de fréquences 0.1 et 0.4 puis observer le signal avant et après filtrage.

9. Refaire le même travail pour  $y(n)=\frac{1}{5} \sum_{i=0}^4 x(n-i)$

## III. Autres manipulations

1. Tester une cellule passe-tout.
2. Tester la fonction residuez
3. Tester d'autres filtres vus en TD et comparer avec les résultats théoriques obtenus.

**TP n°2: Analyse des filtres numériques par la TZ (sous Python)**

**Rappel :** Soit  $H(z)$  la transformée en  $z$  d'un filtre numérique donné dont la décomposition sous forme fraction rationnelle est donnée par :

$$H(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_N z^{-N}}{1 + a_1 z^{-1} + \dots + a_M z^{-M}}$$

Grâce à la seule connaissance du vecteur  $b$  et du vecteur  $a$ , on peut analyser tout filtre et :

- Déterminer les pôles et les zéros du filtre (et étudier sa stabilité)
- Déterminer la réponse impulsionnelle ou indicielle
- Déterminer la réponse fréquentielle et le retard de groupe (dérivée de la phase), etc.

**Quelques fonctions utiles**

$s = \text{scipy.signal.lfilter}(b, a, e)$  : filtre numériquement les données stockées dans le vecteur  $e$  avec le filtre décrit à la fois par le vecteur  $b$  (coefficients du numérateur de  $H(z)$ ) et le vecteur  $a$  (coefficients du dénominateur de  $H(z)$ ) pour une entrée  $e$ . Si  $a_0 \neq 1$ ,  $b$  et  $a$  seront divisés par  $a_0$ .

$z, p, k = \text{scipy.signal.tf2zpk}(b, a)$  retourne les zéros, les pôles et le gain du filtre dont les coefficients du numérateur (respectivement dénominateur) sont donnés par  $b$  (respectivement  $a$ )

$f, H = \text{scipy.signal.freqz}(b, a, N, fe)$  : retourne  $N$  valeurs du gain complexe (Réponse fréquentielle TFD) du filtre numérique échantillonné à la fréquence  $fe$  (Hertz), décrit par  $b$  et  $a$ . Ces valeurs sont stockées dans  $H$  et calculées pour  $N$  fréquences mises dans  $f$ . Les fréquences sont equi-espacées sur l'intervalle  $[0, fe/2]$ .

$f, \text{Tau} = \text{scipy.signal.group\_delay}(b, a, N, fe)$  retourne le retard de groupe (dérivée de la phase) du filtre numérique décrit par  $b$  et  $a$ . La réponse impulsionnelle est calculée en  $N$  fréquences mises dans  $f$ .

$\text{zplane}(b, a)$  : permet de tracer les pôles et les zéros dans le plan complexe. `scatter(real(p), imag(p))`  
`scatter(real(z), imag(z))`

**I. Analyse d'un filtre RII**

Soit le filtre  $h(n)$  décrit par l'équation aux différences suivantes :

$$y(n) = 1.2 y(n-1) - 0.516 y(n-2) + 0.079 x(n) + 2 * 0.079 x(n-1) + 0.079 x(n-2)$$

- La première étape consiste à déterminer les vecteurs  $a$  et  $b$ . On calcule  $H(z)$  (coefficients en  $z^{-1}$ ) et on trouve :  
 Numérateur :  $b = [0.079 \quad 2 * 0.079 \quad 0.079]$  et Dénominateur :  $a = [1 \quad -1.2 \quad 0.516]$
- Puis, par programme on peut : déterminer et tracer la réponse impulsionnelle, la réponse fréquentielle (module et phase), le retard de groupe, les pôles et les zéros, étudier la stabilité, la nature du filtre, etc.

```
import numpy as np; import scipy.signal as sp; import matplotlib.pyplot as plt
from plot_zplane import zplane
b = np.array([0.079, 2*0.079, 0.079]); a = np.array([1, -1.2, 0.516]); z,p,k=zplane(b,a)
delta = np.zeros(32); delta[0] = 1; h=sp.lfilter(b,a,delta) ;
Echellon = np.ones(32); h_ind=sp.lfilter(b,a,Echellon)
plt.figure(2); plt.subplot(211); plt.stem(h, use_line_collection=True)
plt.title('Réponse impulsionnelle'); plt.grid(True); plt.ylabel('Amplitude')
plt.subplot(212); plt.stem(h_ind, use_line_collection=True)
plt.title('Réponse Indicielle'); plt.grid(True); plt.ylabel('Amplitude'); plt.show()
L = 256; fe=1; f,H= sp.freqz(b,a,L); f,Tau = sp.group_delay((b,a),f); f=f*0.5*fe/np.pi;
plt.figure(3); plt.subplot(311); plt.plot(f, np.abs(H))
plt.title('Module du Filtre'); plt.grid(True); plt.xlabel('Fréquence (Hz)'); plt.ylabel('Amplitude')
plt.subplot(312); plt.plot(f, np.angle(H))
plt.title('Phase du Filtre'); plt.grid(True); plt.xlabel('Fréquence (Hz)'); plt.ylabel('Amplitude')
plt.subplot(313); plt.plot(f, Tau)
plt.title('Retard de groupe du Filtre'); plt.grid(True); plt.xlabel('Fréquence (Hz)'); plt.ylabel('Amplitude'); plt.show()
```

1. Calculer les pôles et zéros de ce filtre (à préparer), correspondent-ils à ceux de la figure 1 ?
2. A partir du tracé des pôles et des zéros, esquisser l'allure de  $h(n)$  et  $H(f)$  en justifiant vos réponses (à préparer). Confirmer avec les figures 2 et 4.
3. Etudier la stabilité du filtre (à partir du tracé des pôles et de  $h(n)$ ). Quel est le rôle de ce filtre?
4. Quelle valeur de  $b$  faut-il changer pour faire de ce filtre un passe-haut?
5. Modifier les valeurs de  $a$  pour avoir une réponse impulsionnelle divergente. Le filtre obtenu est-il stable?
6. Comparer les figures 5 et 6. Quel lien les relie ?
7. Quel retard de groupe souhaite-t-on avoir dans la bande passante du filtre ?
8. Rétablir les valeurs par défaut et rajouter les lignes suivantes

```

from scipy.io import wavfile as wf; import winsound ;
from scipy.io.wavfile import write as wavwrite
fname = 'vousavezducourrierenattente.wav';
winsound.PlaySound(fname, winsound.SND_FILENAME)
fe, x = wf.read(fname);
Te=1/fe; N=len(x); t = np.linspace(0, N-1, N)*Te;
y=sp.lfilter(b,a,x) ;
plt.figure(4); plt.subplot(211);plt.plot(t,x,label="Signal Original")
plt.subplot(212);plt.plot(t,y,label="Signal Filtré");plt.xlabel('Temps (s)')
plt.title('Signal Audio Filtré');
zz=np.int8(y);
wavwrite("z.wav", fe, zz);
winsound.PlaySound("z.wav", winsound.SND_FILENAME);
    
```

9. Comparer les deux signaux en utilisant le zoom et commenter.
10. Prendre une petite portion du signal et observer sa TF avant et après filtrage en commentant.
11. Créer un signal composé de la somme de 2 sinusoides de fréquences 0.1 et 0.4 puis observer le signal avant et après filtrage.

## II. Analyse d'un filtre RIF

On considère la récurrence suivante :  $y(n)=0.5 x(n)-0.5 x(n-1)$

1. Déterminer  $h(n)$ , les pôles et zéros et esquisser  $H(f)$ . En déduire le rôle de  $H(f)$ , puis calculer le retard de groupe (en préparation).
2. Vérifier ces réponses par matlab.
3. Que peut-on dire sur la stabilité, la nature et le retard de groupe de ce filtre ?
4. Si l'on remplace l'un des coefficients 0.5 par 1, que devient le retard de groupe ?
5. Quelle serait la sortie d'un tel filtre si l'entrée était constante ?
6. Rétablir les valeurs par défauts et rajouter les lignes concernant le fichier audio puis commenter.
7. Prendre la même portion du signal et observer sa TF avant et après filtrage en commentant.
8. Créer un signal composé de la somme de 2 sinusoides de fréquences 0.1 et 0.4 puis observer le signal avant et après filtrage.
9. Refaire le même travail pour  $y(n)=\frac{1}{5} \sum_{i=0}^4 x(n-i)$

## III. Autres manipulations

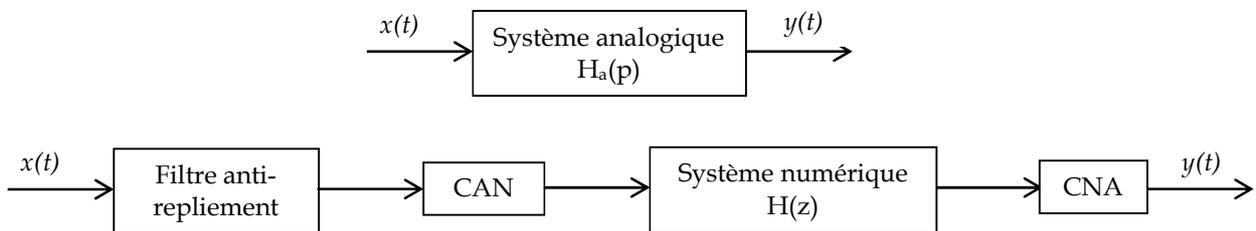
1. Tester une cellule passe-tout.
2. Tester d'autres filtres vus en TD et comparer avec les résultats théoriques obtenus.

### III. Conception des Filtres Numériques RIF

Le traitement numérique a conduit à une amélioration importante des dispositifs de filtrage linéaire notamment en termes de fiabilité, de reproductibilité, de souplesse et de complexité des fonctions réalisables. En outre, les filtres numériques ont aussi d'autres propriétés difficiles qu'il n'est pas aisé de mettre en œuvre dans le cas des filtres analogiques, entre autres : le filtrage numérique en temps réel (transmissions numériques, codage des sons MP3, synthèse de parole, télévision numérique par exemple). La plupart des modèles de filtres analogiques peuvent ainsi être reproduits sous forme numérique. Les éléments physiques (résistance, capacité, inductance, amplificateurs opérationnels) sont en quelque sorte transposés en éléments logiques [17].

#### 1. Synthèse des Filtres numériques et Gabarit analogique

La synthèse d'un filtre est un ensemble de processus qui débute par la définition des caractéristiques du filtre, jusqu'à sa réalisation informatique et/ou électronique, en passant par la détermination de ses coefficients. Pour synthétiser un filtre numérique, on considère connu le gabarit du filtre analogique et on cherche un système numérique caractérisée par une fonction de transfert  $H(z)$  à insérer dans le circuit ci-dessus permettant de satisfaire le gabarit analogique.



La détermination de la fonction de transfert d'un filtre numérique, par une méthode directe, n'est pas toujours très simple. Par contre, le problème qui consiste à transformer un filtre analogique en un filtre numérique est relativement simple. De ce fait, de nombreuses méthodes sont proposées pour concevoir un filtre numérique à partir du filtre analogique équivalent. Dans tous les cas, la synthèse d'un filtre numérique est une approximation d'un filtre analogique idéal équivalent. Il est nécessaire de contraindre un certain nombre de paramètres.

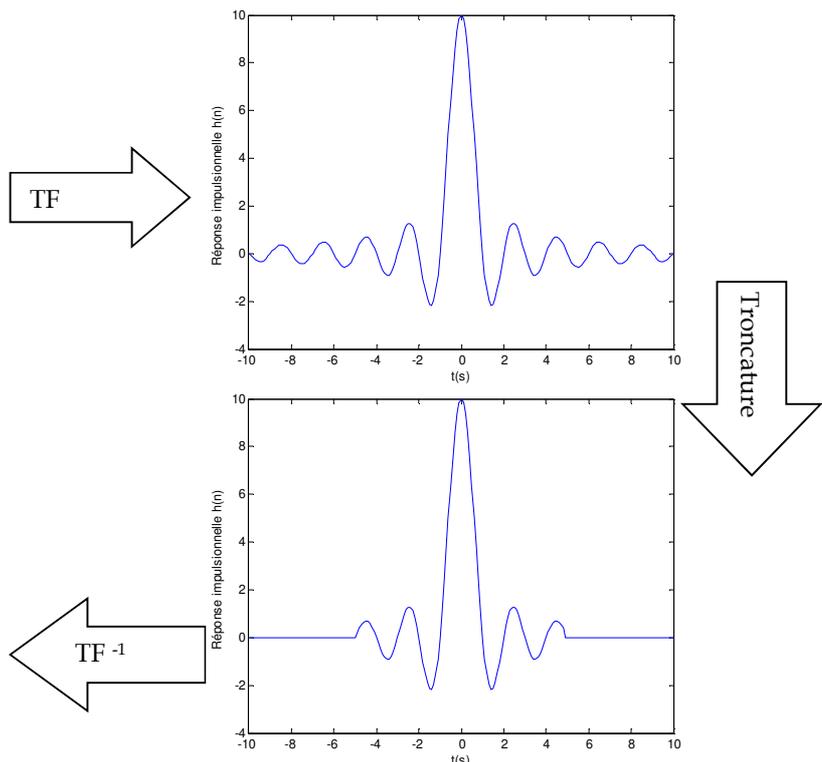
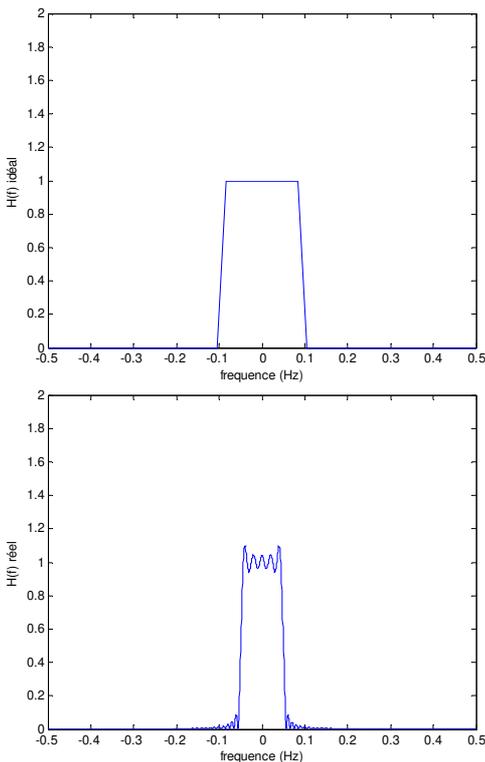
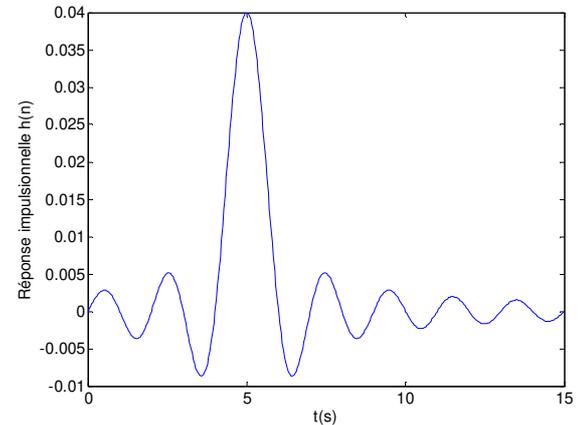
La synthèse d'un filtre numérique comprend les étapes suivantes :

1. la détermination d'une réponse en fréquence idéale souhaitée ;
2. la détermination de la meilleure approximation sous un certain nombre de contraintes (stabilité, rapidité, précision, déphasage linéaire, etc.) ;
3. le choix d'une structure de calcul réalisant le filtre approximé.

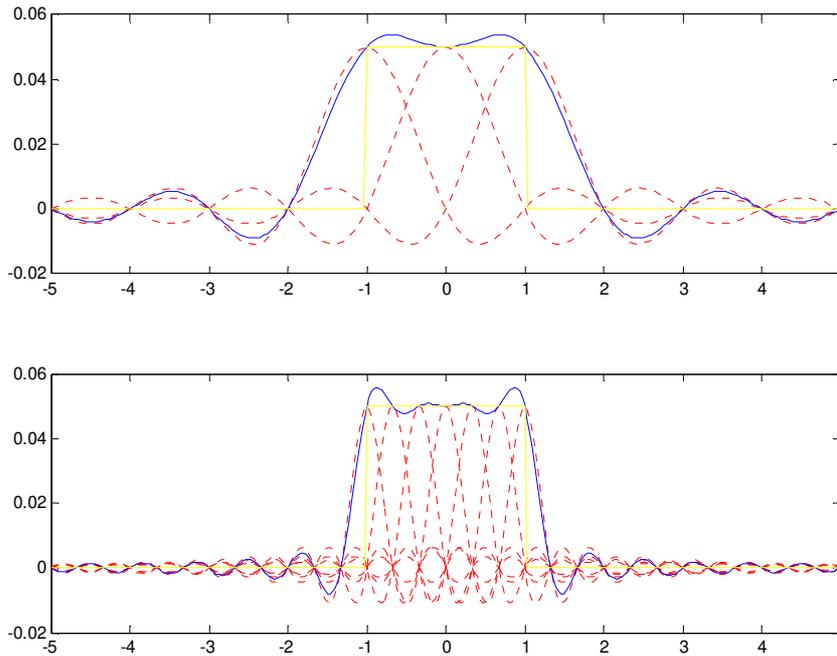
Les fonctions modèles utilisées pour la synthèse des filtres sont soit la réponse impulsionnelle soit la réponse en fréquence (celle-ci est préférée) de filtres analogiques connus. Si l'on emploie la réponse impulsionnelle, les éléments  $h(n)$  de la réponse impulsionnelle numérique sont obtenus en calculant  $h(t)$ , la réponse impulsionnelle du filtre analogique, aux instants  $t=nT_e$ .

Rappelons que les filtres idéaux présentent un déphasage linéaire et ne sont pas physiquement réalisables, car les réponses fréquentielles idéales correspondent à une réponse temporelle non-causale. Par exemple, en considérant le filtre passe-bas  $H(f) = \Pi(f) e^{-2\pi j f T}$ , on obtiendra un sinc décalé de  $T$ .

On peut observer qu'il faut annuler une partie du signal. De ce fait, il n'est plus possible d'obtenir un filtre passe-bas idéal (droit et avec une droite de transition perpendiculaire). Il s'ensuit que les filtres qui vont pouvoir être réellement synthétisés n'ont pas de réponse fréquentielle correspondant à la fonction porte, mais pourront s'en rapprocher.



Comme l'illustre la figure suivante, la troncature (multiplication par une porte de largeur  $N T_e$ ) du sinc dans le domaine temporel se traduira par une convolution dans le domaine fréquentiel du filtre idéal avec un sinc s'annulant tous les  $N T_e$ . Pour de grandes valeurs de  $N$ , les sinc dans la bande passante se compensent les uns les autres mais autour des points de discontinuité (fréquence de coupure), les ondulations restent apparentes.



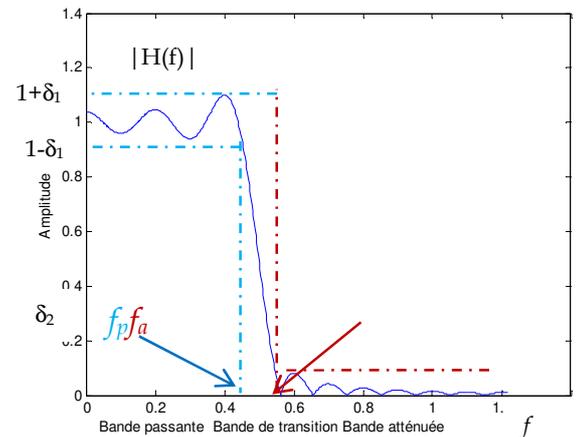
On peut observer que les différences vis-à-vis du filtre idéal (soit la fonction porte) sont principalement les ondulations dans la bande passante et dans la bande atténuée ainsi que la largeur de la transition.

C'est ainsi que les spécifications du filtre vont être définies par un gabarit fréquentiel linéaire ou en dB (décibels). Ce gabarit indique la ou les fréquences de coupure, la largeur de la bande de transition minimale souhaitée, le maximum d'ondulation de la bande passante et de la bande atténuée, la fréquence d'échantillonnage et éventuellement l'ordre maximal permis.

**Gabarit d'un filtre**

Le gabarit d'un filtre n'est autre que l'ensemble des caractéristiques du filtre, à savoir :

- la bande passante (BP) de 0 jusqu'à  $f_p$
- la bande atténuée (ou coupée BA) de  $f_a$  jusqu'à  $f_e/2$
- Le gain du filtre dans la bande passante.
- L'atténuation du filtre en bande coupée  $f_a$ .
- la largeur  $\Delta f = f_a - f_p$  de la zone de transition  $\Rightarrow f_c = f_a + \Delta f / 2 = (f_a + f_p) / 2$
- l'amplitude des oscillations en bande passante :  $\delta_1 \Rightarrow A_p = 20 \log(1 + \delta_1)$  ondulation permise en BP
- l'amplitude des ondulations en bande atténuée :  $\delta_2 \Rightarrow A_a = -20 \log(\delta_2)$  ondulation permise en BA



En pratique, plus les fréquences  $f_a$  et  $f_p$  sont proches, plus l'ordre du filtre devra être élevé. Pour un filtre idéal, ces valeurs seraient confondues

**2. Synthèse des filtres RIF**

L'emploi des filtres RIF peut se révéler attrayant eu égard à ses nombreux avantages : stabilité inconditionnelle (Tous les pôles sont en 0), phase linéaire possible. Néanmoins, ils présentent l'inconvénient de nécessiter un plus

grand nombre de coefficients que les filtres RII pour obtenir les mêmes caractéristiques fréquentielles à cause de l'absence de pôles hors 0. Ainsi, toute fonction de filtrage numérique stable et causale peut être approchée par la fonction de transfert d'un filtre RIF.

Rappelons que la sortie d'un filtre RIF va s'exprimer comme une combinaison linéaire d'un ensemble fini d'éléments d'entrée :

$$y(n) = \sum_{i=0}^{N-1} b_i x(n-i) \quad \text{d'où} \quad H(z) = \sum_{n=0}^{N-1} b_n \cdot z^{-n} \Rightarrow H(f) = \sum_{n=0}^{N-1} b_n \cdot e^{-2\pi j f n T_e}$$

Ainsi, les coefficients de pondération ne sont rien d'autre que les valeurs de la réponse impulsionnelle du filtre. Ces coefficients constituent les coefficients du développement en série de Fourier de la fonction de transfert H(f) (voir TFTD chapitre 1)

Du fait qu'un filtre RIF possède une fonction de transfert polynomiale (non rationnelle), il ne peut être obtenu par transposition d'un filtre continu. Les deux méthodes les plus utilisées pour l'approximation des filtres numériques RIF sont alors:

- o Développement par série de Fourier : cette série est ensuite tronquée par des fonctions fenêtres pour limiter la réponse impulsionnelle. Les coefficients de Fourier coïncident avec les échantillons de la réponse impulsionnelle du Filtre.
- o Echantillonnage de la réponse fréquentielle : Cette méthode fait appel à la TFD. Celle-ci est appliquée aux coefficients recherchés  $b_i$  pour obtenir une suite fréquentielle qui corresponde à la réponse fréquentielle du filtre.

Il existe d'autres méthodes telles les méthodes d'optimisation qui sont basées sur la minimisation d'un critère d'erreur entre la courbe réelle et le filtre idéal [9].

### 3. Méthode de la fenêtre

Cette technique consiste, connaissant l'expression analytique H(f) de la réponse fréquentielle continue (dont la formulation mathématique connue) à approcher, à déterminer par utilisation de la transformée de Fourier à temps discret inverse, la réponse impulsionnelle. Cette réponse temporelle non causale obtenue sera retardée pour la rendre causale [18]. Ainsi :

1. A partir du gabarit idéal du filtre, on détermine les coefficients du filtre en limitant le calcul à N valeurs réparties symétriquement autour de n=0. Puis, on calcule de la TFTD inverse du filtre idéal qui nous permettra de retrouver les échantillons de la réponse impulsionnelle soient les coefficients du filtre :

$$h(n) = \begin{cases} \frac{1}{f_e} \int_{-f_e/2}^{f_e/2} H(f) e^{2\pi j f n T_e} df & N \text{ impair (Filtre de type I)} \\ \frac{1}{f_e} \int_{-f_e/2}^{f_e/2} H(f) e^{-j\pi f T_e} e^{2\pi j f n T_e} df & N \text{ pair (Filtre de type II)} \end{cases}$$

Remarques :

- Le filtre de type I est le plus employé (C'est celui que nous développons dans ce qui suit) hormis pour les différenciateurs.
- Le filtre de type II possède un zéro en -1 (Nombre de zéros est impair donc répartition de zéros en paires conjuguées + un zéro en  $f_e/2$  puisque une somme de cosinus) , il ne peut donc être employé pour un passe-haut ou un coupe-bande.
- Il existe des filtres de type III (N impair) et IV (N pair) permettant d'obtenir une réponse impulsionnelle anti-symétrique avec déphasage linéaire également (somme de sinus au lieu de cosinus). Ils possèdent tous deux des zéros en 1 (sin en  $f=0$  est nulle), leur utilisation n'est donc pas adaptée aux filtres passe-bas. En outre le filtre de type III possède également un 0 en -1, on ne peut l'employer que pour un passe-bande.

2. Cette méthode produit une série infinie de coefficients, on limite, alors la réponse impulsionnelle à N échantillons (troncature). Sachant que la troncature induit des ondulations, on peut faire appel aux fenêtres de pondération pour les atténuer. Ainsi, la réponse impulsionnelle idéale  $h(n)$  sera multipliée par la fenêtre discrète  $w_N(n)$  de longueur N:

$$h'_N(n) = h(n) \cdot w_N(n). \text{ (Choix de la fenêtre : voir chapitre 1)}$$

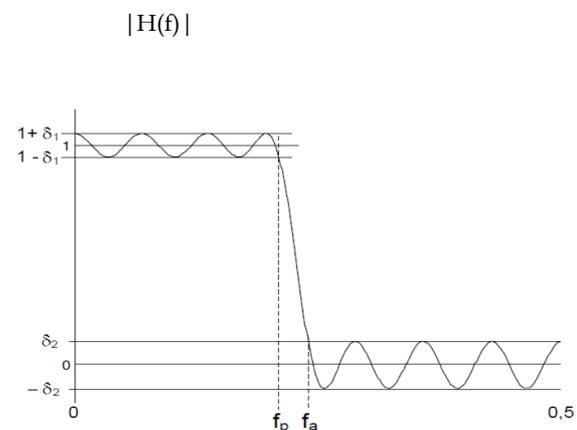
3. Il ne reste plus qu'à décaler la réponse impulsionnelle  $h(n)$  pour avoir une solution causale.

Remarque : Pour le choix de  $f_c$  il faudra faire attention aux fréquences de coupure à prendre en compte. Afin d'avoir de bons résultats lors de la synthèse, ce ne sont pas les fréquences de coupure du filtre idéal qu'il faut utiliser mais il faut déplacer celles-ci afin de les centrer dans la zone de transition. Pour un passe-bas, l'augmenter de la demi zone de transition ( $\Delta f$ ), soit  $f_p + \Delta f$  et pour un passe-haut, la diminuer de la demi zone de transition. Pour un passe-bande, diminuer la première fréquence de coupure de la demi zone de transition et augmenter la seconde de la demi zone de transition, pour un rejeteur de bande, on fera l'inverse.

Exemple :

- Calcul de la réponse impulsionnelle idéale (cas N impair)

$$h(n) = \frac{1}{f_e} \int_{-f_e/2}^{f_e/2} H(f) e^{2\pi j f n T_e} df$$



On pose  $f_c = (f_p + f_a) / 2 \Rightarrow h(n) = \frac{1}{f_e} \int_{-f_c}^{f_c} e^{2\pi j f n T_e} df = \frac{1}{2\pi j n f_e T_e} e^{2\pi j f n T_e} \Big|_{-f_c}^{f_c}$

f

$\Rightarrow h(n) = \frac{e^{2\pi j f_c n T_e} - e^{-2\pi j f_c n T_e}}{2\pi j n} = \frac{\sin(2\pi f_c n T_e)}{\pi n} = \frac{\sin(\pi n f_c / (f_e / 2))}{\pi n}$

On normalise les fréquences par rapport à  $f_e/2$  (ou  $f_e$ ) c.a.d que l'on remplace partout  $f_c$  par  $f_c/(f_e/2)$ , on obtient alors :

$h(n) = \frac{\sin(\pi n f_c / (f_e / 2))}{\pi n} \frac{f_c / (f_e / 2)}{f_c / (f_e / 2)} = f_c \frac{\sin(\pi n f_c)}{\pi f_c}$

Tout aussi facilement, on peut déterminer les réponses impulsionnelles d'un passe-haut ( $1-H(f)$ ), d'un passe-bande (différence de 2 passe-bas) et d'un coupe-bande donnés.

Les valeurs de la réponse impulsionnelle idéale  $h(n)$  sont données au tableau suivant. Les fréquences  $f_c$  indiquées dans ce tableau (fréquences de coupure désirées) s'expriment également en fréquences normalisées (divisées par  $f_e/2$ ).

Type de filtre	Réponse impulsionnelle $h(n)$	
	$h(n)$ pour $n \neq 0$	$h(n)$ $n=0$
Passe-bas	$f_c \frac{\sin(\pi n f_c)}{\pi f_c}$	$f_c$
Passe-haut	$-f_c \frac{\sin(\pi n f_c)}{\pi f_c}$	$1-f_c$
Passe-bande	$f_{c2} \frac{\sin(\pi n f_{c2})}{\pi f_{c2}} - f_{c1} \frac{\sin(\pi n f_{c1})}{\pi f_{c1}}$	$f_{c2} - f_{c1}$
Rejcteur de bande	$f_{c1} \frac{\sin(\pi n f_{c1})}{\pi f_{c1}} - f_{c2} \frac{\sin(\pi n f_{c2})}{\pi f_{c2}}$	$1-(f_{c2} - f_{c1})$

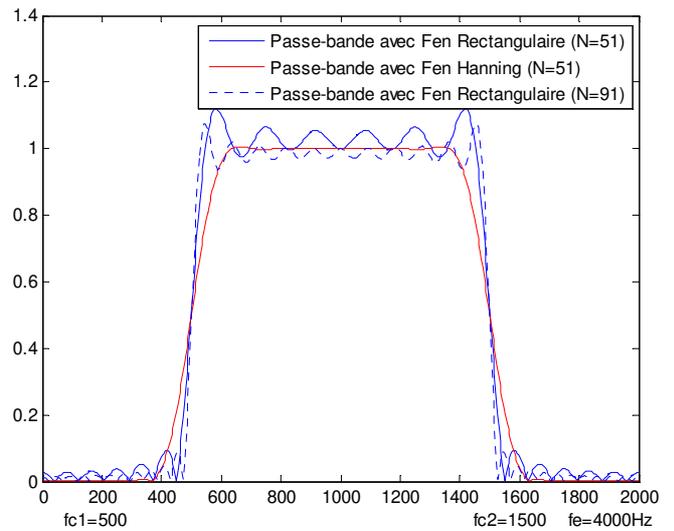
- Limitation du nombre d'échantillons à N

$h'_N(n) = h(n) \cdot w(n)$

si  $w(n) = \begin{cases} 1 & \text{si } |n| \leq \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases} \Rightarrow |W(f)| = \frac{\sin(N\pi f)}{\sin(\pi f)} \quad H'_N(f) = H(f) * W(f)$

La troncature temporelle introduit des ondulations et induit une zone de transition moins rapide déterminée par la largeur du lobe principal. Un compromis est à faire entre la raideur et l'amplitude des ondulations. Notons que cette méthode donne des ondulations de même amplitude dans la bande passante et dans la bande atténuée.

Remarque: Si N augmente, l'étendue des oscillations diminue (la réponse est plus plate) et la largeur de la bande de transition décroît ( $\Delta f$  faible). On peut observer que les oscillations les plus fortes tendent à se concentrer aux discontinuités. Ces dernières ne diminuent pas si N augmente : on a toujours un dépassement à peu près égal à 9% se concentrant aux points de discontinuité : c'est le phénomène de Gibbs (Voir TP n°3). Pour diminuer les oscillations : on utilise les fenêtres de pondération qui permettent d'obtenir de forte atténuation des oscillations mais cela se fait au détriment de la largeur de la bande de transition qui devient plus grande.

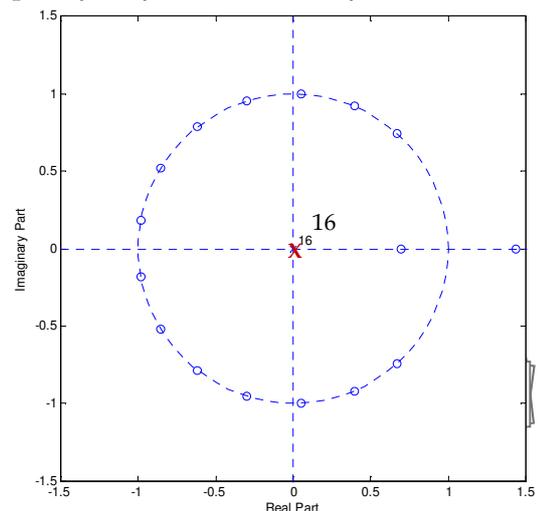


Pour le choix de la fenêtre de pondération, on procédera comme suit : En fonction de l'atténuation  $\delta_2$  requise dans la spécification du filtre, on choisira le type de fenêtre  $w(n)$  à utiliser. Puis en fonction de la largeur de la zone de transition  $\Delta f$  (spécifiée aussi au départ) et du type de la fenêtre  $w(n)$ , on déterminera la longueur de la réponse impulsionnelle N.

Fenêtres $w_N(n)$	Largeur de Transition : $\Delta f (2\Delta f / f_c)$	Atténuation en bande atténuée A
$w_{Rect}(n) = \begin{cases} 1 & \text{pour }  n  \leq \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases}$	$1.8/N$	21
$w_{Han}(n) = \begin{cases} 0.5 + 0.5 \cos\left(\frac{2\pi n}{N-1}\right) & \text{pour }  n  \leq \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases}$	$6.2/N$	44
$w_{Ham}(n) = \begin{cases} 0.54 + 0.46 \cos\left(\frac{2\pi n}{N-1}\right) & \text{pour }  n  \leq \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases}$	$6.6/N$	53
$w_{Black}(n) = \begin{cases} 0.42 + 0.5 \cos\left(\frac{2\pi n}{N-1}\right) + 0.08 \cos\left(\frac{4\pi n}{N-1}\right) & \text{pour }  n  \leq \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases}$	$11/N$	74

Exemple : On veut synthétiser un filtre passe-bas de fréquence de coupure  $f_c = f_e/10$  avec  $\Delta f = f_e/5$  et une ondulation en bande atténuée  $> 50$  db (voir TP n°3)

- a- On normalise les fréquence  $f_c / (f_e/2) \Rightarrow f_c = 0.2$   $\Delta f / (f_e/2) \Rightarrow \Delta f = 0.4$
- b- 
$$h(n) = 0.2 \frac{\sin(\pi n/5)}{\pi n/5}$$
- c- On choisit  $w(n)$  comme étant la fenêtre de Hamming

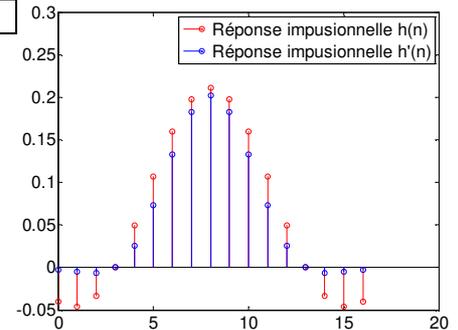


$$(A_a = 20 \log(\delta_2) = -53) \Rightarrow \Delta f = 6.6/N$$

d- On calcule  $N = 6.6/\Delta f = 16,5$  on prend  $N = 17$

e- on calcule les valeurs  $h(n) = 0.2 \frac{\sin(\pi n/5)}{\pi n/5}$  pour  $-8 \leq n \leq 8$

N	-8	-7	-6	-5	-4	-3	-2	-1	0
h(n)	-0,0399	-0,0456	-0,0329	0	0,0493	0,1064	0,1597	0,1974	0,2110
N	1	2	3	4	5	6	7	8	
h(n)	0,1974	0,1597	0,1064	0,0493	0	-0,0329	-0,0456	-0,0399	

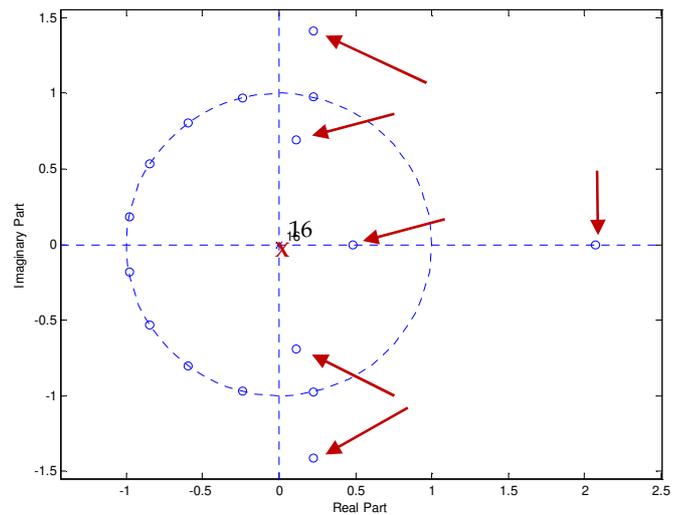
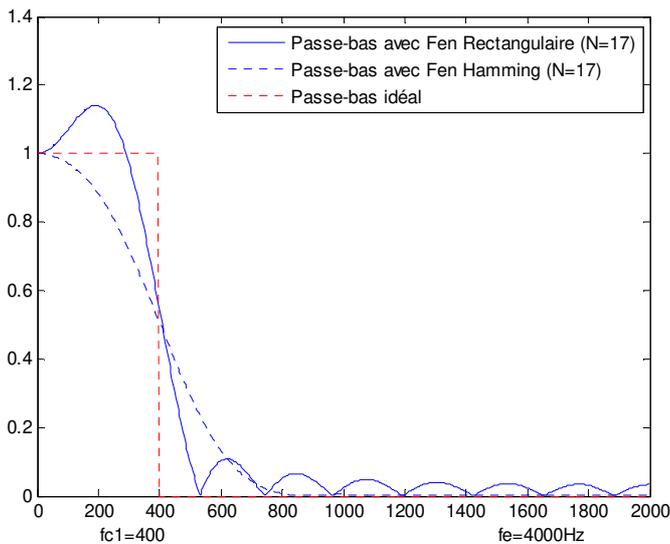


f- On multiplie h(n) par w(n) pour trouver  $h'_N(n) = h(n).w(n)$

$$h'_N(n) = 0.2 \frac{\sin(\pi n/5)}{\pi n/5} \left[ 0.54 + 0.46 \cos\left(\frac{2\pi n}{16}\right) \right]$$

g- On translate le résultat de 8 échantillons.

N	0	1	2	3	4	5	6	7	8
h(n)	-0,0399	-0,0456	-0,0329	0	0,0493	0,1064	0,1597	0,1974	0,2110
h'_N(n)	-0,0031	-0,0050	-0,0067	0	0,0255	0,0730	0,1325	0,1826	0,2023
N	9	10	11	12	13	14	15	16	
h(n)	0,1974	0,1597	0,1064	0,0493	0	-0,0329	-0,0456	-0,0399	
h'_N(n)	0,1826	0,1325	0,0730	0,0255	0	-0,006	-0,005	-0,0030	



On peut remarquer l'emplacement de zéros autour du premier 0 en 1 permet d'atténuer les lobes secondaire et de maintenir une réponse cste autour du zéro.

4. Méthode de l'échantillonnage fréquentiel

La méthode de synthèse par échantillonnage en fréquence est appliquée depuis la réponse fréquentielle d'un filtre continu idéal  $H(f)$  dont on ne connaît pas la formule mathématique (on ne peut alors calculer  $h(n)$  par TF inverse de  $H(f)$ ). On utilise alors la transformation de Fourier Discrète inverse. C'est-à-dire que l'on "échantillonne" la réponse désirée dans le domaine fréquentiel, on obtient  $N$  points de cette réponse fréquentielle auxquels on fait correspondre  $N$  points de la réponse temporelle équivalente obtenus par TFD inverse [18] comme suit :

On commence par échantillonner  $H(f)$  :

$$H(k) = H(f) \Big|_{f=k/N} \quad k = -(N-1)/2 \text{ à } (N-1)/2$$

puis on applique la TFD inverse 
$$h(n) = \frac{1}{N} \sum_{k=-(N-1)/2}^{(N-1)/2} H(k) e^{2\pi jkn/N}$$

Cette méthode de synthèse est très simple et permet de réaliser toute forme de filtre (chose qu'on ne peut réaliser avec la méthode précédente). Cependant, cette méthode de synthèse ne garantit que les points fréquentiels  $H(k)$ . Entre ces points, la valeur de  $H(f)$  n'est pas maîtrisée, il peut y avoir des oscillations qui ne sont pas également réparties avec un maximum d'erreur entre la réponse idéale et la réponse obtenue se situant autour de la bande de transition.

Pour obtenir la réponse en fréquence du filtre finalement obtenu, on peut par exemple appliquer une TFD à la réponse impulsionnelle  $h(n)$  de taille  $N$  obtenue, après avoir ajouté un grand nombre de zéros. Par ailleurs, du fait de l'emploi d'une TFD inverse sur  $N$  points, la réponse impulsionnelle  $h(n)$  obtenue est périodique de période  $N$  bien que la réponse impulsionnelle idéale souhaitée ne soit pas de durée limitée.

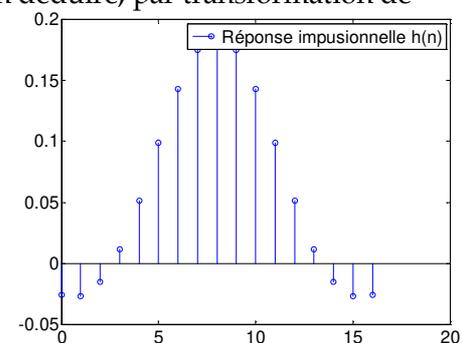
Exemple : On cherche à réaliser sous forme numérique un filtre passe-bas idéal de fréquence de coupure  $f_c = f_e/10$  avec  $\Delta f < f_e/16$ . On prend donc  $N=17$ , ce qui nous donne  $\Delta f = 0.0588$  (voir TP n°3)

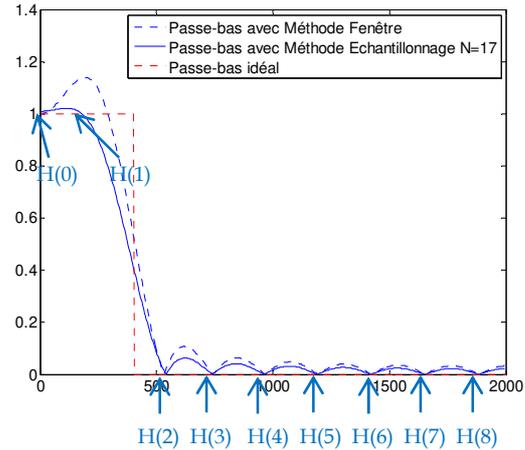
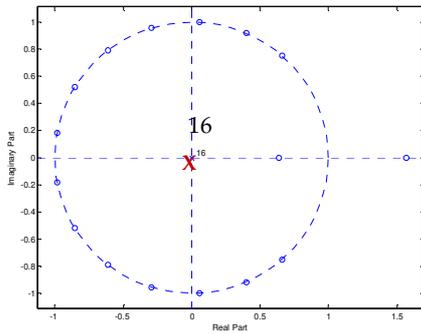
On a  $N = 17$ ,  $H(0)=H(-1)=H(1)=1$  et  $H(2)=H(-2)=\dots=H(8)=H(-8)=0$ . On peut en déduire, par transformation de Fourier discrète (TFD), les valeurs de la réponse impulsionnelle

$$h(n) = \frac{1}{N} \sum_{k=-(N-1)/2}^{(N-1)/2} H(k) e^{2\pi jkn/N} \quad h(n) = \frac{1}{17} (1 + e^{-2\pi jn/17} + e^{2\pi jn/17})$$

$$h(n) = \frac{1}{17} (1 + 2 \cos(2\pi n/17)) \text{ pour } -8 \leq n \leq 8$$

Enfin, pour rendre ce filtre réalisable physiquement, on translate cette réponse impulsionnelle de 8 échantillons.



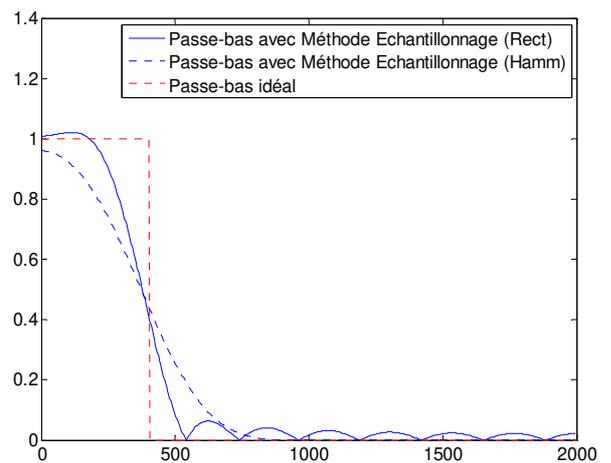
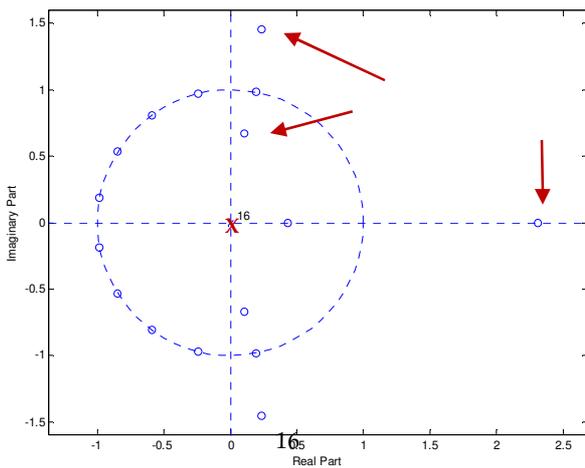


Sachant que  $H(k)^*=H(-k)$  pour un signal  $h(n)$  réel, on peut, de manière générale, démontrer que :

$$h(n) = \frac{1}{N} \left( H(0) + 2 \sum_{k=1}^{(N-1)/2} H(k) \cos\left(\frac{2\pi n k}{N}\right) \right) \quad \text{pour } -(N-1)/2 \leq n \leq (N-1)/2$$

La réduction des oscillations peut aussi s'obtenir à travers le fenêtrage. Ci-dessous les coefficients du filtre  $h(n)$  suivis de ceux obtenus après fenêtrage de Hamming  $h'_N(n)$ .

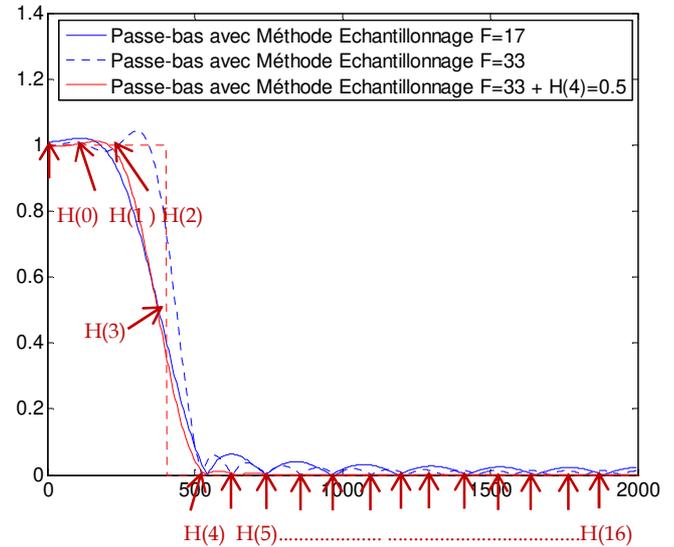
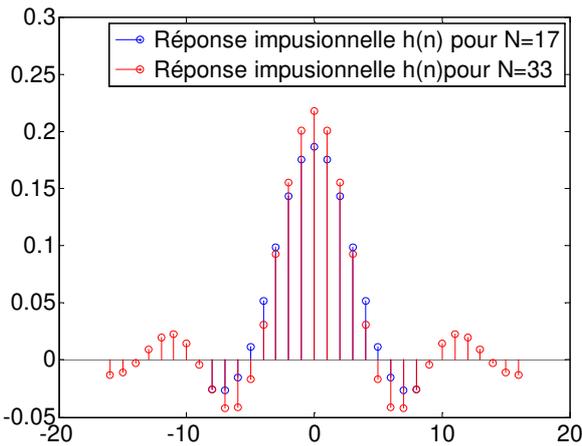
$n$	0	1	2	3	4	5	6	7	8
$h(n)$	-0,0257	-0,0269	-0,0153	0,0114	0,0514	0,0985	0,1430	0,1749	0,1865
$h'_N(n)$	-0,0020	-0,0031	-0,0033	0,0041	0,0279	0,0705	0,1237	0,1688	0,1865
$n$	9	10	11	12	13	14	15	16	
$h(n)$	0,1749	0,1430	0,0985	0,0514	0,01139	-0,0153	-0,0269	-0,0257	
$h'_N(n)$	0,1688	0,1237	0,0705	0,0279	0,0041	-0,0032	-0,0031	-0,0020	



Remarques: Pour le choix de  $N$ , il faut veiller à ce que  $f_e/N$  soit inférieur à  $\Delta f$ .

Par ailleurs, on peut quelques peu atténuer les ondulations en adoucissant les transitions dans la transmittance du filtre. Pour cela, on introduira 0.5 entre 1 et 0, Mais de ce fait  $\Delta f$  augmentera, on ajustera, alors, la valeur de N en conséquence ( $2\Delta f < f_e/16$ ). Ce qui nous fournira une valeur de N de 33 et  $\Delta f = 0.0303$ . On prendra  $H(3) = H(-3) = 0.5$ . Ainsi  $H(0) = H(-1) = H(1) = H(2) = H(-2) = 1$  et  $H(3) = H(-3) = 0.5$  et  $H(4) = H(-4) = \dots = H(32) = H(-32) = 0$ .

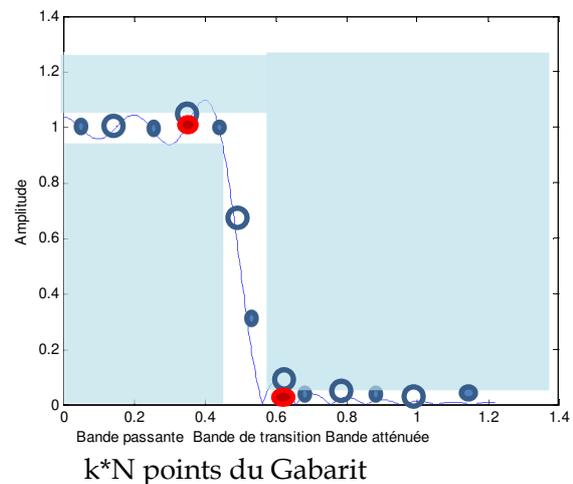
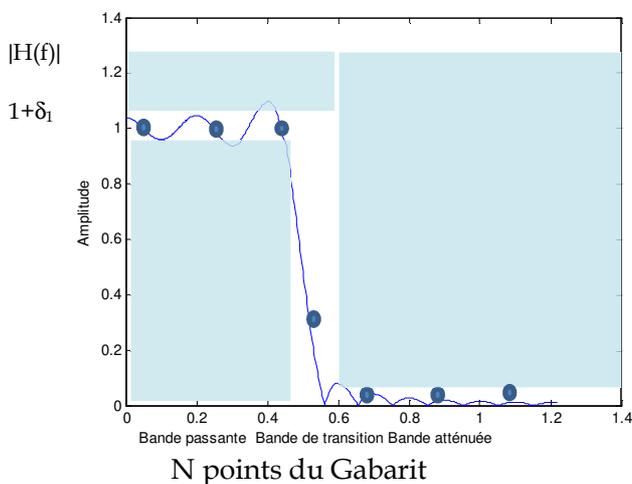
$$h(n) = \frac{1}{33} (1 + 2 \cos(2\pi n / 33) + 2 \cos(4\pi n / 33) + \cos(6\pi n / 33)) \text{ pour } -16 \leq n \leq 16$$



### 5. Méthode de Remez (Optimisation)

Le gabarit du filtre est supposé connu. On cherche les N coefficients de la réponse impulsionnelle du filtre RIF dont la réponse en fréquence  $H(f)$  entre dans le Gabarit :  $h(k)$  pour  $k = 0, \dots, N-1$

Le principe consiste à mettre en œuvre une méthode itérative qui cherche à minimiser à chaque étape la « distance » du filtre au gabarit. Puis, à chaque étape on détermine et on calcule la réponse en fréquence par Transformée de Fourier Discrète. Puis, on utilise la technique du Zéro-Padding pour sur-échantillonner la réponse en fréquence et vérifier précisément la distance au gabarit.



Les points qui n'entrent pas dans le Gabarit sont déplacés pour obtenir une nouvelle fonction de transfert qui vérifie le Gabarit. Par transformée de Fourier inverse on obtient la réponse impulsionnelle de durée  $k \cdot N$  points d'un filtre qui entre dans le gabarit. On la tronque à  $N$  points pour obtenir un nouveau filtre et on vérifie qu'il entre dans le Gabarit en calculant la distance au Gabarit qui est la moyenne des écarts, on la désire la plus petite possible [9] ;

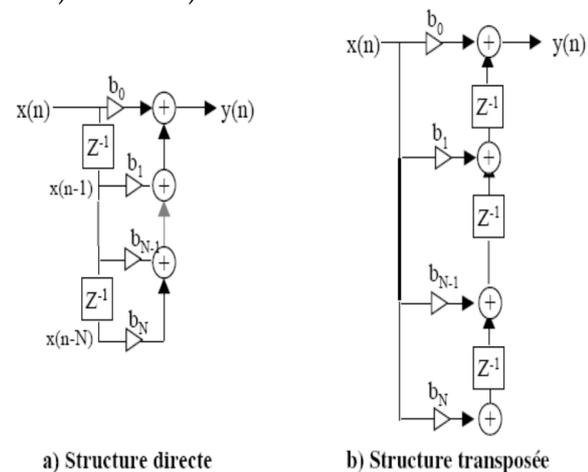
6. Constitution et réalisation des filtres numériques RIF

La réalisation concrète d'un filtre numérique consistera en fait à matérialiser l'algorithme de calcul pour la structure retenue. On aura la possibilité de travailler : Soit en logique câblée (assemblage d'organes logiques, tels que portes, mémoires, etc ...), soit en logique programmée (organisation autour d'un processeur de traitement du signal (DSP) ou, même, utilisation d'un microprocesseur(micro-ordinateur) standard).

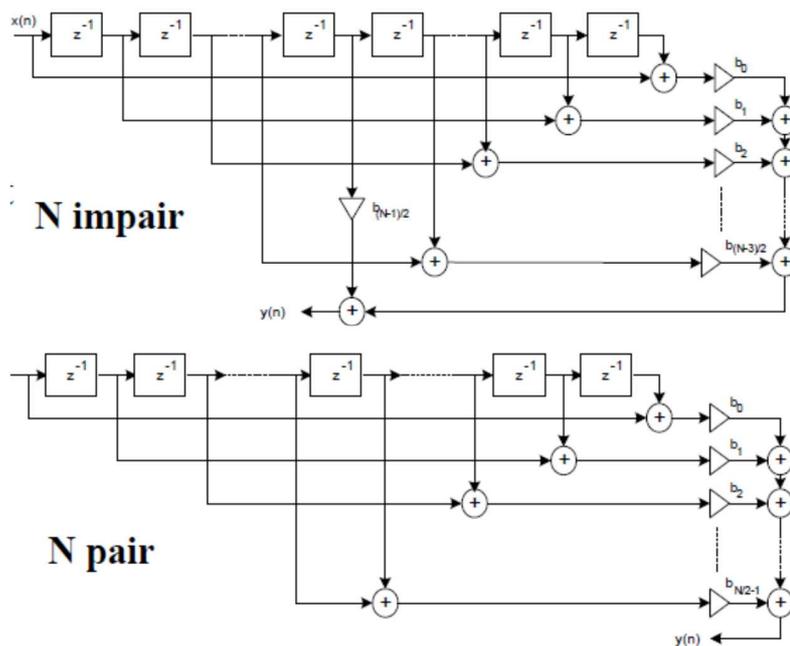
La structure canonique directe (transversale ou non réursive) est donnée ci-contre:

Un filtre RIF nécessite  $(N-1)$  opérations de multiplication,  $N$  opération d'addition pour chaque nouvel échantillon à filtrer.

On peut également exprimer la complexité en nombre de multiplication-accumulation qui, dans le cas d'un filtrage RIF, vaut  $N$  [9]. Le coût mémoire d'un filtrage RIF est de  $2(N+1)$  [( $N+1$ ) coefficients  $b_i$  et ( $N+1$ ) points mémoire pour le vecteur des entrées  $x(i)$ ]. Si la fréquence d'échantillonnage du signal d'entrée vaut  $f_e$ , cela signifie que le filtre devra être réalisé en un temps  $T_{calcul}$  inférieur à  $T_e = 1/f_e$ .



Pour un filtre à retard de groupe constant, la structure sera la suivante :



Série n° 3

1. Obtenir les coefficients d'un filtre à RIF passe-bas par la méthode de fenêtrage pour obtenir les spécifications suivantes :

- Fréquence de coupure idéale :  $f_c=1.75$  kHz
- Largeur de transition :  $\Delta f=0.5$  kHz
- Atténuation en bande atténuée :  $A=-20 \log_{10}(\delta) > 51$  dB avec  $\delta = \min(\delta_1, \delta_2)$
- Fréquence d'échantillonnage :  $f_e=8$  kHz

2. On souhaite approcher un filtre idéal passe-haut par un filtre à réponse impulsionnelle finie, synthétisé par la méthode du fenêtrage. Ce filtre doit répondre aux spécifications suivantes :

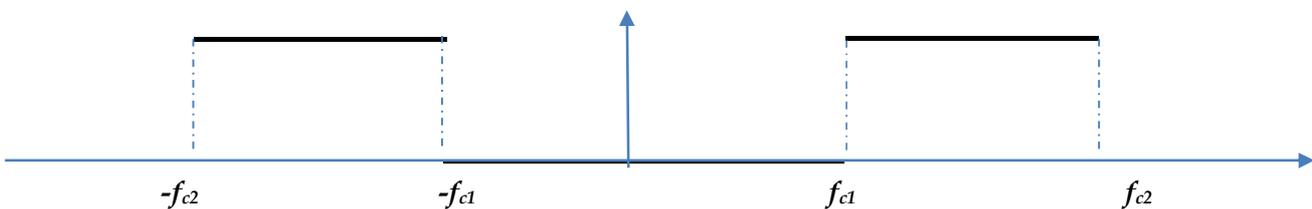
- ✓ Fréquence de coupure  $f_c=2$  kHz
- ✓ Largeur de transition :  $\Delta f=0.5$  kHz
- ✓ Atténuation en bande atténuée :  $A=-20 \log_{10}(\delta) > 40$  dB avec  $\delta = \min(\delta_1, \delta_2)$
- ✓ Fréquence d'échantillonnage :  $f_e=8$  kHz
- Déterminer l'expression mathématique exacte de  $h'(n)$ .
- Calculer  $h'(0)$  et tracer approximativement  $h'(n)$ .
- Quel est l'intérêt du fenêtrage et quel est son inconvénient ?
- Quel est l'inconvénient de cette technique de synthèse des filtres?
- Citer un avantage et un inconvénient de la synthèse par des filtres RIF de même que par les RII.

3. Soit un filtre passe-bas decoupure 0.25 et  $\Delta f < 0.08$  ( $f_e=1$ ). On souhaite synthétiser un filtre d'ordre par la méthode de l'échantillonnage fréquentielle.

4. Un filtre numérique est à implanter. La fréquence d'échantillonnage est fixée à 8kHz. Idéalement, on veut supprimer la bande fréquentielle de 1kHz à 2 kHz.

- Tracez, pour ce filtre idéal, le module de  $H(f)$
- Si on utilise la méthode d'échantillonnage en fréquence en prenant N points sur cette réponse idéale, quel serait  $h(n)$  pour  $N=21$ .
- Quelle est l'inconvénient de cette méthode? comment y remédier

5. On suppose le filtre  $h(n)$  dont le module du spectre  $H(f)$  est donné ci-dessous :



On veut déterminer le filtre numérique  $h(n)$  équivalent par la méthode de l'échantillonnage fréquentiel. Le filtre doit répondre aux spécifications suivantes  $f_{c1}=f_e / 5$ ,  $f_{c2}=2 f_{c1}$  et une largeur de transition  $\Delta f < f_e / 20$ . Prendre  $f_e=4,2$  KHz

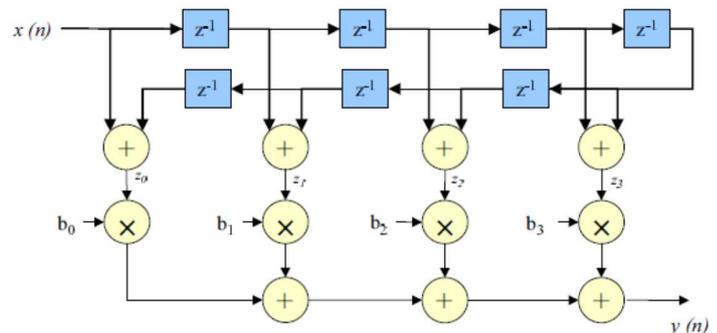
- Déterminer N,  $\Delta f$  puis  $H(k)$  et donner son tracé
- Déterminer  $h(n)$
- Si on utilise la méthode des fenêtres avec une atténuation en bande atténuée  $A_a > 40$  db et une zone de transition de  $\Delta f = f_e / 7$  donner l'expression de  $h(n)$  exacte (avec application numérique)

- Calculer  $h(0)$  avant translation dans chacun des 2 cas

6. On considère un filtre RIF de longueur  $N = 8$  dont la Structure est donnée ci-contre :

Avec  $b_0=0.2, b_1=0.3, b_2=0.4, b_3=0.5$

- Déterminer l'équation aux différences
- Déterminer la fonction de transfert puis  $h(n)$
- Identifier alors le type de filtre RIF (I,II,III ou IV)
- Quel est son inconvénient ?



**Solutions**

1.  $\Delta f=0.125, f_c = 0.4375, h(n) = f_c \frac{\sin(\pi n f_c)}{\pi n f_c} = \frac{\sin(\pi n 0.4375)}{\pi n}$  pour  $k \neq 0$  et  $h(0)=f_c$  hamming  $\Rightarrow N=53$

Les coefficients du filtre sont symétriques, il suffira par conséquent de calculer les valeurs de  $h(0)$  à  $h(26)$ .

$h(0) = f_c = 0.4375, w(0) = 0.54 + 0.46 \cos(0) = 1, h(0) = h(N(0)), w(0) = 0.4375$

$$w_{Ham}(n) = \begin{cases} 0.54 + 0.46 \cos\left(\frac{2\pi n}{N-1}\right) & \text{pour } |n| \leq \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases} = \begin{cases} 0.54 + 0.46 \cos\left(\frac{2\pi n}{52}\right) & \text{pour } |n| \leq 26 \\ 0 & \text{ailleurs} \end{cases}$$

$h(0)=0.4375, h(1)=h(-1)=0.311, h(2)=h(-2)=0.060, h(3)=h(-3)=-0.0856, h(4)=h(-4)=-0.053, h(5)=h(-5)=0.0325, h(6)=h(-6)=0.0434, h(7)=h(-7)=-0.0075, h(8)=h(-8)=-0.0319, \dots, h(26)=h(-26)=-0.0009.$

Pour rendre le filtre causal, on ajoute 26 à chacun des indices.

2.  $\Delta f=0.5, f_c = 0.125, h(n) = -f_c \frac{\sin(\pi n f_c)}{\pi n f_c}$  pour  $k \neq 0$  et  $h(0)=1-f_c, A > 40$  Hanning  $\Rightarrow N=51$

$h'(n) = -0.5 \frac{\sin(\pi n / 2)}{\pi n / 2} (0.5 + 0.5 \cos(\frac{2\pi n}{50}))$   $h'(0) = 0.5, -25 \leq n \leq 25$

3.  $f_c = 0.25, f_e = 1$  et  $N=13$  ainsi :  $(f_e/N=0.07, 2f_e/N=0.15, 3f_e=0.23, 4f_e/N=0.30) \Rightarrow H(0)=H(1)=H(-1)=H(2)=H(-2)=H(3)=H(-3)=1$  et les autres à 0

$h(n) = \frac{1}{N} \left( H(0) + 2 \sum_{k=1}^{(N-1)/2} H(k) \cos\left(\frac{2\pi kn}{N}\right) \right) = \frac{1}{13} \left( H(0) + 2 \sum_{k=1}^6 H(k) \cos\left(\frac{2\pi kn}{13}\right) \right) \prod_N(n)$

$h(n) = \frac{1}{13} \left( H(0) + 2 \cos\left(\frac{2\pi n}{13}\right) + 2 \cos\left(\frac{4\pi n}{13}\right) + 2 \cos\left(\frac{6\pi n}{13}\right) \right) \prod_{13}(n)$

4.  $f_{c1} = 1, f_{c2} = 2, f_e = 8$  et  $N=21$  ainsi :  $(f_e/N=381, 2f_e/N=762, 3f_e=1143, 4f_e/N=1523, 5f_e/N=1905, 6f_e/N=2286) \Rightarrow H(3)=H(-3)=H(4)=H(-4)=H(5)=H(-5)=0$  et les autres à 1.

$h(n) = \frac{1}{N} \left( H(0) + 2 \sum_{k=1}^{(N-1)/2} H(k) \cos\left(\frac{2\pi kn}{N}\right) \right) = \frac{1}{21} \left( H(0) + 2 \sum_{k=1}^2 H(k) \cos\left(\frac{2\pi kn}{21}\right) + 2 \sum_{k=6}^{10} H(k) \cos\left(\frac{2\pi kn}{21}\right) \right) \prod_N(n)$

$h(n) = \frac{1}{21} \left( H(0) + 2 \cos\left(\frac{2\pi n}{21}\right) + 2 \cos\left(\frac{4\pi n}{21}\right) + 2 \cos\left(\frac{12\pi n}{21}\right) + \dots + 2 \cos\left(\frac{20\pi n}{21}\right) \right) \prod_{21}(n)$

La TF de ce  $H(n)$ , va effectivement passer par ces points mais en dehors de ces points le comportement du filtre peut dévier du filtre idéal. Pour améliorer, il faudra prendre plus de points, d'où un filtre plus long.

5.  $N=21$ ,  $\Delta f=200$  Hz,  $H(0)=H(1)=H(-1)=\dots=H(4)=H(-4)=H(9)=H(-9)=H(10)=H(-10)=0$  Reste à 1

$$h(n) = \frac{2}{21} \left( \cos\left(\frac{10\pi n}{21}\right) + \cos\left(\frac{12\pi n}{21}\right) + \cos\left(\frac{14\pi n}{21}\right) + \cos\left(\frac{16\pi n}{21}\right) \right) \quad -10 \leq n \leq 10 \quad h(0)=0.38$$

Décaler les indices de 10

Hanning,  $N=23$ ,  $h(n)=0.8 \operatorname{sinc}(0.8n) - 0.4 \operatorname{sinc}(0.4n)$  et  $h'(n) = (0.8 \operatorname{sinc}(0.8n) - 0.4 \operatorname{sinc}(0.4n))(0.5 + 0.5\cos(2\pi n/22))\Pi_{23}(n)$

$h'(0)=0.4$

6.  $y(n)=b_3(x(n-3)+x(n-4))+b_2(x(n-2)+x(n-5))+b_1(x(n-1)+x(n-6))+b_0(x(n)+x(n-7))$

Type II, pas adéquat pour un passe-haut (un zéro en -1)

### Exercices supplémentaires

1. Au cours de la transmission d'un signal numérique (échantillonné à une fréquence de 2,5 kHz), il a été affecté d'un bruit localisé entre les bandes de fréquence 350 Hz et 550 Hz. On veut éliminer le bruit par l'emploi d'un filtre RIF possédant une bande de transition  $\Delta f=100$  Hz. Concevoir ce filtre :

A] Par la méthode du fenêtrage. On souhaite une atténuation en bande atténuée  $A=-20 \log_{10}(\delta) > 20$  dB.

- Tracer  $H(f)$  idéal
- Déterminer  $h(n)$  et l'ordre  $N$  du filtre
- Calculer  $h(0)$ ,  $h(1)=h(-1)$
- Tracer alors approximativement  $H(f)$

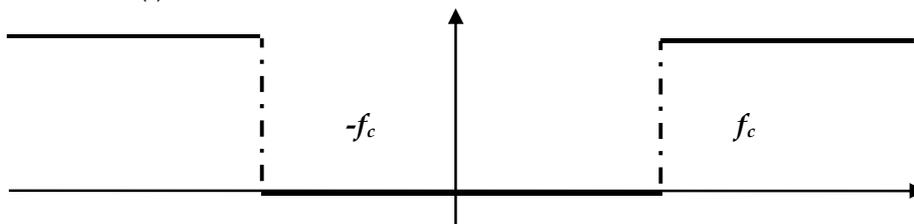
B] Par la technique de l'échantillonnage fréquentiel

- Déterminer  $N$
- Donner les valeurs de  $H(k)$  et son tracé
- Déterminer  $h(n)$
- Comparer ces 2 techniques de synthèse d'un filtre RIF

2. Au cours de la transmission d'un signal numérique (échantillonné à une fréquence de 5 kHz), il a été affecté par un bruit sinusoïdal de fréquence  $f_0=250$  Hz. On veut éliminer le bruit par l'emploi d'un filtre possédant une bande de transition  $\Delta f=\pm 50$  Hz à -3db. Concevoir un filtre RIF par la méthode du fenêtrage. On souhaite une atténuation en bande atténuée  $A=-20 \log_{10}(\delta) > 40$  dB avec  $f_{c1,2}=f_0 \pm \Delta f$

- Déterminer  $h(n)$  et l'ordre  $N$  du filtre
- Calculer  $h'(0)$

3. On suppose le filtre  $H(f)$  suivant :



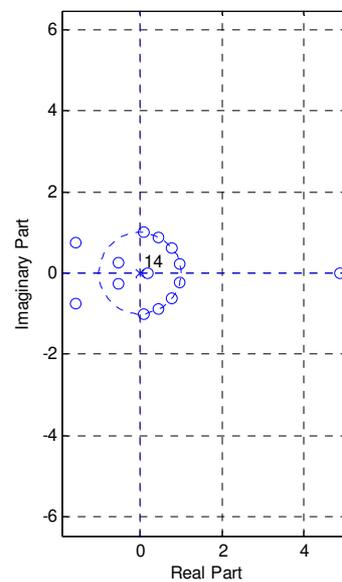
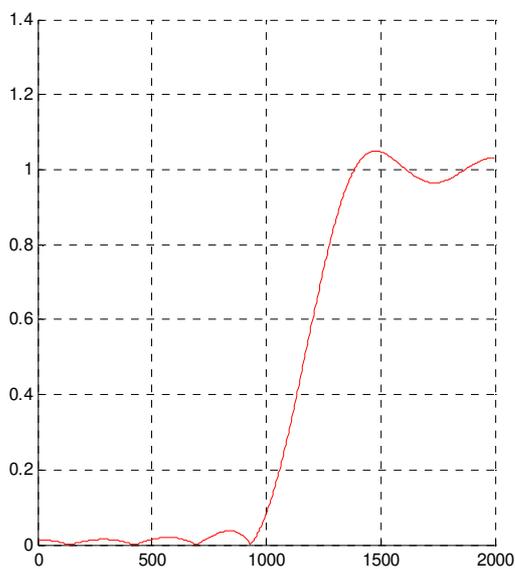
On veut déterminer le filtre numérique  $h(n)$  équivalent par

- a) la méthode de l'échantillonnage fréquentiel.
- b) la méthode des fenêtres

Le filtre doit répondre aux spécifications suivantes  $f_c = f_e/4$  et une largeur de transition  $\Delta f < f_e/16$

- Déterminer et tracer  $H(k)$  pour a)
- Déterminer  $h(n)$
- Tracer  $H(f)$  réel du filtre déterminé
- Comment sera le retard de groupe ? Justifier
- Quelle serait la sortie  $y(n)$  si  $x(n) = 1 + \cos(0.6\pi n) + 3 \cos(0.4\pi n)$ . Prendre  $f_e = 1$ .

4. Les tracés du module de la réponse en fréquence de 0 à  $f_e/2$  et des pôles et zéros d'un filtre numérique synthétisé par la méthode de l'échantillonnage fréquentiel sont donnés comme suit :



- Le déphasage de ce filtre est-il linéaire ? (Justifier)
- Déterminer  $H(k)$  puis la réponse impulsionnelle  $h(n)$
- Calculer  $h(0)$ .
- On veut employer une fenêtre de hanning. Que devient le tracé du module de la réponse en fréquence (le superposer à  $H(f)$ )

TP n°3: Synthèse des filtres RIF (Sous Matlab)

**But du TP :** Dans ce TP, on teste deux méthode différents pour synthétiser un filtre RIF: méthode de séries de Fourier (ou méthode des fenêtres) et méthode de la TFD dite par échantillonnage fréquentielle.

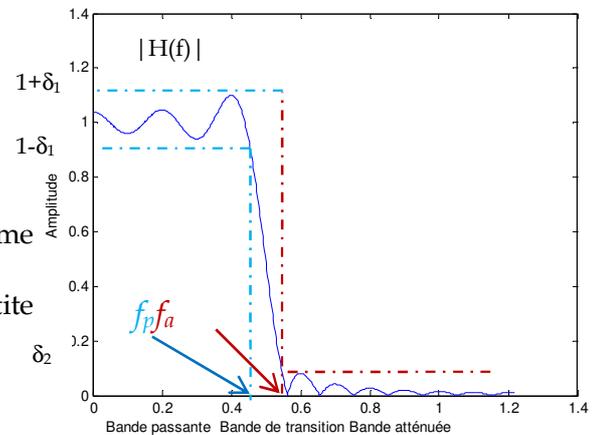
**I. Rappels**

Un filtre de réponse impulsionnelle finie (RIF) possède une fonction de transfert polynomiale. Il ne peut pas être obtenu par transposition d'un filtre continu, comme cela est fait pour les filtres RII. Les filtres RIF présentent l'inconvénient de nécessiter un grand nombre de coefficients pour obtenir les mêmes caractéristiques fréquentielles. Mais par contre, ils sont inconditionnellement stables. On peut synthétiser des filtres RIF à phase linéaire, c'est-à-dire à temps de propagation de groupe constant.

**Gabarit d'un filtre**

Le gabarit d'un filtre n'est autre que l'ensemble des caractéristiques du filtre, à savoir :

- Le gain du filtre dans la bande passante.
- L'atténuation du filtre en bande coupée  $f_a$ .
- La fréquence de coupure  $f_c$ , on l'exprime souvent sous forme normalisée par rapport à la fréquence d'échantillonnage.
- La largeur de bande de transition  $\Delta f$  souhaitée qui doit être la plus petite possible.
- Les éventuelles oscillations en bande passante et/ou atténuée.



- La détermination des coefficients d'un filtre RIF par la méthode de la fenêtre est réalisée par la fonction Matlab FIR1. Pour utiliser la technique d'échantillonnage de la réponse fréquentielle, on emploiera la fonction Matlab FIR2.

**II. Mise en avant du phénomène de Gibbs**

```
clc; clear all; close all;
Te=0.2; fe=1/Te; N=50; t = Te*(-N/2:N/2); N=length(t); h = Te*sinc(t);
figure; subplot(2,1,1); plot(t,h); grid; xlabel('t'); ylabel('Amp'); title('Sinc tronqué')
NF=2048; H=fft(h,NF); H=fftshift(H); axe_f=fe*(-1/2:1/NF:1/2-1/NF);
subplot(2,1,2); plot(axe_f, abs(H)); grid; xlabel('f'); ylabel('Amp'); title('TF Sinc tronq')
```

1. Identifier ce filtre
2. Expliquer la provenance des oscillations observées.
3. Utiliser le zoom pour mesurer le dépassement en bande passante (en %).
4. Déterminer  $f_a, f_p$  la largeur de la bande de transition puis  $f_c$
5. Prendre  $N=100$  puis  $500$  puis  $1000$  et mesurer à nouveau le dépassement et les 3 fréquences. Commenter.
6. Que représente  $N$ ? Quel est l'inconvénient d'augmenter  $N$ ?
7. Comment réduire les oscillations?
8. Choisir un fenêtrage et l'appliquer puis commenter.

**III. Méthode des fenêtres**

Pour synthétiser un filtre passe-bande, par la méthode des fenêtres, répondant aux spécifications suivantes:  $f_{c1} = 500$  Hz,  $f_{c2} = 1$  kHz,  $f_c = 4$  kHz,  $NbCoeff = 47$ , on procède comme suit :

```
clear all; close all; clc;
fc1 = 500 ; %Début de la BP
fc2 = 1000; %Fin de la BP
fe = 4000; %Fréquence d'échantillonnage
fc1N = fc1/(fe/2); fc2N = fc2/(fe/2); %Normalisation de fc1 etfc2
NbCoeff = 47; %Nombre de coefficients
N=NbCoeff-1 ; %Ordre du filtre
h = FIR1(N,[fc1N fc2N], 'band',rectwin(NbCoeff), 'scale') ;
[H F]= freqz(h, 1, 512, fe); % Calcul de H(f)
subplot(2,2,1);hold on; stem(h, 'b'); grid; %Tracé de la réponse impulsionnelle,
subplot(2,2,2);hold on;plot(F,abs(H)); grid; %Tracé du spectre d'amplitude
subplot(2,2,3);hold on; plot(F,20*log10(abs(H))); grid %Tracé du spectre d'amplitude en db
subplot(2,2,4);hold on; zplane(h,1);grid; %Tracé des pôles et zéros
```

1. Donner l'expression théorique de ce filtre.
2. Calculer  $\Delta f$  réelle sachant que  $\Delta f$  normalisée soit  $\Delta f / (fe/2)=1.8/N$  théorique puis déterminer la (pratiquement) à partir du graphe en employant les valeurs à  $1-\delta_1$  et la tangente à  $\delta_2$ .
3. Faites de même pour l'atténuation en db. Dépend-elle-de N?
4. Reprendre les questions 2 et 3 pour un fenêtrage de Hanning `window=('hann')`
5. Superposer les nouveaux graphes pour les 3 précédents graphes (figure 2) et commenter.
6. Comment réduire  $\Delta f$  pour une même fenêtre ? Quel est l'inconvénient de cette solution ?
7. Pour `NbCoeff = 23`, comparer la répartition des zéros pour les 2 fenêtres en établissant la relation avec les tracés des réponses en fréquence leur correspondant (utiliser des figures séparées)

#### IV. Synthèse RIF par échantillonnage fréquentiel

On veut synthétiser le même filtre passe-bande par la méthode de l'échantillonnage fréquentiel et comparer le résultat avec la méthode des fenêtres (garder l'ancien programme et rajouter)

```
Deltaf=fe/(NbCoeff);
AA = zeros(1,N/2+1);
for i=0:N/2
if (i*Deltaf>fc1 && i*Deltaf<fc2)
AA(i+1)=1;
end
end
FF = (0:1/(N/2):1); %frequence normalisée
h = fir2(N, FF, AA,rectwin(NbCoeff)); [H F] = freqz(h, 1, 512, fe);
subplot(2,2,1);hold on;stem(h, 'r');
subplot(2,2,2);hold on;plot(F,abs(H), 'r');
subplot(2,2,3);hold on;plot(F,20*log10(abs(H)), 'r');
```

1. Donner l'expression théorique de ce filtre.
2. Expliquer le rôle de la boucle.
3. Tester la fenêtre de Hanning également et comparer avec la fenêtre rectangulaire ( $\Delta f$  et dépassement).

#### V. Filtrage d'un ECG

Le signal dans `ecg.dat` contient plusieurs secondes d'un ECG échantillonné à 500 Hz. Les petites oscillations visibles principalement entre les complexes PQRST sont dues à la présence d'une interférence à 50 Hz, qu'on peut supprimer avec un filtrage passe-bas, mais en essayant de ne pas déformer les complexes.

1. Ecrire le programme permettant de réaliser un filtre RIF à phase linéaire (méthode des fenêtres)
2. Visualiser la sortie pour vérifier (utiliser `filter`).

TP n°3: Synthèse des filtres RIF (Sous Python)

**But du TP :** Dans ce TP, on teste deux méthode différents pour synthétiser un filtre RIF: méthode de séries de Fourier (ou méthode des fenêtres) et méthode de la TFD dite par échantillonnage fréquentielle.

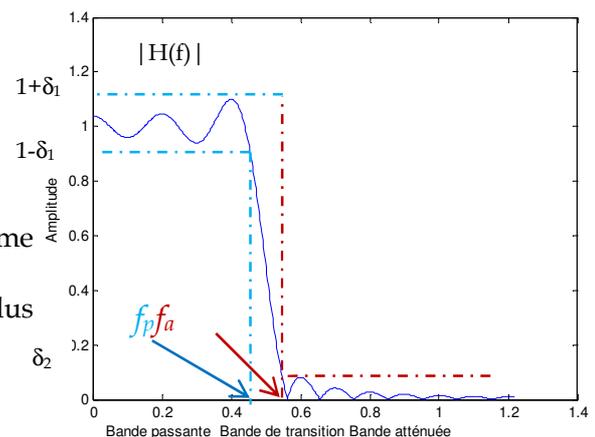
1. Rappels

Un filtre de réponse impulsionnelle finie (RIF) possède une fonction de transfert polynomiale. Il ne peut pas être obtenu par transposition d'un filtre continu, comme cela est fait pour les filtres RII. Les filtres RIF présentent l'inconvénient de nécessiter un grand nombre de coefficients pour obtenir les mêmes caractéristiques fréquentielles. Mais par contre, ils sont inconditionnellement stables. On peut synthétiser des filtres RIF à phase linéaire, c'est-à-dire à temps de propagation de groupe constant.

Gabarit d'un filtre

Le gabarit d'un filtre n'est autre que l'ensemble des caractéristiques du filtre, à savoir :

- Le gain du filtre dans la bande passante.
- L'atténuation du filtre en bande coupée  $f_a$ .
- La fréquence de coupure  $f_c$ , on l'exprime souvent sous forme normalisée par rapport à la fréquence d'échantillonnage.
- La largeur de bande de transition  $\Delta f$  souhaitée qui doit être la plus petite possible.
- Les éventuelles oscillations en bande passante et/ou atténuée.



- La détermination des coefficients d'un filtre RIF par la méthode de la fenêtre est réalisée par la fonction Python `firwin` de la bibliothèque `scipy.signal` . Pour utiliser la technique d'échantillonnage de la réponse fréquentielle, on emploiera la fonction `firwin2`.

2. Démo

# -\*- coding: utf-8 -\*-

```
import numpy as np; import scipy.signal as sp; import matplotlib.pyplot as plt
from plot_zplane import zplane
```

```
#Conception d'un filtre RIF passe-bas fc=200, fe =1000, Deltaf = 100, Aa>20
```

```
plt.figure(1)
fc=200;fe=1000; Deltaf=100
fcn=2*fc/fe ; Deltafn=Deltaf/(fe/2)
Ncoef = int(np.ceil(1.8/Deltafn))
n = np.arange(-(Ncoef//2),Ncoef//2+1)
b = fcn*np.sinc(n*fcn);
```

```
# #passe haut
# b = -fcn*np.sinc(n*fcn);
# b[Ncoef//2] += 1
```

```
#Visualisation de la réponse impulsionnelle et de la réponse fréquentielle
```

```
L=256
f,Hf= sp.freqz(b,1,L,fs=fe);
plt.subplot(211); plt.stem(b);
```

```
plt.subplot(212); plt.plot(f, np.abs(Hf))
plt.title('Module du Filtre'); plt.xlabel('Fréquence (Hz)'); plt.ylabel('Amplitude')
plt.legend()
```

```
fc=0.2; fe=1; fcn=2*fc/fe
A=np.array([21, 61, 101]); i=1;
a = np.array([1]);
L = 256;
plt.figure(2)
for N in A:
    n = np.arange(-(N//2), N//2+1)
    b = fcn*np.sinc(n*fcn);
    f, Hf = sp.freqz(b, a, L, fs=fe);
    plt.subplot(2,3,i); plt.stem(b);
    plt.subplot(212); plt.plot(f, np.abs(Hf), label='N=%d'%N)
    plt.title('Module du Filtre'); plt.grid(True); plt.xlabel('Fréquence (Hz)'); plt.ylabel('Amplitude')
    plt.legend()
    i+=1
```

```
#Effet du fenêtrage
from scipy.signal import windows
plt.figure(3)
N=41;
A=np.array([np.ones(N), windows.hann(N), windows.hamming(N) ]);
i=1;
for Fen in A:
    n1 = np.arange(-(N//2), N//2+1)
    b1 = fcn*np.sinc(n1*fcn)*Fen;
    f, Hf1 = sp.freqz(b1, a, L, fs=fe);
    plt.subplot(2,3,i); plt.stem(b1); plt.title('Filtre * Fen %d'%i);
    plt.subplot(212); plt.plot(f, np.abs(Hf1), label='Fen %d'%i)
    plt.title('Module du Filtre'); plt.grid(True); plt.xlabel('Fréquence (Hz)'); plt.ylabel('Amplitude')
    plt.legend()
    i+=1
```

```
#Effet du fenêtrage (affichage en db)
plt.figure(4)
fc=0.2; fe=2; fcn=2*fc/fe; N=41;
a = np.array([1]);
L = 256;
A=np.array([np.ones(N), windows.hann(N), windows.hamming(N) ]); i=1;
for Fen in A:
    n2 = np.arange(-(N//2), N//2+1)
    b2 = fcn*np.sinc(n2*fcn)*Fen;
    f, Hf1 = sp.freqz(b2, a, L, fs=fe);
    plt.subplot(2,3,i); plt.stem(b2); plt.title('Filtre * Fen %d'%i);
    plt.subplot(212); plt.plot(f, 20*np.log10(np.abs(Hf1)), label='Fen %d'%i)
    plt.title('Module du Filtre'); plt.grid(True); plt.xlabel('Fréquence (Hz)'); plt.ylabel('Amplitude')
    plt.legend()
    i+=1
```

```
#Appel de la fonction firwin
fc=0.2;fe=1; fcn=2*fc/fe; N=41
b1 = sp.firwin(N,fcn, window=('boxcar'),pass_zero='lowpass')
#ou
n = np.arange(-(N//2),N//2+1)
b = fcn*np.sinc(n*fcn);
#Verification
plt.figure(5)
plt.stem(b); plt.plot(b1)

#4 Types de Filtre RIF à retard de groupe constant
#b = np.array([0.0467745,0.10091, 0.151365, 0.187098, 0.2, 0.187098, 0.151365,0.10091, 0.0467745])
#b = np.array([0.0467745,0.10091, 0.151365, 0.187098, 0.187098, 0.151365,0.10091, 0.0467745])
# b = np.array([0.0467745,0.10091, 0.151365, 0.187098, 0, -0.187098, -0.151365, -0.10091, -0.0467745])
# b = np.array([0.0467745,0.10091, 0.151365, 0.187098, -0.187098, -0.151365, -0.10091, -0.0467745])

# plt.figure(6)
# #Détermination et Tracé des pôles et zéros
# a=np.array([1])
# z,p = zplane(b,a)

#Méthode de l'échantillonnage fréquentiel
fc=0.2;fe=1; N=41;
Deltaf=fe/N;
axef = np.linspace(0, 1, N//2+1)*fe/2;
#option 1
Hk = np.zeros(N//2+1);
for i in range(N//2+1):
    if (i*Deltaf<fc): #
        Hk[i]=1;
Hkinv= Hk[::-1]
HkTot=np.concatenate((Hk,Hkinv[:N//2]))
hn = np.fft.ifft(HkTot)
hn = np.fft.fftshift(hn)
#option 2
b2 = sp.firwin2(N,axef,Hk, window=('boxcar'),nyq=fe/2)

#Verification
plt.figure(7)
plt.stem(hn); plt.plot(b2)

#Comparaison
plt.figure(8)
L=256
f2,Hf2= sp.freqz(b2,1,L,fs=fe);
f1,Hf1= sp.freqz(b1,1,L,fs=fe);
plt.plot(f1,np.abs(Hf1),label='Passe-bas ave Méthode des fenêtres N=%d'%N);
plt.plot(f2,np.abs(Hf2),label='Passe-bas ave Méthode de l'échantillonnage fréquentiel N=%d'%N);
plt.title('Module du Filtre'); plt.grid(True); plt.xlabel('Fréquence (Hz)');plt.ylabel('Amplitude')
plt.legend()
```

#Pour la suite utiser firwin et firwin2

### 3. A faire

# -\*- coding: utf-8 -\*-

```
import numpy as np; import matplotlib.pyplot as plt
#from zp_plot import zplane
import scipy.signal as sp;

from scipy.io import wavfile as wf; import winsound ;
fname = 'vousavezducourrierenattente.wav';
winsound.PlaySound(fname, winsound.SND_FILENAME)
fe, X = wf.read(fname);
Te=1/fe; N=len(X); t = np.arange(0,N)*Te;
plt.figure(1);plt.subplot(211); plt.plot(t,X); plt.grid(True);
plt.xlabel('temps'); plt.ylabel('Amp'); plt.title('Phrase');
# Erre dans un fichier audio
# fnameb='Copie du signal.wav';
# Xb=X
# Xb=np.int8(Xb) ; wf.write(fnameb, fe, Xb);
# winsound.PlaySound(fnameb,winsound.SND_FILENAME) ;

#-----A faire-----
# Calculer et Visualiser sa TFD en db dans la figure (1)
# Rajouter du bruit ( 3 sinusoides avec une fréquence de 1500, 2000 et 3000)
# Visualiser la TFD du signal bruité en db
# Enregistrer le fichier avec wf.write dans fnameb
# Ecouter fnameb
# Concevoir Filtre par la méthode des fenêtres avec  $A_a > 50$  et  $\Delta f < f_c / 10$ ,  $f_c$  à déterminer
# en utilisant firwin
# Visualiser la réponse impulsionnelle et la fonction de Transfert du filtre
# Filtre le signal bruité Xb (sp.lfilter) avec le filtre synthétisé
# Enregistrer, écouter et visualiser la TFD du signal filtré
# Commenter
```

### IV. Synthèse des filtres récurrents RII

Le principe de définition du cahier des charges d'un filtre récurrent (RII) se déroule de la même manière que pour un filtre non récurrent (RIF) mais la synthèse se fera de différente manière. L'intérêt d'employer un filtre RII réside principalement dans la possibilité d'obtenir une bande de transition étroite pour un ordre raisonnable bien qu'il présente d'une part, un risque d'instabilité dû à une grande sensibilité numérique des coefficients mais que l'on peut toutefois contrôler en déterminant une structure mieux adaptée, et d'autre part, une variation de phase fortement non linéaire.

Rappelons que pour un filtre RII, la sortie s'exprime comme une combinaison linéaire d'un ensemble fini d'éléments d'entrées et de sortie:

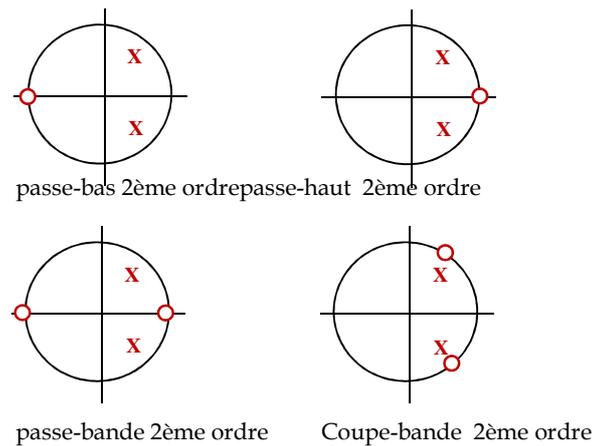
$$y(n) = \sum_{i=0}^N b_i x(n-i) - \sum_{i=1}^M a_i y(n-i) \quad \text{d'où} \quad H(z) = \frac{\sum_{i=0}^N b_i \cdot z^{-i}}{1 + \sum_{i=1}^M a_i \cdot z^{-i}}$$

La méthode directe consiste à placer des pôles et des zéros aux fréquences utiles mais la plus courante (indirecte dite aussi de transposition) est l'utilisation des méthodes de synthèse des filtres analogiques aboutissant à une fonction H(p) correspondant aux spécifications. Une fonction permettant le passage du plan p dans le plan z (p = fct(z)) est ensuite utilisée pour obtenir H(z). Cette fonction doit maintenir la stabilité du filtre analogique et maintenir, au mieux, les caractéristiques de la réponse fréquentielle H(f) du filtre numérique.

#### 1. Méthode des pôles et des zéros

Un filtre est rigoureusement caractérisé par la position de ses pôles et ses zéros dans le plan p (en continu) ou dans le plan z (en discret). Grâce à laquelle, on pourra déterminer la fonction de transfert du système à une constante près.

Rappelons, dans ce contexte que lorsqu'un zéro est placé sur un point donné du plan en z, la réponse fréquentielle présentera un minimum au point considéré. Un pôle quant à lui produira un pic (un maximum) au point correspondant qui sera d'autant plus important que le pôle est proche de 1.



De ce fait, on peut, par un placement adéquat des pôles et des zéros, obtenir un filtre sélectif simple.

La méthode se résume donc ainsi [11] :

- o A partir des spécifications du filtre, placer des pôles et des zéros sur le cercle (ou à partir des zéros et des pôles de H(p) calculer les zéros et les pôles de H(z)).

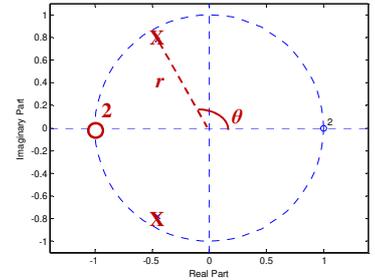
- o Si c'est un passe-bas, le caractère passe-bas est renforcé en complétant l'ensemble des zéros par autant de zéros que possible en  $z=-1$  tout en conservant à  $H(z)$  son caractère propre (degré du numérateur égal au degré du dénominateur). Si c'est un passe-haut, placer les zéros en  $z=1$ .
- o On ajuste le coefficient  $K$  pour obtenir un gain unitaire à  $z=1$  pour un passe-bas, à  $z=-1$  pour un passe-haut.

Il existe une relation "empirique" entre la largeur de la zone de transition  $\Delta f$  à 3 db et la position des pôles sur un rayon supérieure à 0.9:  $r \approx 1 - \frac{\Delta f_{3db}}{f_e} \pi$

- Pour un filtre passe-bas de second ordre, nous avons :

$$H(z) = \frac{K(z+1)(z+1)}{(z-re^{-j\theta})(z-re^{j\theta})} = \frac{K(z^2 + 2z + 1)}{z^2 - 2zr \cos \theta + r^2}$$

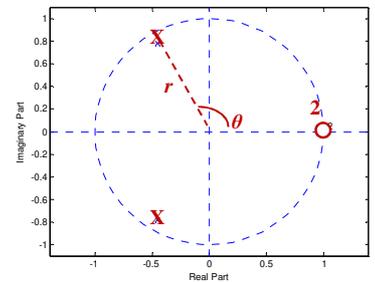
avec  $K = \frac{1+r^2 - 2r \cos \theta}{4}$   $\theta = (f_c / f_e) \times 360^\circ$  et  $r = 1 - \frac{\Delta f_{3db}}{f_e} \pi$



- Pour un filtre passe-haut de second ordre, nous avons :

$$H(z) = \frac{K(z-1)(z-1)}{(z-re^{-j\theta})(z-re^{j\theta})} = \frac{K(z^2 - 2z + 1)}{z^2 - 2zr \cos \theta + r^2}$$

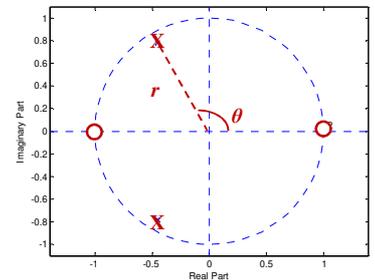
avec  $K = \frac{1+r^2 + 2r \cos \theta}{4}$



- Pour un filtre passe-bande de second ordre, nous avons :

$$H(z) = \frac{K(z-1)(z+1)}{(z-re^{-j\theta})(z-re^{j\theta})} = \frac{K(z^2 - 1)}{z^2 - 2zr \cos \theta + r^2}$$

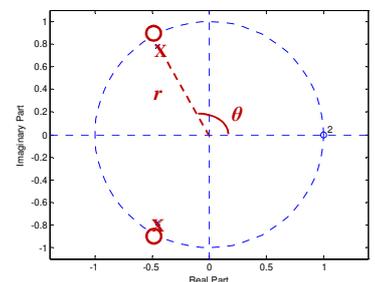
avec  $K = \frac{|e^{2j\theta} - 2r \cos \theta e^{j\theta} + r^2|}{|e^{2j\theta} - 1|}$



- Pour un filtre coupe-bande de second ordre, nous avons :

$$H(z) = \frac{K(z - e^{-j\theta})(z - e^{j\theta})}{(z - re^{-j\theta'})(z - re^{j\theta'})} = \frac{K(z^2 - 2 \cos \theta z + 1)}{z^2 - 2zr \cos \theta' + r^2}$$

avec  $K = \frac{1+r^2 - 2r \cos \theta'}{2 - 2 \cos \theta}$

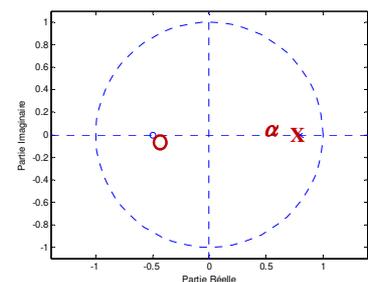


- Pour un filtre passe-bas de premier ordre, nous avons :

$$H(z) = \frac{K(z/\alpha + 1/\alpha)}{z - (1-K)/\alpha} \quad K = \text{tg}(\omega_c / 2) \quad \omega_c = 2\pi f_c / f_e \quad \alpha = 1 + K$$

- Pour un filtre passe-haut de premier ordre, nous avons :

$$H(z) = \frac{K(z/\alpha - 1/\alpha)}{z - (1-K)/\alpha}$$



**Exemple:** Trouver, grâce à la position des pôles et des zéros, la fonction de transfert et l'équation de récurrence d'un filtre numérique simple qui a les caractéristiques suivantes :

- Fréquence de coupure : 50 Hz
- Largeur de bande à 3 dB : ± 10Hz
- Fréquence d'échantillonnage : 500Hz

Pour rejeter les composants à 50Hz, on place une paire de zéros aux points du cercle unité qui correspondent à 50Hz. Ces points se situent à un angle de  $360^\circ \cdot 50 / 500 = \pm 36^\circ$ . Pour obtenir un filtre idéal rejecteur de bande et renforcer la réponse en amplitude de chaque côté de la fréquence de coupure, deux pôles complexes conjugués sont placés sur le cercle de rayon inférieur à 1 [18].

La largeur de la bande rejetée est déterminée par la position de pôles. La relation entre la largeur de bande et le rayon est donc applicable, par conséquent, la norme des pôles est 0,937.

Depuis la figure, la fonction de transfert du filtre est donnée par :

$$H(z) = K \frac{(z - e^{-j36^\circ})(z - e^{j36^\circ})}{(z - 0.937e^{-j36^\circ})(z - 0.937e^{j36^\circ})} = K \frac{1 - 1,618z^{-1} + z^{-2}}{1 - 1,516z^{-1} + 0.878z^{-2}}$$

Avec  $K = 0,947 \approx 1$ , l'équation de récurrence est :

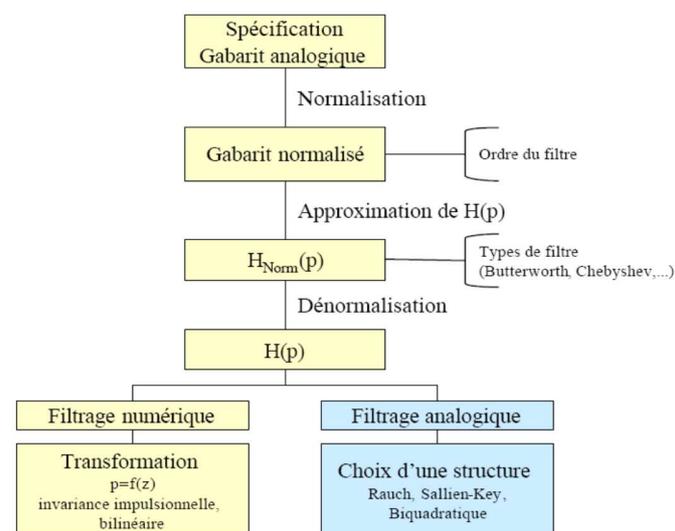
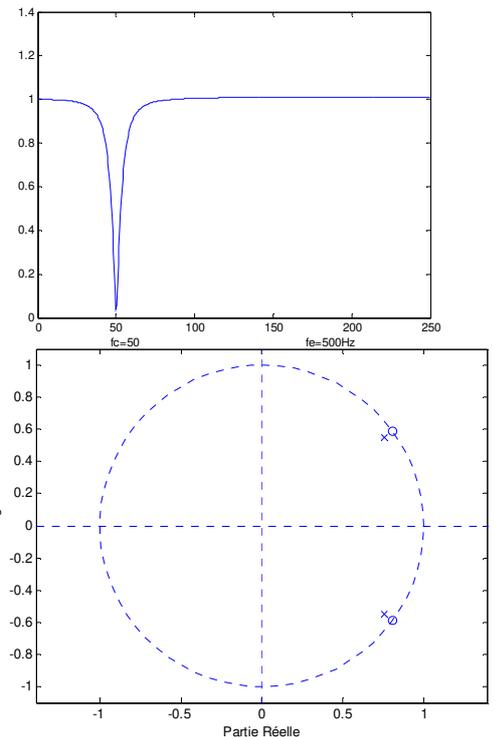
$$y(n) \approx x(n) - 1.618x(n-1) + x(n-2) + 1.5161y(n-1) - 0.878y(n-2)$$

## 2. Rappels sur les filtres analogiques

Il existe de nombreuses méthodes permettant de synthétiser un filtre numérique récursif à partir d'un filtre analogique pris comme modèle [18]:

- o le filtre doit avoir une réponse impulsionnelle ou indicielle imposée : ce sont les méthodes de l'invariance impulsionnelle et de l'invariance indicielle.
- o le filtre doit avoir une réponse fréquentielle entrant dans un gabarit donné : c'est la transformation bilinéaire.

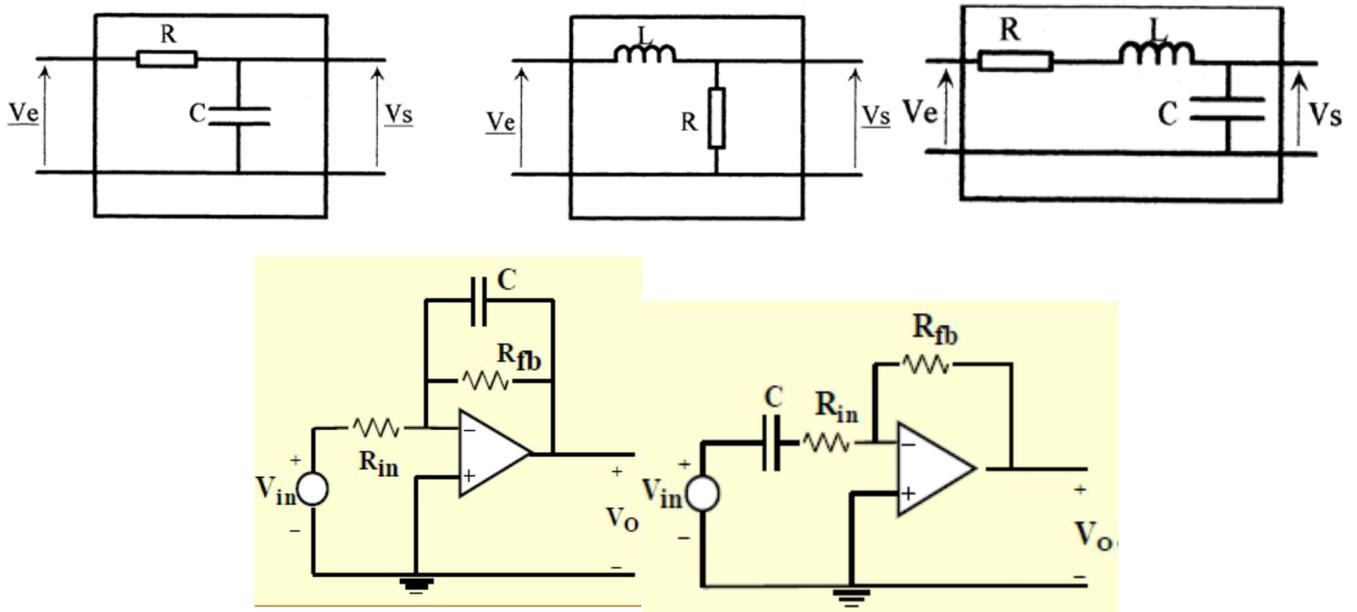
Pour concevoir un filtre analogique, on peut employer des filtres passifs obtenus par combinaison de résistances, de condensateurs et/ou de bobines ou encore utiliser des filtres actifs comportant un élément amplificateur (transistor, AO, etc.) qui permet donc de modifier les amplitudes des signaux. Ainsi, un filtre passif de base (de premier ordre) peut être composé d'une cellule RC ou d'une cellule RL.



Il faut noter que :

- les filtres numériques sont limités à des fréquences  $< 100\text{MHz}$
- les filtres passifs (L,C, quartz, etc ;) sont utilisés pour les hautes fréquences
- les filtres actifs (R,C, ALI) utilisent des amplificateurs linéaires intégrés (ALI) limités à  $1\text{Mhz}$
- les filtres à capacité commutés (R et C intégrés, ALI, Interrupteur commandé MOS) ont des fréquences ( $<10\text{ Mhz}$ ) programmables

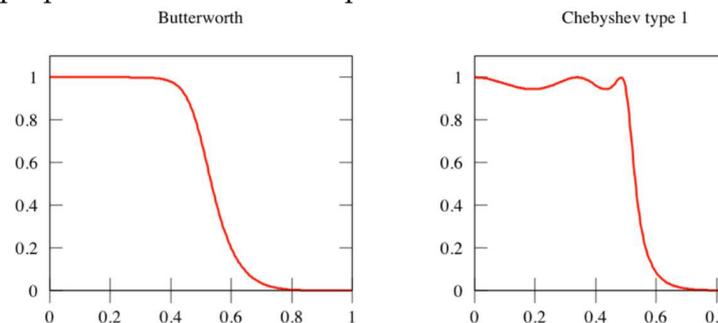
Ci-dessous, sont donnés quelques exemples de structure pour des filtres passifs passe-bas de premier et second ordre suivis de filtres actifs passe-bas de premier et second ordre aussi.



Pour obtenir des filtres d'ordre N plus élevé, on emploiera des filtres du premier et du deuxième ordremis en cascade. Les pentes asymptotiques seront proportionnelles au nombre de cellules ( $+N \uparrow$  et  $+\Delta f \downarrow$ ). A noter également que les filtres passe-bas ou passe-haut peuvent avoir un nombre entier d'ordre (1, 2, 3...) tandis que les filtres passe-bande ou coupe-bande ne peuvent qu'avoir un ordre pair (2, 4, 6, ...) car ils sont formés de paires de cellules : 2 cellules RC ou une cellule RC et une cellule RL.

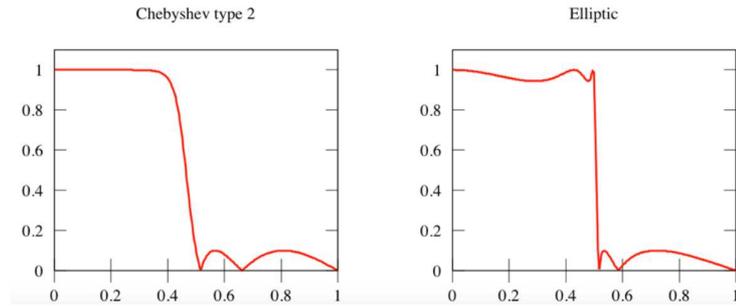
Pour synthétiser des filtres analogiques répondant à un gabarit donné, on choisira parmi un ensemble de filtres analogiques testés et éprouvés donc connus pour leurs propriétés en termes de pente d'atténuation et d'ondulation dans la bande passante et atténuée telles que :

- Filtres de Butterworth : Coupure peu raide mais gain constant en bande passante
- Filtres de Tchebychev (Chebyshev) :Raideur de coupure importante mais ondulations dans la bande passante



(Chebyshev 1) ou atténuée (Chebyshev 2) et simple à mettre en oeuvre

- Filtres de Caer (dit aussi elliptiques) : Coupure extrêmement raide mais ondulations dans la bande passante et atténuée mais circuits plus complexes à réaliser.
- Filtres de Bessel : Retard de groupe constant mais mauvaise sélectivité même pour ordre élevé.



Remarque : Le filtre de Bessel est intéressant pour filtrer les signaux large bande (modulations HF haut débit : débit : PSK, 8-PSK, OFDM...) en préservant les phases. Il n'a pas intérêt dans le domaine du filtrage numérique.

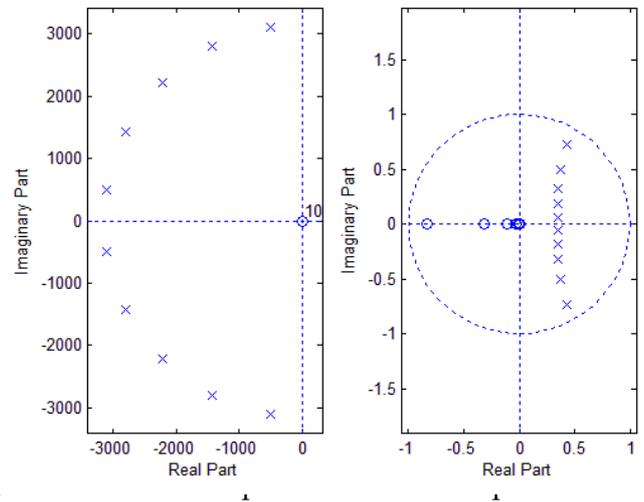
1. Un filtre de Butterworth est caractérisé par le fait que la réponse d'amplitude est maximale plate dans la bande passante et monotonement décroissante à partir d'une certaine fréquence (fréquence de coupure).

L'amplitude d'un filtre analogique Passe-bas de Butterworth d'ordre N est définie par l'expression :

$$|H(\omega)| = \frac{1}{\sqrt{1 + \left(\frac{\omega}{\omega_c}\right)^{2N}}} \quad \text{ou} \quad |H(\omega)| = \frac{1}{\sqrt{1 + \varepsilon^2 \left(\frac{\omega}{\omega_c}\right)^{2N}}}$$

$\omega_c$  est la pulsations de coupure limites et  $\varepsilon$  est un paramètre de conception qui fixe la région de tolérance dans la bande passante  $\delta_1 = 1/(1+\varepsilon^2)^{1/2}$ . Le filtre analogique aura N pôles placés sur le cercle unité (à partie réelle négative).

Ordre du filtre	Dénominateur (le numérateur est à 1)
2	$p^2 + \sqrt{2}p + 1$
3	$(p^2 + p + 1)(p + 1)$
4	$(p^2 + 1.8477p + 1)(p^2 + 0.7653p + 1)$
5	$(p^2 + 1.6180p + 1)(p^2 + 0.6180p + 1)(p + 1)$
6	$(p^2 + 1.9318p + 1)(p^2 + \sqrt{2}p + 1)(p^2 + 0.5176p + 1)$



L'ordre N est déterminé par la zone de transition ( $\omega_a - \omega_p$ ,  $\log(1+\delta_1)$  et bande atténuée ( $A_a = -\log\delta_2$ )

$$N \geq \frac{\log\left(10^{\frac{A_a}{10}} - 1\right)}{2 \log(\omega_a / \omega_p)}$$

$$N \geq \frac{\log\left(\left(10^{\frac{A_a}{10}} - 1\right) / \left(10^{\frac{A_p}{10}} - 1\right)\right)}{2 \log(\omega_a / \omega_p)} = \frac{\log\left(\left(\frac{1}{\delta_2^2} - 1\right) / \left(\frac{1}{\delta_1^2} - 1\right)\right)}{2 \log(\omega_a / \omega_p)}$$

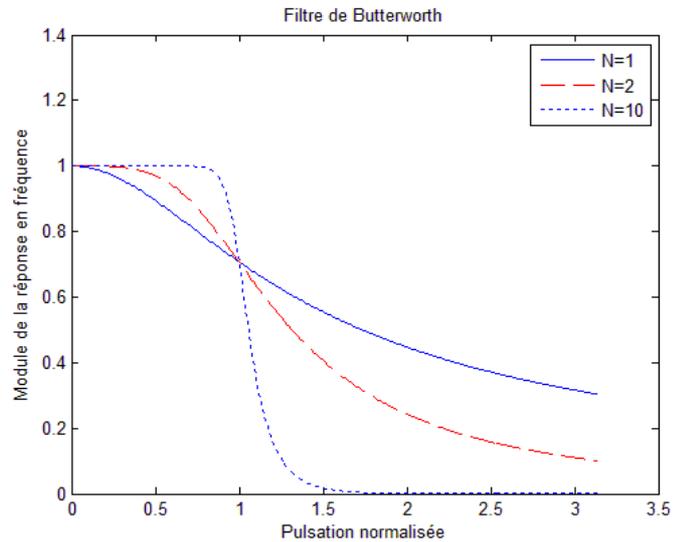
Remarque :

Plus N est élevé, plus grande est la sélectivité (zone de transition  $\Delta f$  étroite) mais un filtre RII de fait de sa récursivité a une période d'initialisation avant le régime permanent qui augmente avec l'ordre du filtre. En outre, son déphasage augmente avec l'ordre du filtre.

Exemple : On souhaite réaliser un filtre passe-bas de gain unitaire ne comportant pas d'oscillations ni dans la bande passante, ni dans la bande atténuée tel que :  $A_p=1\text{db}$ ,  $A_a=40\text{db}$ ,  $f_p=1\text{kHz}$  et  $f_a=3\text{kHz}$ .

Déterminer le filtre et N.

Filtre : Butterworth 
$$N \geq \frac{\log((10^4)/0.2589)}{2 \log(3/1)} = 4.80 \approx 5$$

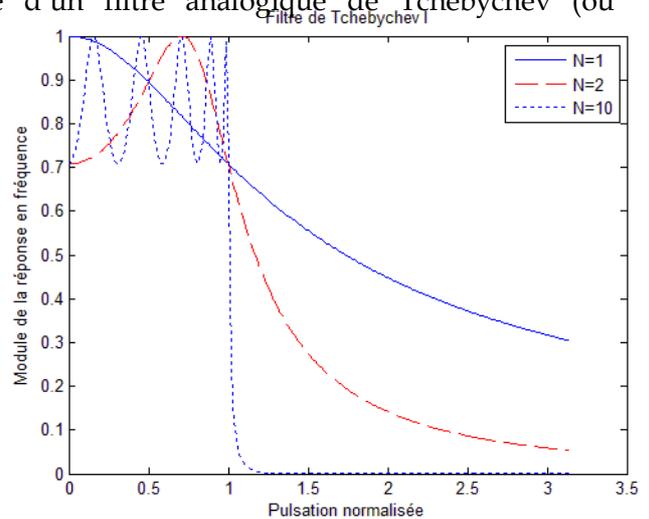


2. Un filtre de Tchebychev est caractérisé généralement par une ondulation équilibrée dans la bande coupée. La forme analytique du module de réponse fréquentielle d'un filtre analogique de Tchebychev (ou Chebyshev) d'ordre N est donnée par :

$$|H(\omega)| = \frac{1}{\sqrt{1 + \epsilon^2 T_N^2\left(\frac{\omega}{\omega_c}\right)}}$$

Où  $\epsilon$  est un paramètre réel inférieur à l'unité qui détermine l'amplitude des oscillations dans la bande passante et où  $T_N(x)$  est le polynôme de Tchebychev d'ordre N, définie par :

$$T_N(x) = \begin{cases} \cos[N \cdot \arccos(x)] & \text{pour } 0 \leq |x| \leq 1 \\ \cosh[N \cdot \operatorname{arccosh}(x)] & \text{pour } |x| \geq 1 \end{cases}$$



L'ordre N est déterminé par la zone de transition ( $\omega_a - \omega_p$ ) et les ondulations permises en bande passante  $A_p=20 \log(1+\delta_1)$  et bande atténuée ( $A_a = -\log \delta_2$ ) \*

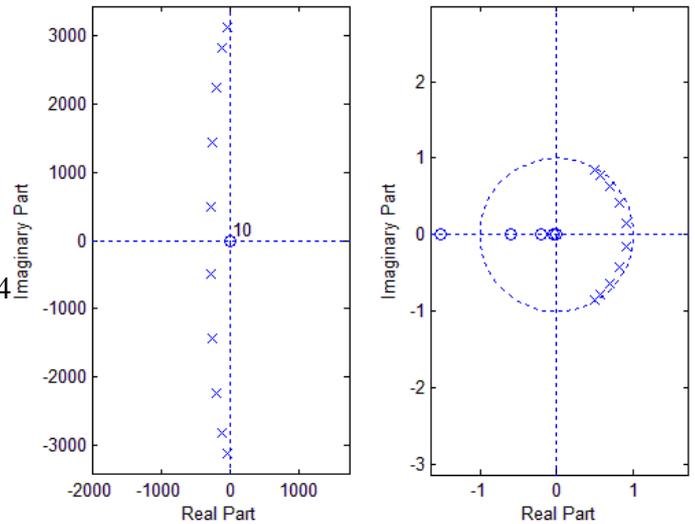
$$N \geq \frac{\log\left(\frac{\sqrt{\left(10^{\frac{A_a}{20}} - 1\right)} / \sqrt{\left(10^{\frac{A_p}{20}} - 1\right)} + \sqrt{\left(10^{\frac{A_a}{20}} - 1\right)} / \sqrt{\left(10^{\frac{A_p}{20}} - 1\right)} - 1\right)}{\log\left(\omega_a / \omega_p + \sqrt{\left(\omega_a / \omega_p\right)^2 - 1}\right)}$$

$$N \geq \frac{\log\left(2.10^{\frac{A_a}{20}} / \sqrt{\left(10^{\frac{A_p}{20}} - 1\right)}\right)}{\log\left(\omega_a / \omega_p + \sqrt{\left(\omega_a / \omega_p\right)^2 - 1}\right)}$$

Exemple

On souhaite réaliser un filtre passe-bas de gain unitaire avec un filtre de Tchebychev tel que :  $A_p=1\text{db}$ ,  $A_a=40\text{db}$ ,  $f_p=1\text{kHz}$  et  $f_a=3\text{kHz}$ . Déterminer N.

$$N \geq \frac{\log\left(2.10^{\frac{A_a}{20}} / \sqrt{\left(10^{\frac{A_p}{10}} - 1\right)}\right)}{\log\left(\omega_a / \omega_p + \sqrt{(\omega_a / \omega_p)^2 - 1}\right)} = \frac{2.100/0.5089}{\sqrt{3+\sqrt{8}}} = 3.39 \approx 4$$



Dans la pratique, on utilise trois valeurs d'ondulation,  $\epsilon = 0.1 \text{ dB}$ ,  $0.5 \text{ dB}$  et  $1 \text{ dB}$ . Pour ces 3 valeurs  $H_N(p)$  est donnée :

Ordre du filtre	Dénominateur (le numérateur est à 1)	Ordre du filtre	Dénominateur (le numérateur est à 1)
2	$0.3017 p^2 + 0.7158 p + 1$	2	$0.6595 p^2 + 0.9402 p + 1$
3	$(0.5918 p^2 + 0.5736 p + 1)(1.031 p + 1)$	3	$(0.8753 p^2 + 0.5483 p + 1)(1.596 p + 1)$
4	$(0.7518 p^2 + 0.3972 p + 1)(1.6053 p^2 + 2.0475 p + 1)$	4	$(0.9402 p^2 + 0.3297 p + 1)(2.8057 p^2 + 2.3755 p + 1)$
5	$(0.8368 p^2 + 0.2787 p + 1)(1.5725 p^2 + 1.3712 p + 1)(1.855 p + 1)$	5	$(2.0974 p^2 + 1.2296 p + 1)(0.9654 p^2 + 0.2161 p + 1)(2.759 p + 1)$

Ordre du filtre	Dénominateur (le numérateur est à 1)
2	$0.9070 p^2 + 0.9956 p + 1$
3	$(1.0058 p^2 + 0.497 p + 1)(2.023 p + 1)$
4	$(1.0136 p^2 + 0.2828 p + 1)(3.5791 p^2 + 2.4113 p + 1)$
5	$(2.3293 p^2 + 1.0911 p + 1)(1.0118 p^2 + 0.1610 p + 1)(3.454 p + 1)$

On rappelle que les fonctions de transfert  $H_N(p)$  des filtres analogiques polynômiaux (Butterworth, Tchebychev, Bessel, Cauer, etc.) sont données pour des fréquences de coupure normalisées et uniquement pour des filtres passe-bas [2].

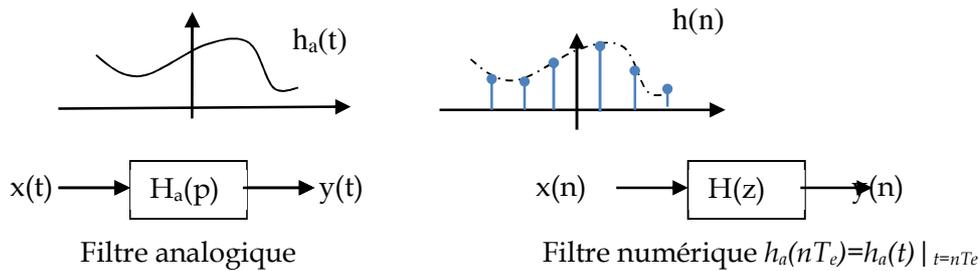
Passe-bas	$p = p / \omega_A$	Passe-bande	$p = \frac{1}{B} \left( \frac{p}{\omega_A} + \frac{\omega_A}{p} \right)$	Pulsation centrale $\omega_A = \sqrt{\omega_{A1} \omega_{A2}}$
Passe-haut	$p = \omega_A / p$	Coupe-bande	$p = \left[ \frac{1}{B} \left( \frac{p}{\omega_A} + \frac{\omega_A}{p} \right) \right]^{-1}$	Largeur de Bande $B = (\omega_{A2} - \omega_{A1})$

A partir de l'expression de  $H_N(p)$  normalisée, on dénormalise en remplaçant  $p$  par les valeurs données au tableau précédent permettant d'aboutir à une fonction de transfert dénormalisée  $H(p)$ .

### 3. Méthode de l'invariance de la réponse impulsionnelle

La méthode de l'invariance impulsionnelle consiste à effectuer la synthèse d'un filtre numérique dont la réponse impulsionnelle  $h(n)$  est l'échantillonnage de la réponse impulsionnelle  $h(t)$  du filtre analogique équivalent [18] :

- on détermine la réponse impulsionnelle désirée  $h_a(t)$  d'un filtre analogique connu.
- on échantillonne cette réponse impulsionnelle à la fréquence  $f_e$  et on en déduit la suite  $h(n)$ .
- on recherche la fonction de transfert  $H(z)$  du filtre numérique qui a pour réponse impulsionnelle la suite  $h(n)$ .



Rappelons toutefois que l'échantillonnage temporel se traduit par une périodisation en fréquence telle que :

$$X_e(f) = f_e \sum_n X(f - nf_e) \text{ Pour compenser la multiplication par } f_e, \text{ on prendre } h(n) = T_e \cdot h_a(nT_e)$$

Pour obtenir  $h(n)$ , on peut opérer de 2 manières:

$$H_a(p) \xrightarrow{L^{-1}} h_a(t) \xrightarrow{t=nT_e} h(n) = T_e h_a(nT_e) \xrightarrow{TZ} H(z) \text{ ou } H(z) = T_e \sum_{\text{pôles } p_i \text{ de } H_a(p)} \text{Résidus} \left( \frac{H_a(p)}{1 - z^{-1} e^{pT_e}} \right) \Bigg|_{p=p_i}$$

#### Exemple 1

On veut synthétiser un filtre numérique qui possède la même réponse impulsionnelle qu'un filtre passe-bas analogique du 1er ordre de transmittance normalisée:  $H_N(p) = \frac{1}{(1+p)}$

$$H_N(p) = \frac{1}{(1+p)}$$

On commence par dénormaliser  $H_a(p) = H_N(p) \Big|_{p=p/\omega_c} = \frac{1}{1+(p/\omega_c)} = \frac{\omega_c}{\omega_c + p}$

La réponse impulsionnelle est alors :  $h_a(t) = \omega_c e^{-\omega_c t}$  pour  $t \geq 0$

$$\Rightarrow h(n) = T_e \omega_c e^{-nT_e \omega_c} \text{ pour } n \geq 0 \Rightarrow H(z) = T_e \omega_c \frac{1}{1 - e^{-T_e \omega_c} z^{-1}} = T_e \omega_c \frac{z}{z - e^{-T_e \omega_c}}$$

Il s'avère qu'il y a égalité entre les 2 réponses à un coefficient K près, on posera donc :  $H(z) = K T_e \omega_c \frac{1}{1 - e^{-T_e \omega_c} z^{-1}}$

Sachant que  $p = 2\pi j f$ , pour  $f=0$ ,  $H_a(0)=1$  et  $z = e^{2\pi j f T_e}$ , pour  $f=0$ ,  $H(0) = K T_e \omega_c / (1 - e^{-T_e \omega_c}) \Rightarrow H(z) = \frac{1 - e^{-T_e/\tau}}{1 - e^{-T_e/\tau} z^{-1}}$

Remarque : La réponse du filtre numérique sera proche de celle du filtre analogique dans la bande  $[-f_e/2, f_e/2]$  si le filtre analogique a une réponse fréquentielle nulle en dehors de cette bande. Cette méthode est utile seulement dans le cas de filtres analogiques à bande limitée.

En effet, cette méthode n'est applicable que si l'on est en mesure de définir une fréquence d'échantillonnage adéquate [19]. Pour cela, il faut que la réponse en fréquence du filtre continu soit nulle (ou presque) au-delà d'une certaine fréquence de valeur finie. Cette méthode ne peut donc s'appliquer aux filtres de types passe haut ou coupe-bande ( $f_{max}$  est  $\infty$  donc on ne peut définir  $f_e > 2f_{max}$ ).

Exemple 2

On considère le filtre analogique d'ordre 3 associé à la fonction d'approximation de Butterworth. Réaliser le filtre numérique passe-bas correspondant en utilisant la méthode de l'invariance impulsionnelle. La fréquence de coupure est  $f_c = 1$  kHz et la fréquence d'échantillonnage  $f_e = 10$  kHz.

Rappelons que le filtre de Butterworth d'ordre 3 est formé de 2 cellules d'ordre 1 et 2, respectivement :

$$H_N(p) = \frac{1}{(1+p)(1+p+p^2)} \text{ alors } H(p) = \frac{1}{(1+p/\omega_c)(1+p/\omega_c+(p/\omega_c)^2)}$$

Pour éviter de décomposer en éléments simple, on peut calculer H(z) en utilisant les relations entre transformées de Laplace et en z pour les systèmes du premier et du second ordres.

$H(p)$	$T_e.H(z)$
$\frac{1}{p+a}$	$T_e \frac{1}{1 - e^{-aT_e} z^{-1}}$
$\frac{p+a}{(p+a)^2 + \omega_0^2}$	$T_e \frac{1 - e^{-aT_e} z^{-1} \cos(\omega_0 T_e)}{1 - 2e^{-aT_e} z^{-1} \cos(\omega_0 T_e) + e^{-2aT_e} z^{-2}}$
$\frac{\omega_0}{(p+a)^2 + \omega_0^2}$	$T_e \frac{e^{-aT_e} z^{-1} \sin(\omega_0 T_e)}{1 - 2e^{-aT_e} z^{-1} \cos(\omega_0 T_e) + e^{-2aT_e} z^{-2}}$

A cette fin, on décompose alors  $H(p)$  sous la forme :

$$H(p) = \frac{A}{(\omega_c + p)} + \frac{Bp + C}{(\omega_c^2 + \omega_c p + p^2)}$$

Par identification, on trouve  $C=0, A=-B=\omega_c \Rightarrow H(p) = \frac{\omega_c}{(\omega_c + p)} - \frac{\omega_c p}{(\omega_c^2 + \omega_c p + p^2)}$

Les pôles de ce filtre analogique sont  $p_1 = -\omega_c = -2000\pi$   $p_{2,3} = -\omega_c \frac{1 \pm \sqrt{3}j}{2} = -1000\pi(1 \pm \sqrt{3}j) = -a \pm j\omega_0$

$$H(p) = \frac{\omega_c}{(\omega_c + p)} - \frac{\omega_c p}{\left(p + \frac{\omega_c}{2}\right)^2 + 3\frac{\omega_c^2}{4}} = \frac{\omega_c}{(\omega_c + p)} - \omega_c \frac{p + \frac{\omega_c}{2} - \frac{\omega_c}{2}}{\left(p + \frac{\omega_c}{2}\right)^2 + 3\frac{\omega_c^2}{4}}$$

$$H(p) = \frac{\omega_c}{(\omega_c + p)} - \omega_c \frac{p + \frac{\omega_c}{2}}{\left(p + \frac{\omega_c}{2}\right)^2 + 3\frac{\omega_c^2}{4}} + \frac{\omega_c}{\sqrt{3}} \frac{\frac{\sqrt{3}\omega_c}{2}}{\left(p + \frac{\omega_c}{2}\right)^2 + 3\frac{\omega_c^2}{4}}$$

En se servant du tableau donné, on peut alors déterminer H(z) :

$$H(z) = T_e \omega_c \frac{1}{1 - e^{-\alpha T_e} z^{-1}} - T_e \omega_c \frac{1 - e^{-\pi/10} \cos(\pi\sqrt{3}/10)z^{-1}}{1 - 2e^{-\pi/10} \cos(\pi\sqrt{3}/10)z^{-1} + e^{-\pi/5} z^{-2}} + T_e \omega_c \frac{e^{-\pi/10} \sin(\pi\sqrt{3}/10)}{1 - 2e^{-\pi/10} \cos(\pi\sqrt{3}/10)z^{-1} + e^{-\pi/5} z^{-2}}$$

Ce qui nous donne :

$$H(z) = \pi/10 \frac{1}{1 - 0.5335z^{-1}} - \pi/10 \frac{(1 - 0.625z^{-1})}{1 - 1.25z^{-1} + 0.5335z^{-2}} + \pi/10 \frac{0.218}{1 - 1.25z^{-1} + 0.5335z^{-2}}$$

$$H(z) = \pi/10 \left( \frac{1}{1 - 0.5335z^{-1}} - \frac{0.781 - 0.625z^{-1}}{1 - 1.25z^{-1} + 0.5335z^{-2}} \right)$$

Il reste à déterminer le gain K en adaptant les gains des deux filtres analogique et numérique pour une fréquence donnée. Comme il s'agit d'un filtre passe-bas, on supposera l'égalité pour  $f=0$  ( $p=0$  et  $z=1$ )

#### 4. Méthode de la transformation bilinéaire

Cette méthode a pour objectif de faire coïncider au mieux les domaines analogique et numérique. Les filtres qui en sont dérivés sont plus stables que sont obtenus à travers l'emploi de la méthode de la variance impulsionnelle. Mais, en contrepartie, elle introduit une distorsion sur l'axe des fréquences.

On sait qu'un filtre analogique est caractérisé par sa fonction de transfert H(p) et un filtre numérique est défini par sa fonction de transfert H(z) avec  $z = e^{2\pi j f T_e} = e^{p T_e}$ . Ce qui implique que  $p = \ln(z)/T_e$ . Dans les deux cas, cette transformation entraîne une relation non linéaire entre les fréquences  $f_A$  du domaine analogique et les fréquences  $f_N$  du domaine numérique.

Pour conserver le caractère polynomial des fonctions de transfert, une approximation de  $\ln(z)$  par les séries de Laurent est employée :

$$\ln(z) \approx 2 \left( \frac{1-z^{-1}}{1+z^{-1}} + \frac{1}{3} \left( \frac{1-z^{-1}}{1+z^{-1}} \right)^3 + \frac{1}{5} \left( \frac{1-z^{-1}}{1+z^{-1}} \right)^5 + \dots \right)$$

En ne conservant que le terme du premier ordre, on définit la transformation bilinéaire :

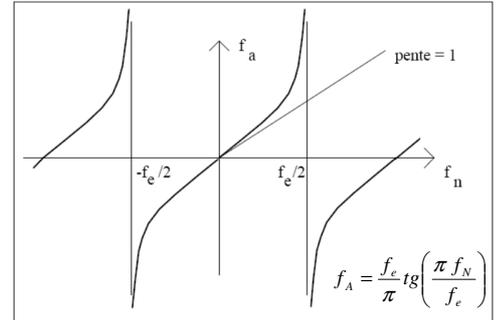
$$p = \frac{\ln(z)}{T_e} \approx \frac{2}{T_e} \frac{1-z^{-1}}{1+z^{-1}} = \frac{2}{T_e} \frac{z-1}{z+1}$$

Sachant que 
$$p = j\omega_A = \frac{\ln(z)}{T_e} = \frac{2}{T_e} \frac{e^{j\omega_N T_e} - 1}{e^{j\omega_N T_e} + 1} = \frac{2}{T_e} \frac{e^{j\omega_N T_e/2} (e^{j\omega_N T_e/2} - e^{-j\omega_N T_e/2})}{e^{j\omega_N T_e/2} (e^{j\omega_N T_e/2} + e^{-j\omega_N T_e/2})} = \frac{2}{T_e} \frac{j \sin(\omega_N T_e / 2)}{\cos(\omega_N T_e / 2)}$$

Ainsi, on peut montrer qu'on obtient la correspondance entre fréquence analogique  $f_A$  (ou pulsation analogique  $\omega_A$ ) et la fréquence numérique  $f_N$  (ou  $\omega_N$ ) par :

$$\omega_A = \frac{2}{T_e} \operatorname{tg} \left( \frac{\omega_N T_e}{2} \right) = \frac{2}{T_e} \operatorname{tg} \left( \frac{\pi f_N}{f_e} \right)$$

Cette équation montre qu'il n'y a pas égalité entre pulsation analogique et pulsation discrète et que la relation les liant n'est pas non plus linéaire puisqu'il y a distorsion des fréquences, y compris du retard de groupe.



Pour faire la synthèse d'un filtre numérique par la transformation bilinéaire, on procède comme suit :

- On définit les caractéristiques souhaitées du filtre numérique (fréquence d'échantillonnage, de coupure, etc.)
- On calcule les pulsations analogiques  $\omega_A$  correspondant aux pulsations numériques  $\omega_A = \frac{2}{T_e} \operatorname{tg} \left( \frac{\pi f_N}{f_e} \right)$
- On détermine le gabarit du filtre analogique  $H_N(p)$  normalisée d'ordre  $n$  (Chebyshev, Butterworth, etc.) qui servira de modèle au filtre numérique et on écrit la fonction de transfert dénormalisée  $H(p)$  de ce filtre analogique (qu'il faut recalculer en fonction de  $\omega_A$ ).
- On applique la transformation bilinéaire à  $H(p)$  en remplaçant  $p = \frac{2}{T_e} \frac{1 - z^{-1}}{1 + z^{-1}}$ , ce qui donne la fonction  $H(z)$ .

Exemple 1

On désire concevoir un filtre passe-bas numérique de premier ordre à partir d'une fonction de transfert d'un filtre RC dans le domaine continu ( $H_N(p)=1/(1+p)$ ). La fréquence de coupure désirée est  $f_N=30$  Hz et la fréquence d'échantillonnage est  $f_e=150$  Hz.

On calcule d'abord  $\omega_A = \frac{2}{T_e} \operatorname{tg} \left( \frac{\pi}{5} \right) = \frac{2}{T_e} 0.7265$  puis  $H(p) = \frac{1}{(1+p)} \Big|_{p=p/\omega_A} = \frac{\frac{2}{T_e} 0.7265}{p + \frac{2}{T_e} 0.7265}$

$$H(z) = H(p) \Big|_{p=\frac{2}{T_e} \frac{z-1}{z+1}} = \frac{\frac{2}{T_e} 0.7265}{\frac{2}{T_e} \frac{z-1}{z+1} + \frac{2}{T_e} 0.7265} = \frac{0.7265(z+1)}{(z-1) + 0.7265(z+1)} = \frac{0.7265(z+1)}{1.7265z - 0.2735}$$

On peut remarquer que  $f_A = \frac{2}{T_e (2\pi)} \operatorname{tg} \left( \frac{\pi}{5} \right) = \frac{f_e}{\pi} 0.7265 = 34.69 \neq f_N = 30$

Exemple 2 : On désire réaliser un filtre numérique passe-bas du second ordre avec les caractéristiques suivantes [18]: fréquence de coupure  $f_N= 500$  Hz (gain de 1), fréquence d'échantillonnage  $f_e=5$  kHz,

$$\omega_A = \frac{2}{T_e} \operatorname{tg}\left(\frac{\omega_N T_e}{2}\right) = \frac{2}{T_e} \operatorname{tg}\left(\frac{\pi}{10}\right) = \frac{2}{T_e} 0.325$$

$$H_N(p) = \frac{0.1075}{p^2 + 0.1075p + 1} \Rightarrow H(p) = H_N(p = p/\omega_A) = \frac{0.1075\omega_A^2}{p^2 + 0.1075p\omega_A + \omega_A^2} = \frac{\frac{4}{T_e^2} 0.0114}{p^2 + \frac{2}{T_e} 0.035p + \frac{4}{T_e^2} 0.1056}$$

$$H(z) = H(p) \Big|_{p=\frac{2}{T_e} \frac{z-1}{z+1}} = \frac{\frac{4}{T_e^2} 0.0114}{\left(\frac{2}{T_e} \frac{z-1}{z+1}\right)^2 + \frac{2}{T_e} 0.035 \frac{2}{T_e} \frac{z-1}{z+1} + \frac{4}{T_e^2} 0.1056} = \frac{0.0114}{\left(\frac{z-1}{z+1}\right)^2 + 0.035 \frac{z-1}{z+1} + 0.1056}$$

$$H(z) = \frac{0.0114(z+1)^2}{(z-1)^2 + 0.035(z-1)(z+1) + 0.1056(z+1)^2} = \frac{0.0114(z+1)^2}{1.14z^2 - 1.7888z + 1.0706}$$

Pour un filtre numérique passe-bas ou passe-haut, on peut déterminer  $H(z)$  directement à partir de  $H_N(p)$  en utilisant la pulsation normalisée  $\Omega_c = \operatorname{tg}\left(\frac{\pi f_N}{f_e}\right)$  et remplacer  $p$  dans  $H_N(p)$  comme suit:

- pour un passe-bas par  $p = \frac{1}{\Omega_c} \frac{z-1}{z+1}$  soit  $H(z) = H_N(p) \Big|_{p=\frac{1}{\Omega_c} \frac{z-1}{z+1}}$

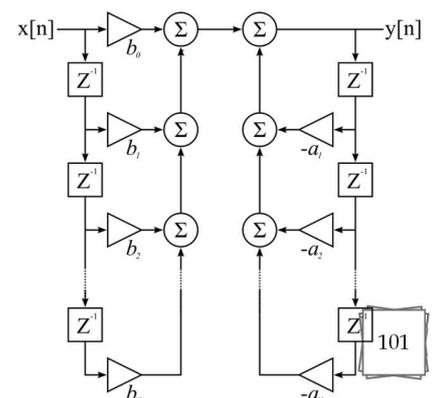
- pour un passe-haut par  $p = \Omega_c \frac{z+1}{z-1}$  soit  $H(z) = H_N(p) \Big|_{p=\Omega_c \frac{z+1}{z-1}}$

- pour un passe bande ou un rejeteur de bande, la détermination de  $\omega_A$  s'obtient par la pulsation centrale  $\omega_A = \sqrt{\omega_{A1}\omega_{A2}}$

Remarque: Puisqu'il n'est pas possible de faire coïncider précisément la réponse fréquentielle du filtre analogique (sur  $[0, +\infty[$  et numérique (sur  $[0, f_e/2]$ ) la transformation bilinéaire aura pour objet de faire coïncider au mieux les deux réponses. Cette méthode donne d'assez bons résultats à condition que les fréquences caractéristiques ne soient pas trop proches de la demi-fréquence d'échantillonnage.

### 5. Réalisation du filtre RII

Un filtre RII nécessite  $(2N+1)$  opérations de multiplication,  $2N$  opérations d'addition pour chaque nouvel échantillon à filtrer ou  $(2N + 1)$  MAC (Multiplication-Accumulation). Le coût mémoire d'un filtre RII en structure directe est de  $(4N+3)$  mémoire  $[(2N+1)$  coefficients  $b_i$  et  $2(N+1)$  points mémoire ou retard pour les vecteurs des entrées  $x(n)$  et des sorties  $y(n)$  [9].



Série 4

**Exercice 1 :** Déterminer, grâce à la position des pôles et des zéros, la fonction de transfert, les coefficients du filtre et l'équation de récurrence d'un filtre numérique passe-bande qui a les caractéristiques suivantes :

- Fréquence centrale du passe-bande : 125 Hz
- Largeur de bande à 3 dB : ± 10Hz
- Fréquence d'échantillonnage : 500Hz

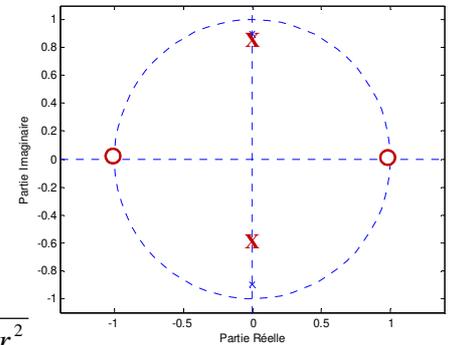
Solution

- Passe bande ⇒ Placer un zéro en 1 et un zéro en -1
- Calculer  $\theta = (125/500)*360^\circ=90^\circ$  et  $r=1-(10/500)x\pi \approx 0.937$

- Remplacer  $\theta$  et  $r$  dans l'équation  $H(z) = \frac{K(z-1)(z+1)}{(z-re^{-j\theta})(z-re^{j\theta})} = \frac{K(z^2-1)}{z^2-2zr\cos\theta+r^2}$

$$H(z) = \frac{K(z^2-1)}{z^2+0.878} = \frac{K(1-z^{-2})}{1+0.878z^{-2}} \text{ avec } K = \frac{|e^{2j\theta} - 2r\cos\theta e^{j\theta} + r^2|}{|e^{2j\theta} - 1|}$$

- $b_0=K \quad b_1=0 \quad b_2=-K \quad a_0=1 \quad a_1=0 \quad a_2=0.878$
- L'équation de récurrence est :  $y(n) = K.x(n) - K.x(n-2) - 0.878y(n-2)$



**Exercice 2 :** On souhaite approcher un filtre idéal passe-haut de second ordre par un filtre à réponse impulsionnelle infinie, synthétisé par la méthode des pôles et des zéros. Ce filtre doit répondre aux spécifications suivantes

- Fréquence de coupure  $f_c=2$  kHz
  - Fréquence d'échantillonnage :  $f_e=8$  kHz
  - Tracer  $H(f)$  idéal
  - Donner l'expression théorique de  $H(z)$
  - Placer les pôles et les zéros pour obtenir une approximation de  $H(f)$
  - Déterminer l'expression mathématique exacte de  $H(z)$  en déterminant  $K$
  - Déterminer l'équation aux différences.
- Largeur de transition à 3db:  $\Delta f=240$  Hz

$$\theta = (f_c / f_e) \times 360^\circ = 2/8 * 360 = 90^\circ \quad H(z) = \frac{K(z-1)(z-1)}{(z-re^{-j\theta})(z-re^{j\theta})} = \frac{K(z^2-2z+1)}{z^2-2zr\cos\theta+r^2} = \frac{K(z^2-2z+1)}{z^2+0.82}$$

$$r = 1 - \frac{\Delta f_{3db}}{f_e} \pi = 0.906 \quad \text{avec } K = \frac{1+r^2+2r\cos\theta}{4} = 0.455$$

**Exercice 3 :** Considérer la fonction de transfert passe bas dénormalisée  $H(p)$  suivant :  $2.25/(p^2+0.3p+2.25)$ . Utiliser la méthode de l'invariance impulsionnelle puis celle des pôles et zéros pour transformer ce filtre en numérique.

Solution

$$H(p) = \frac{2.25}{p^2+0.3p+2.25} = 1.5075 \frac{1.4925}{(p+0.15)^2+1.4925} \xrightarrow{\text{invariance impulsionnelle}} H(z) = \frac{1.292z}{z^2-0.1347z+0.7408}$$

$\frac{\omega_0}{(p+a)^2 + \omega_0^2}$	$T_e \frac{e^{-aT_e} z^{-1} \sin(\omega_0 T_e)}{1 - 2e^{-aT_e} z^{-1} \cos(\omega_0 T_e) + e^{-2aT_e} z^{-2}}$
---	--

Remarque On peut renforcer le caractère passe-bas en rajoutant un zéro en -1 soit  $H(z) = \frac{1.292z(z+1)}{z^2-0.1347z+0.7408}$

$p_{1,2} = -0.15 \pm j1.4925j \Rightarrow a = 0.15, \omega_0 = 1.4925 \Rightarrow$  pôles transposés  $z_{1,2} = 0.0671 \pm j0.8574$

$$H(p) = \frac{2.25}{p^2 + 0.3p + 2.25} \xrightarrow{\text{pôles et zéros}} H(z) = \frac{K(z+1)^2}{z^2 - 0.1347z + 0.7408}$$

**Exercice 4 :** Considérer la fonction de transfert passe bas  $H(p)$  dénormalisée suivante :  $6/(p^2+5p+6)$ . Utiliser la méthode de l'invariance impulsionnelle pour transformer ce filtre en numérique.

Solution  $H_A(p) = \frac{6}{p+2} \frac{1}{p+3} = \frac{6}{p+2} + \frac{-6}{p+3} \Rightarrow h_a(t) = 6e^{-2t}U(t) - 6e^{-3t}U(t)$

$$\Rightarrow h(n) = 6T_e e^{-2nT_e} U(n) - 6T_e e^{-3nT_e} U(n) \Rightarrow H(z) = \frac{6T_e}{1 - e^{2T_e} z^{-1}} + \frac{-6T_e}{1 - e^{3T_e} z^{-1}}$$

**Exercice 5 :** On considère le filtre passe-bas analogique normalisé à un pôle unique  $H_N(p) = 1/(p+1)$ . On considère que la fréquence de coupure normalisée à -3db se produit pour  $f_c = f_e/10$ . Par transformation bilinéaire, trouver le filtre numérique passe-bas équivalent  $H(z)$ .

$$\omega_A = \frac{2}{T_e} \operatorname{tg}\left(\frac{\pi}{10}\right) = \frac{0.65}{T_e} \Rightarrow H(p) = H_N(p) \Big|_{p/\omega_A} = \frac{\omega_A}{p + \omega_A} = \frac{0.65/T_e}{p + 0.65/T_e}$$

$$p = \frac{2}{T_e} \frac{1-z^{-1}}{1+z^{-1}} \Rightarrow H(z) = H(p) \Big|_{p=\frac{2}{T_e} \frac{1-z^{-1}}{1+z^{-1}}} = \frac{0.65/T_e}{\frac{2}{T_e} \frac{1-z^{-1}}{1+z^{-1}} + 0.65/T_e} = \frac{0.245(1+z^{-1})}{1-0.509z^{-1}}$$

**Exercice 6:** Soit la fonction de transfert passe bas dénormalisée  $H(p)$  suivante :  $(p+0.1)/((p+0.1)^2+\omega_a^2)$ . Sachant que  $\omega_a=4$ , utiliser la méthode de la transformée bilinéaire pour transformer ce filtre en numérique. Le filtre numérique aura sa résonance pour  $\omega_N = \pi/(2.T_e)$ .

$$\omega_a = 4 = \frac{2}{T_e} \operatorname{tg}\left(\frac{\pi}{4}\right) \Rightarrow T_e = 0.5 \Rightarrow p = 4 \frac{1-z^{-1}}{1+z^{-1}} \Rightarrow H(z) = \frac{4 \frac{1-z^{-1}}{1+z^{-1}} + 0.1}{\left(4 \frac{1-z^{-1}}{1+z^{-1}} + 0.1\right)^2 + 16} = \frac{0.125 + 0.006z^{-1} - 0.118z^{-2}}{1 + 0.0006z^{-1} - 0.950z^{-2}}$$

**Exercices supplémentaires**

1. Au cours de la transmission d'un signal numérique (échantillonné à une fréquence de 5 kHz), il a été affecté d'un bruit sinusoïdal de fréquence  $f_0=250$  Hz. On veut éliminer le bruit par l'emploi d'un filtre possédant une bande de transition  $\Delta f = \pm 50$  Hz à -3db.

Concevoir un filtre RII de second ordre par placement des pôles et zéros

- Tracer  $H(f)$  idéal
- Déterminer  $\theta$  et  $R$  puis donner le racé des pôles et zéros
- Déterminer  $H(z)$  et le gain  $K$ .
- Donner l'équation aux récurrences
- Citer un inconvénient des filtres RII de 1er ordre et de 2ème ordre

2. On considère le filtre analogique dénormalisé  $H_a(p) = (p+0.1)/((p+0.1)^2+9)$ . Par la méthode de l'invariance de la réponse impulsionnelle, trouver le filtre numérique passe-bas équivalent  $H(z)$ .

**Solution** 3 pôles  $p_1 = -0.1$ ,  $p_{2,3} = -0.1 \pm 3j$   $H(z) = T_e \sum_{\text{pôles } p_i \text{ de } H_A(p)} \text{Résidus} \left( \frac{H_A(p)}{1 - z^{-1} e^{pT_e}} \right) \Big|_{p=p_i}$

$$H_A(p) = \frac{0.5}{p + 0.1 - 3j} + \frac{0.5}{p + 0.1 + 3j}$$

$$H(z) = \frac{0.5T_e}{1 - e^{0.1T_e} e^{3jT_e} z^{-1}} + \frac{0.5T_e}{1 - e^{-0.1T_e} e^{3jT_e} z^{-1}} = T_e \frac{1 - e^{-0.1T_e} \cos(3T_e) z^{-1}}{1 - 2e^{-0.1T_e} \cos(3T_e) z^{-1} + e^{-0.2T_e} z^{-2}}$$

3. On considère le filtre analogique dénormalisé  $H(p) = 1/[(1+p)(p+2)^2]$ .

- Déterminer la réponse impulsionnelle

- Par la méthode de l'invariance de la réponse impulsionnelle, trouver le filtre numérique équivalent  $h(n)$ .

$$H(p) = \frac{1}{p+1} + \frac{-1}{(p+2)^2} + \frac{-1}{(p+2)} \Rightarrow h_a(t) = e^{-t}U(t) - te^{-2t}U(t) - e^{-2t}U(t)$$

$$h(n) = T_e h_a(nT_e) = T_e e^{-nT_e}U(n) - nT_e^2 e^{-2nT_e}U(n) - T_e e^{-2nT_e}U(n)$$

4. Soit à construire un filtre numérique passe-bas échantillonné à la fréquence  $f_c = 4\text{kHz}$  par la méthode de l'invariance impulsionnelle. La fonction modèle est la réponse en fréquence d'un filtre passe-bas de type Butterworth du 2ème ordre dont la fréquence de coupure à -3dB est égale à 500Hz.

$$H_N(p) = \frac{1}{p^2 + \sqrt{2}p + 1} \Rightarrow H(p) = H_N(p) \Big|_{p=p/\omega_c} = \frac{\omega_c^2}{p^2 + \sqrt{2}\omega_c p + \omega_c^2}$$

$$H(p) = \frac{\omega_c^2}{p^2 + \sqrt{2}\omega_c p + \omega_c^2} = \frac{\omega_c^2}{\left(p + \frac{\sqrt{2}}{2}\omega_c\right)^2 + \frac{\omega_c^2}{2}} = \frac{\omega_c^2}{\left(p + \frac{1}{\sqrt{2}}\omega_c\right)^2 + \left(\frac{\omega_c}{\sqrt{2}}\right)^2} = \sqrt{2}\omega_c \frac{\omega_c/\sqrt{2}}{\left(p + \frac{1}{\sqrt{2}}\omega_c\right)^2 + \left(\frac{\omega_c}{\sqrt{2}}\right)^2}$$

$$H(z) = T_e \frac{\sqrt{2}\omega_c z^{-1} \sin(\omega_c T_e / \sqrt{2}) e^{-\omega_c T_e / \sqrt{2}}}{1 - 2z^{-1} \cos(\omega_c T_e / \sqrt{2}) e^{-\omega_c T_e / \sqrt{2}} + z^{-2} e^{-2\omega_c T_e / \sqrt{2}}} \quad \omega_c T_e = \pi/4 \Rightarrow \omega_c T_e / \sqrt{2} = 0.555$$

$$H(z) = T_e \frac{\sqrt{2}\omega_c z^{-1} \sin(\omega_c T_e / \sqrt{2}) e^{-\omega_c T_e / \sqrt{2}}}{1 - 2z^{-1} \cos(\omega_c T_e / \sqrt{2}) e^{-\omega_c T_e / \sqrt{2}} + z^{-2} e^{-2\omega_c T_e / \sqrt{2}}} = \frac{0.336z^{-1}}{1 - 0.975z^{-1} + 0.328z^{-2}}$$

5. On considère le filtre analogique d'ordre 3 associé à la fonction d'approximation de Butterworth. Réaliser le filtre numérique passe-bas correspondant en utilisant la transformation bilinéaire. La fréquence de coupure est  $f_c = 1\text{ kHz}$  et la fréquence d'échantillonnage  $f_e = 10\text{ kHz}$ .

$$H_N(p) = \frac{1}{(1+p)(1+p+p^2)} \Rightarrow H(p) = H_N(p) \Big|_{p=p/\omega_c} = \frac{1}{(1+p/\omega_c)(1+p/\omega_c+(p/\omega_c)^2)}$$

$$\omega_c = \frac{2}{T_e} \text{tg}\left(\frac{\pi}{10}\right) = \frac{2 * 0.325}{T_e} \quad H(z) = H(p) \Big|_{p=\frac{2}{T_e} \frac{z-1}{z+1}}$$

$$H(z) = \frac{0.034(z+1)^3}{(1.325z - 0.675)(1.43z^2 - 1.79z + 0.78)}$$

Remarque :  $H(z) = H_N(p) \Big|_{p=\frac{1}{\Omega_c} \frac{z-1}{z+1}} = H_N(p) \Big|_{p=\frac{1}{0.325} \frac{z-1}{z+1}}$

6. En employant un filtre de Butterworth passe-bas de second ordre, concevoir un filtre numérique correspondant (passe-bas) en utilisant la transformation bilinéaire. La fréquence de coupure est  $f_c=100$  kHz et la fréquence d'échantillonnage  $f_e=1$  kHz.

$$\omega_a = \frac{2}{T_e} 0.325 \text{ rad/s} \quad H_N(p) = \frac{1}{p^2 + \sqrt{2} p + 1} \Rightarrow H(z) = H_N(p) \Big|_{p=p/\omega_c} = \frac{\omega_c^2}{p^2 + \sqrt{2} \omega_c p + \omega_c^2}$$

$$H(z) = H(p) \Big|_{p=\frac{z-1}{T_e z+1}} = \frac{0.1056(z+1)^2}{1.65z^2 - 1.8z + 0.65}$$

7. On cherche à réaliser un filtre numérique équivalent au filtre analogique dénormalisé de transmittance :

$$H_A(p) = \frac{1}{1 + 0.2p}$$

- Déterminer la réponse impulsionnelle  $h_a(t)$
- Calculer la réponse en  $z$  de ce filtre obtenue par transformation bilinéaire pour une fréquence d'échantillonnage  $T_e = 0.2$ .
- Quelle est sa fréquence de coupure ?
- Calculer la réponse en  $z$  d'un filtre similaire de même fréquence de coupure que le filtre analogique.

8. En employant un filtre de Butterworth passe-bas de second ordre, concevoir un filtre numérique correspondant (passe-bas) en utilisant la transformation bilinéaire. La fréquence de coupure est  $f_c=200$  Hz et la fréquence d'échantillonnage  $f_e=1$  kHz.

$$H_N(p) = \frac{1}{p^2 + \sqrt{2} p + 1}$$

- Déterminer  $H(z)$
- Etablir le tracé des pôles et zéros
- Comparer les fréquences analogiques et numériques pour  $f_c=200$  Hz et  $f_c=400$  Hz
- Si  $f_c=400$  Hz, quel problème cela posera-t-il?
- Quelle méthode alors utiliser? Quel est son inconvénient?

$$\omega_a = \frac{2}{T_e} 0.7265 \text{ rad/s} \quad H_N(p) = \frac{1}{p^2 + \sqrt{2} p + 1} \Rightarrow H(z) = H_N(p) \Big|_{p=p/\omega_c} = \frac{\omega_c^2}{p^2 + \sqrt{2} \omega_c p + \omega_c^2}$$

9. On désire choisir, dans chacun des cas suivants, la meilleure approche de synthèse de filtres numériques RII

- Filtre passe-haut ( $f_c=200$ Hz,  $f_e=1$ kHz)
- Filtre rejeteur ( $f_c=400$ Hz,  $f_e=1$ kHz)
- Filtre passe-bas ( $f_c=450$ Hz,  $f_e=1$ kHz)

10. En employant un filtre de Butterworth passe-bas de second ordre, concevoir un filtre numérique correspondant **passe-haut** en utilisant la transformation bilinéaire. La fréquence de coupure désirée est  $f_c=250$  Hz et la fréquence d'échantillonnage  $f_e$  vaut **1kHz**.

$$H_N(p) = \frac{1}{p^2 + \sqrt{2} p + 1}$$

1. Déterminer  $H(z)$
2. Calculer la fréquence de coupure analogique et justifier la différence avec celle numérique.

## TP n°4: Conception des filtres RII (Sous Matlab)

**But du TP :** Dans ce TP, on synthétisera un filtre RII par la méthode des pôles et zéros puis par la méthode de l'invariance impulsionnelle et enfin par transformation bilinéaire en utilisant des filtres analogiques (Chebychev et Butterworth).

### Rappels sur les filtres RII

L'intérêt des filtres récurrents (RII) est leur faible coût en calcul. Avec très peu de pôles et zéros on peut assurer la plupart des réponses fréquentielles dont on peut avoir besoin dans les différentes applications. Les inconvénients des filtres récurrents sont : leur non-linéarité en phase linéaire (phase linéaire : temps de propagation constant pour toute fréquence), et leur instabilité numérique. En effet, le filtre étant rétroactif, les erreurs de précision numérique deviennent une question d'importance, car elles peuvent s'amplifier et devenir hors contrôle, d'abord dans la forme de bruit, mais éventuellement dans la forme d'instabilité. A noter que les filtres RII peuvent être conçus par des méthodes semblables à ceux utilisés pour les filtres analogiques.

### 1. Synthèse d'un filtre numérique par placement des pôles et zéros

Il s'agit de déterminer, grâce à la position des pôles et des zéros, la fonction de transfert, les coefficients d'un filtre et l'équation de récurrence d'un filtre numérique passe-bande de second ordre qui a les caractéristiques suivantes:

- Fréquence à rejeter : 125 Hz
- Largeur de bande à 3 dB :  $\pm 10$ Hz
- Fréquence d'échantillonnage : 500Hz

```
clearall; clc; close all;
fe=500; df=10; fc=125;
teta=360*fc/fe; tet=2*pi*fc/fe; R=1-df*pi/fe;
K=(exp(2*j*teta)-2*R*cos(teta)*exp(j*teta)+R*R)/(exp(2*j*teta)-1); K=abs(K);
a=[1 -2*R*cos(tet) R*R]; b=K*[1 0 -1];
dirac=[1; zeros(99,1)]; h=filter(b,a,dirac);
[H F] = freqz(b, a, 512, fe); [tau,f]=grpdelay(b,a,512,fe);
figure; subplot(2,2,1);hold on;plot(h,'red');
subplot(2,2,2);hold on;plot(F,abs(H),'red');
subplot(2,2,3);hold on;zplane(b,a);
subplot(2,2,4);hold on;plot(f,tau,'red');
```

1. Quelle est la nature de la bande passante créée
2. Modifier ce programme pour en faire un coupe-bande et commenter les graphes obtenus.
3. Reprendre ce programme pour filtrer le signal  $ecg(fe=500 \text{ Hz et bruit à } 50\text{Hz})$ .
  - Visualiser les TFD du signal avant et après filtrage.
  - Comparer avec la méthode de fenêtres (TP n°3), laquelle vous semble préférable (justifier)?

## 2. Synthèse d'un filtre numérique par transformation d'un filtre analogique

Pour déterminer les coefficients du filtre RII, il suffit de synthétiser chaque filtre en continu  $H(p)$  puis de passer à  $H(z)$  soit par transformation bilinéaire ou par invariance impulsionnelle

On veut, par exemple, synthétiser un filtre passe-bas avec les spécifications suivantes :  $f_e=3000\text{Hz}$ ;  $f_p=500\text{Hz}$  et une oscillation en bande passante  $\delta_1 < 3\text{ db}$  et une atténuation en bande atténuée  $\delta_2 > 40\text{ db}$ .

**I.** On commence par choisir un filtre analogique normalisé  $H_n(p)$  (en continu) d'ordre  $N$  puis on le dénormalise pour créer  $H(p)$  continu. Ensuite, on utilise une des méthodes de transformation permettant trouver les paramètres du filtre numérique

```

clc;clear all; close all;

Fe=3000;fp=500; att_p=3; att_a=40; N=10;
wp=fp*2*pi; %transformation des fréquences en pulsations

%Création d'un filtre analogique normalisé (wc=1) passe-bas Hn(p) d'ordre N
[z,p,k] = cheblap(N,att_p); %cheb2ap(N,att_a); %ellipap(N,att_p,att_a) %besselap(N); %buttap(N);

%Dénormalisation passage de Hn(p) vers H(p)
[Bpn,Apn] = zp2tf(z,p,k);
[Bp,Ap] = lp2lp(Bpn,Apn,wp); %lp2hp %lp2bp %lp2bs

% Détermination du filtre numérique
[Bn,An] =impinvar(Bp,Ap,Fe);
figure; subplot(121); zplane(1,Ap); subplot(122); zplane(1,An);

% Calcul de la réponse impulsionnelle analogique et numérique
[r,p,k]=residue(Bp,Ap); t=0:1/(5*Fe):.02; ha=exp(t*(p.))*r;
hn=filter(Bn,An,[1;zeros(49,1)]);
figure; subplot(1,2,1);plot(t,ha);hold on; stem(0:1/Fe:49/Fe, Fe*hn,'r.')
```

```

% Détermination de Ha(f) et H(f)
[Ha,W]=freqs(Bp,Ap,2*pi*(1:20:Fe/2)); [H,f]=freqz(Bn,An, 1:20:Fe/2,Fe);
subplot(1,2,2);plot(W/(2*pi),abs(Ha));hold on; stem(f,abs(H),'r.')
```

1. Que contiennent  $z$ ,  $p$  et  $k$ ,  $B_{pn}, A_{pn}, B_p, A_p$  ?
2. A partir du tracé des pôles, étudier la stabilité des 2 filtres.
3. Aidez vous du help pour expliquer les instructions  
`zp2tf(z,p,k);lp2lp(Bpn,Apn,wp); [r,p,k]=residue(Bp,Ap); t=0:1/(5*Fe):.02; ha=exp(t*(p.))*r; freqs(N,D,2*pi*(1:20:Fe/2)); plot(W/(2*pi),abs(Ha));Fe*hn`
4. Commenter les réponses impulsionnelles et fréquentielles
5. Quelles sont les caractéristiques du filtre de Chebychev (avantage et inconvénient)
6. Que se passe-t-il si on augmente  $N$ ?
7. Tester les autres filtres analogiques et commenter les différences (oscillations,  $\Delta f$ , etc.).
8. Que faut-il modifier pour en faire un passe-haut? Quel est l'inconvénient principal de l'approche par invariance impulsionnelle?

**II.** On veut refaire le même travail en employant la transformation bilinéaire

```
[Bn,An] = bilinear(Bp,Ap,Fe,fp);
```

9. Comparer les 2 méthodes lorsque  $f_p$  est proche de la demi-fréquence d'échantillonnage  $f_s/2$ .
10. Tester la fonction `cheblord`

## TP n°4: Conception des filtres RII (Sous Python)

**But du TP :** Dans ce TP, on synthétisera un filtre RII par la méthode des pôles et zéros puis par la méthode de l'invariance impulsionnelle et enfin par transformation bilinéaire en utilisant des filtres analogiques (Chebychev et Butterworth).

### Rappels sur les filtres RII

L'intérêt des filtres récurrents (RII) est leur faible coût en calcul. Avec très peu de pôles et zéros on peut assurer la plupart des réponses fréquentielles dont on peut avoir besoin dans différentes applications. Les inconvénients des filtres récurrents sont : leur non-linéarité en phase linéaire (phase linéaire : temps de propagation constant pour toute fréquence), et leur instabilité numérique. En effet, le filtre étant rétroactif, les erreurs de précision numérique deviennent une question d'importance, car elles peuvent s'amplifier et devenir hors contrôle, d'abord dans la forme de bruit, mais éventuellement dans la forme d'instabilité. A noter que les filtres RII peuvent être conçus par des méthodes semblables à ceux utilisés pour les filtres analogiques.

### 1. Démo

# -\*- coding: utf-8 -\*-

```
import numpy as np; import scipy.signal as sp; import matplotlib.pyplot as plt
from plot_zplane import zplane
```

```
# Placement des pôles et zéros
```

```
# fe=500; df=40; fc=50;
```

```
# tet=360*fc/fe; tet=2*np.pi*fc/fe;
```

```
# R=1-df*np.pi/fe;
```

```
# a = np.array([1, -2*R*np.cos(tet), R*R]);
```

```
# b = np.array([1, -2, 1]);
```

```
# #K = (1+R*R-2*R*np.cos(tet))/(2-2*np.cos(tet))
```

```
# K=(np.exp(2*1j*tet)-2*R*np.cos(tet)*np.exp(1j*tet)+R*R)/(np.exp(2*1j*tet)-1);
```

```
# K=np.abs(K); b=K*b;
```

```
# dirac = np.zeros(100);dirac[0] = 1; h=sp.lfilter(b,a,dirac);
```

```
# plt.figure(1);
```

```
# plt.subplot(211);plt.plot(h); plt.title('Réponse impulsionnelle'); plt.grid(True);plt.ylabel('Amplitude')
```

```
# L = 256;
```

```
# f,Hf= sp.freqz(b,a,L,fs=fe);
```

```
# plt.subplot(212); plt.plot(f, np.abs(Hf)); plt.title('Module du Filtre'); plt.grid(True); plt.xlabel('Fréq (Hz)');plt.ylabel('Amp')
```

```
# plt.figure(2);
```

```
# z,p=zplane(b,a);
```

```
# Transformation bilinéaire
```

```
fe=500; fp=100; att_p=1; att_a=40;
```

```
wp = fp*2*np.pi; #Num= Anal pour l'invariance impulsionnelle
```

```
wpp = 2*fe*np.tan(np.pi*fp/fe) #Compensation de pulsation
```

```

N, WN = sp.cheb1ord(2*np.pi*fp,2*np.pi*105, att_p,att_a, fs=fe)
z,p,k = sp.cheb1ap(N,att_p); """"Filtre analogique hn(p) sous forme de poles, zéros et gain""""
# # sp.cheb2ap(N,att_a); sp.buttap(N); sp.ellipap(N,att_p,att_a)
Bpn,Apn = sp.zpk2tf(z,p,k); """"Filtre analogique hn(p) sous forme num, den """"
Bp, Ap = sp.lp2lp (Bpn,Apn,wpp); """"Denormalisation """"
# # sp.lp2hp sp.lp2bp sp.lp2bs
poles_anal = np.roots(Ap);

Bz1, Az1 = sp.bilinear(Bp,Ap,fe); """"Passage de H(p) vers H(z) par Transformation Bilineaire""""

# Comparaison filres analogique Ha(f) et numérique Hz(f)
L = 256;
fz,HZ= sp.freqz(Bz1,Az1,L,fs=fe);
wa,Ha = sp.freqs(Bp,Ap, worN=np.arange(1,np.pi*fe)); fa=wa*0.5/ np.pi;
plt.figure(3);
plt.plot(fz, np.abs(Hz),label='H(f) du filtre Numérique');
plt.plot(fa, np.abs(Ha), label='H(f) du filtre Analogique');plt.legend()

plt.figure(4);
z,p=zplane(Bz1,Az1);

#invariance impulsionnelle
Bz2,Az2 = sp.iirfilter(N, fp, rp=att_p, btype='lowpass',analog=False, ftype='cheby1', output='ba', fs=fe)
Bp2,Ap2 = sp.iirfilter(N, 2*np.pi*fp, rp=att_p, btype='lowpass',analog=True, ftype='cheby1', output='ba')

# Comparaison filres analogique Ha(f) et numérique Hz(f)
L = 256;
fz,HZ= sp.freqz(Bz2,Az2,L,fs=fe);
wa,Ha = sp.freqs(Bp2,Ap2,worN=np.arange(1,np.pi*fe)); fa=wa*0.5/ np.pi;
plt.figure(5);
plt.plot(fz, np.abs(Hz),label='H(f) du filtre Numérique');
plt.plot(fa, np.abs(Ha), label='H(f) du filtre Analogique');plt.legend()

plt.figure(7);
z,p=zplane(Bz2,Az2);

```

## 2. A faire

```

# -*- coding: utf-8 -*-

import numpy as np; import matplotlib.pyplot as plt
#from zp_plot import zplane
import scipy.signal as sp;

from scipy.io import wavfile as wf; import winsound ;
fname = 'vousavezducourrierenattente.wav';
winsound.PlaySound(fname, winsound.SND_FILENAME)
fe, X = wf.read(fname);

```

```
Te=1/fe; N=len(X); t = np.arange(0,N)*Te;
plt.figure(1);plt.subplot(211); plt.plot(t,X); plt.grid(True);
plt.xlabel('temps'); plt.ylabel('Amp'); plt.title('Phrase');
# #Erire dans un fichier audio
# fnameb='Copie du signal.wav';
# Xb=X
# Xb=np.int8(Xb) ; wf.write(fnameb, fe, Xb);
# winsound.PlaySound(fnameb,winsound.SND_FILENAME) ;

#-----A faire-----
# Calculer et Visualiser sa TFD en db dans la figure (1)
# Rajouter du bruit ( 3 sinusoides avec une fréquence de 1500, 2000 et 3000)
# Visualiser la TFD du signal bruité en db
# Enregistrer le fichier avec wf.write dans fnameb
# Ecouter fnameb
# Concevoir Filtre par la méthode des pôles et zéros (employer autant de filtres que nécessaire ) puis
par la transformation bilinéaire avec  $Aa > 50$  et  $\Delta f < f_e / 10$ , avec N à déterminer, il faut essayer
plusieurs filtres analogiques, lequel vous semble adéquat?
# Visualiser la réponse impulsionnelle et la fonction de Transfert du filtre pour chaque méthode
# Filtre le signal bruité Xb (sp.lfilter) avec le filtre synthétisé
# Enregistrer, écouter et visualiser la TFD du signal filtré
# Commenter
# Comparer entre les méthodes des synthèse des filtres RIF. Pour cet exemple, lequel vous semble
préférable ?
```

### V. Filtres numériques multicaudences

Le traitement multi-cadences implique plusieurs fréquences d'échantillonnage. Il peut avoir plusieurs motivations. Il peut être le fruit de contraintes liées à une application dans laquelle les flux de signaux numériques à traiter et à générer doivent l'être à des fréquences d'échantillonnage différentes [20]. Cette situation peut se rencontrer par exemple dans les applications audio où co-existent plusieurs fréquences d'échantillonnage : 32kHz (diffusion); 44,1 kHz (disque compact numérique) ; 48kHz (bande sonore numérique) ; 96kHz ; ...

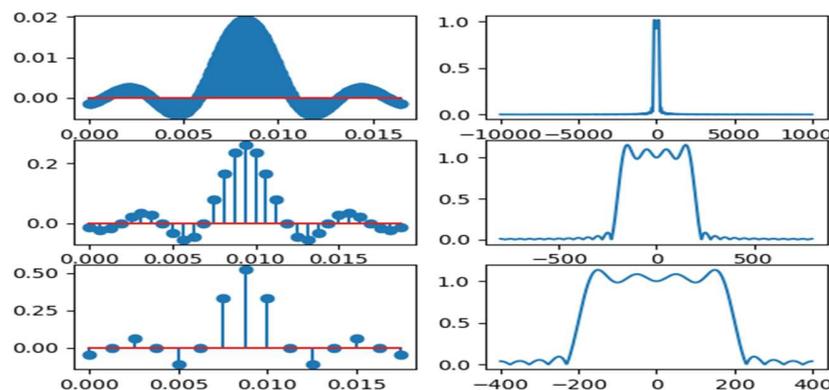
Cette approche peut aussi permettre d'améliorer les caractéristiques d'implantation d'un algorithme en adoptant pour chacune de ses étapes un échantillonnage que l'on appelle critique dans le sens où la fréquence de traitement est choisie égale à la fréquence de Nyquist. Ainsi la chaîne de traitement correspondant à l'émetteur bande de base d'un modulateur DPQSK où l'on a trois fréquences de traitement : la fréquence bit, la fréquence symbole et la fréquence d'échantillonnage [20]

Dans les systèmes considérés aux chapitres précédents, n'était considéré qu'une seule fréquence ou cadence d'échantillonnage  $f_e = 1/T_e$ . Mais il arrive que la fréquence maximale  $f_{max}$  soit diminuée (filtrage passe-bas) ou augmentée (le cas d'une modulation). Dans les deux cas, il est plus judicieux d'adapter la fréquence d'échantillonnage afin de minimiser le temps de calcul. Il s'agira d'une décimation ou sous-échantillonnage (augmentation de  $f_{max}$ ) et d'une interpolation ou sur-échantillonnage (diminution de  $f_{max}$ ). A noter, également que le sur-échantillonnage est souvent utilisé pour augmenter le SNR dans les opérations de CNA. Tous deux interviennent dans le changement de fréquence et dans les techniques des bancs de filtres

Exemple : Supposons que l'on veuille synthétiser un filtre passe-bas avec les spécifications suivantes :  $f_p=100\text{Hz}$   $f_a=300\text{Hz}$   $A_a>50\text{ db}$   $f_e=20\text{kHz}$ .

L'emploi de la fenêtre de Hamming nous donne  $N=331$ , soit un nombre de coefficients important. Or le filtre occupe une petite portion de la bande passante (passe-bas avec  $f_c=1\% f_e$ ). On peut donc sous-échantillonner et puis sur-échantillonner sans perte d'informations.

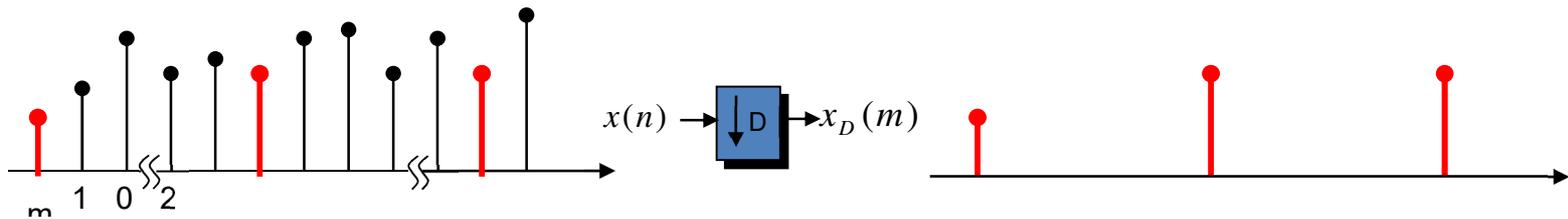
Ainsi en prenant  $f_e=f_e/25=800\text{Hz}$  cela nous donnerait  $N=15 \ll 331$  et on obtient, ainsi, un filtre plus rapide. C'est le principe des bancs de filtres qui revient à diviser la bande passante en plusieurs filtres à bande étroite donc des filtres que l'on pourra facilement décimer ce qui représente donc un gain de temps.



1. Sous échantillonnage et sur-échantillonnage

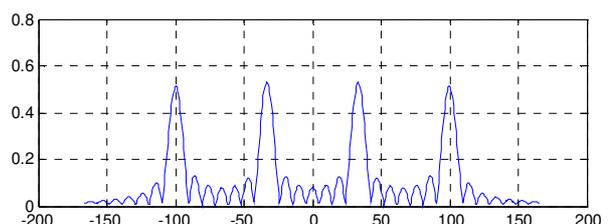
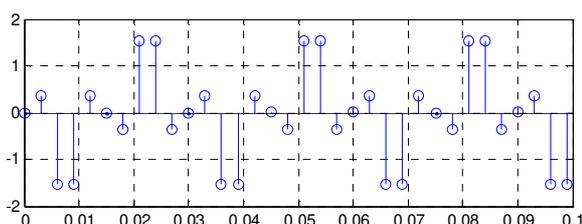
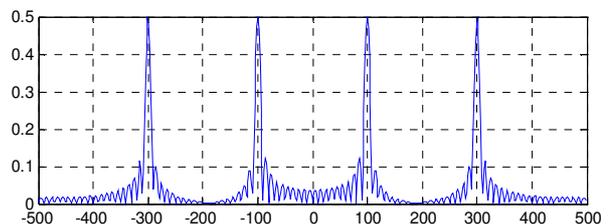
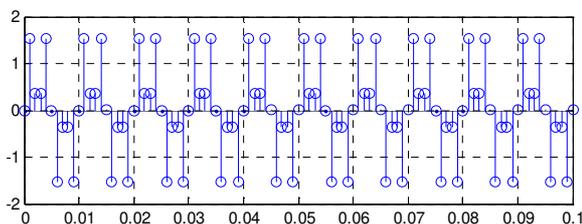
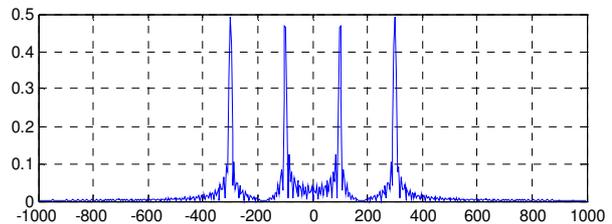
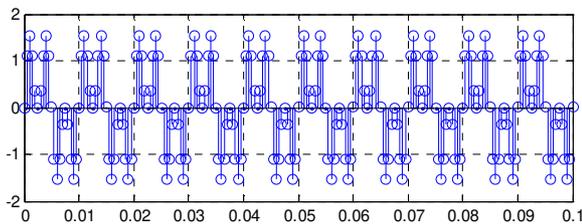
Rappelons que lorsqu'on souhaite échantillonner un signal, nous devons respecter le théorème de Shannon à savoir prendre une fréquence d'échantillonnage  $f_e / 2 > f_{max}$ . Si ce signal a subi des traitements qui ont eu pour résultat, entre autres, la modification de la fréquence maximale. On parlera alors de sous-échantillonnage ou de sur-échantillonnage.

Sous-échantillonnage : Il consiste à fabriquer à partir du signal original  $x$  un signal  $x_D$  comportant moins d'échantillons que le signal d'origine. Il faut "enlever" des échantillons, d'où le nom décimation (En anglais : down sampling). On parle de sous-échantillonnage d'un facteur  $D$  quand le signal d'origine  $x$  comporte  $N$  échantillons et le signal sur-échantillonné  $x_D$  en comporte  $N/D$ , avec  $x_D(n) = x(n/D)$

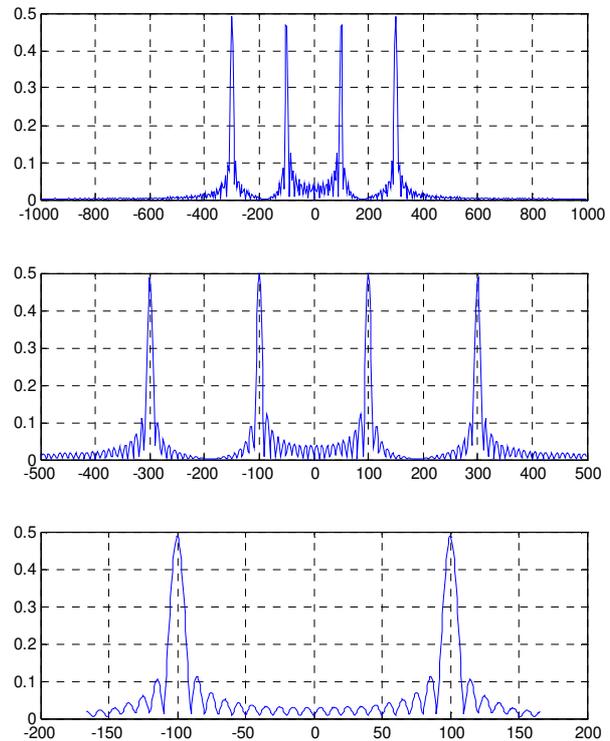
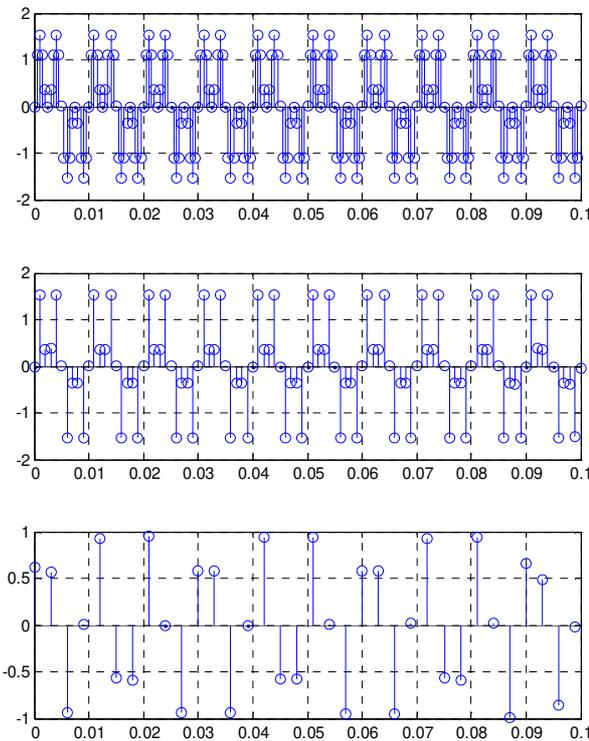


Le sous-échantillonnage naïf consiste à se débarrasser de  $D - 1$  échantillons sur  $D$ . Cela revient à devoir échantillonner directement à  $f_e / D$ , mais rien ne garantit que  $f_e / D > 2 f_{max}$ . Il y a donc un risque de repliement spectral.

Exemple : Le signal suivant est le résultat de la somme de 2 sinusoides de fréquences 100 et 300Hz. On le décime de 2 et de 6. On remarque que lorsque  $f_e = f_e/6$  il y a un repliement puisque la condition de Shannon n'est plus respectée



Pour sous-échantillonner correctement, il va falloir supprimer le contenu fréquentiel de  $x(n)$  entre  $f_e / 2D$  et  $f_e / D$  puisque le spectre devient périodique de période  $f_e / D$  et ce avant la décimation.



Dans ce cas, nous avons pour la fonction de transfert  $X_D(z) = \sum_{-\infty}^{+\infty} x_D(m)z^{-m} = \sum_{-\infty}^{+\infty} x(n/D)z^{-n/D}$

**Rappels :**

Peigne de Dirac :  $\sum_{n=-\infty}^{\infty} \delta(t - nT_e)$  périodique de période  $T_e \Rightarrow$  décomposable en série de Fourier telle que

$$C_n = \frac{1}{T_e} \Rightarrow \sum_{n=-\infty}^{\infty} \delta(t - nT_e) = \frac{1}{T_e} \sum_{n=-\infty}^{\infty} e^{2\pi j n \frac{t}{T_e}} \quad \text{En discret } \sum \delta(n - mM) = \frac{1}{M} \sum_{m=0}^{M-1} e^{2\pi j m \frac{n}{M}}$$

**La décimation de M** est équivalente à un échantillonnage de M  $\Rightarrow y(m) = y\left(\frac{n}{M}\right) = \sum x(n) \cdot \delta(n - mM)$

► Fonction de transfert  $Y(z) = \sum_{n=-\infty}^{\infty} \left( x(n) \frac{1}{M} \sum_{k=0}^{M-1} e^{\frac{2\pi j k n}{M}} \right) z^{-n/M} = \frac{1}{M} \sum_{k=0}^{M-1} \left( \sum_{n=-\infty}^{\infty} x(n) \left( e^{-\frac{2\pi j k}{M}} z^{\frac{1}{M}} \right)^{-n} \right)$

$\Rightarrow Y(z) = \frac{1}{M} \sum_{k=0}^{M-1} X\left(W_M^k z^{\frac{1}{M}}\right)$  avec  $W_M = e^{-\frac{2\pi j}{M}}$

► Réponse en fréquence

$y(m) = x(n) \cdot \sum \delta(n - mM) \Rightarrow Y(f) = X(f) * \frac{f_e}{M} \sum_{m=0}^{M-1} \delta\left(f - \frac{mf_e}{M}\right)$

$\Rightarrow Y(f) = \frac{f_e}{M} \sum_{m=0}^{M-1} X\left(f - \frac{mf_e}{M}\right) \Rightarrow$  **Périodisation de  $\frac{f_e}{M}$**

Exemples:

- Décimation de 4

$$Y(z) = \frac{1}{4} \sum_{k=0}^3 X(W_4^k z^{\frac{1}{4}}) \text{ avec } W_4 = e^{-\frac{2\pi j}{4}}$$

$$Y(z) = \frac{1}{4} \left( X(W_4^0 z^{\frac{1}{4}}) + X(W_4^1 z^{\frac{1}{4}}) + X(W_4^2 z^{\frac{1}{4}}) + X(W_4^3 z^{\frac{1}{4}}) \right)$$

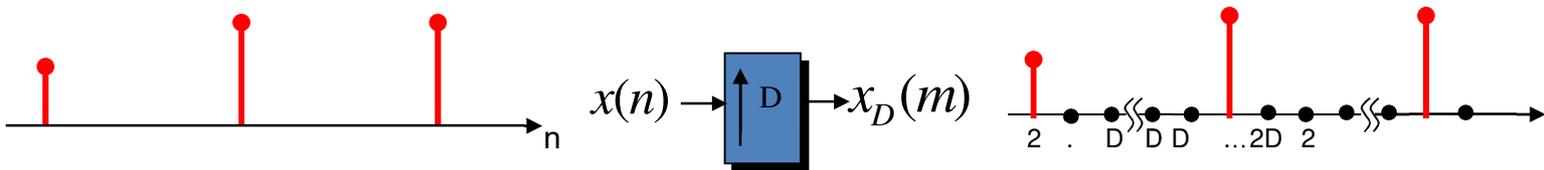
$$Y(z) = \frac{1}{4} \left( X(z^{\frac{1}{4}}) + X(-jz^{\frac{1}{4}}) + X(-z^{\frac{1}{4}}) + X(jz^{\frac{1}{4}}) \right) \Rightarrow Y(f) = \frac{1}{4} \sum_{k=0}^3 X\left(f - \frac{kf_e}{4}\right)$$

- Décimation de 2

$$Y(z) = \frac{1}{2} \sum_{k=0}^1 X(W_2^k z^{\frac{1}{2}}) \text{ avec } W_2 = e^{-\frac{2\pi j}{2}}$$

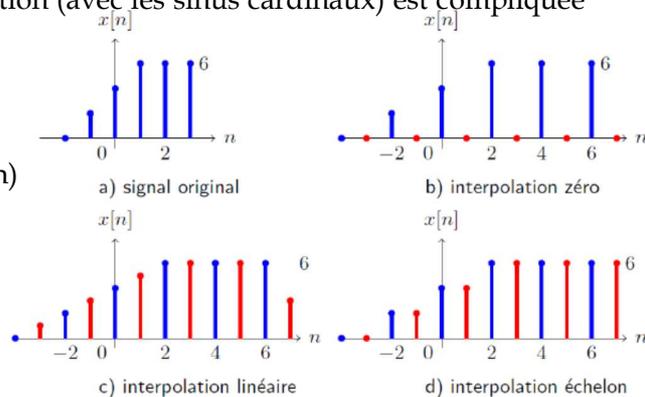
$$Y(z) = \frac{1}{2} \left( X(z^{\frac{1}{2}}) + X(-z^{\frac{1}{2}}) \right) \Rightarrow Y(f) = \frac{1}{2} \left[ X(f) + X\left(f - \frac{f_e}{2}\right) \right]$$

Sur-échantillonnage : Il consiste à fabriquer à partir du signal original  $x$  un signal  $x_M$  un signal de même durée mais comportant davantage d'échantillons que le signal d'origine. Il faut "rajouter" des échantillons, d'où le nom d'interpolation (en anglais : upsampling). On parle de sur-échantillonnage d'un facteur  $M$  quand le signal d'origine  $x$  comporte  $N$  échantillons et le signal sur-échantillonné en comporte  $MN$ , avec  $x_M(Mn) = x(n)$ .



Théoriquement, si le signal d'origine a été bien échantillonné, la reconstruction parfaite est possible, donc le sur-échantillonnage aussi. Cependant, la formule de reconstruction (avec les sinus cardinaux) est compliquée à mettre en pratique. On fait appel à des idées plus simples [25]:

- Intercaler des zéros (interpolation zéro)
- Recopier la valeur de l'instant précédent (interpolation échelon)
- Tracer une droite entre deux échantillons disponibles Adjacents (interpolation linéaire)



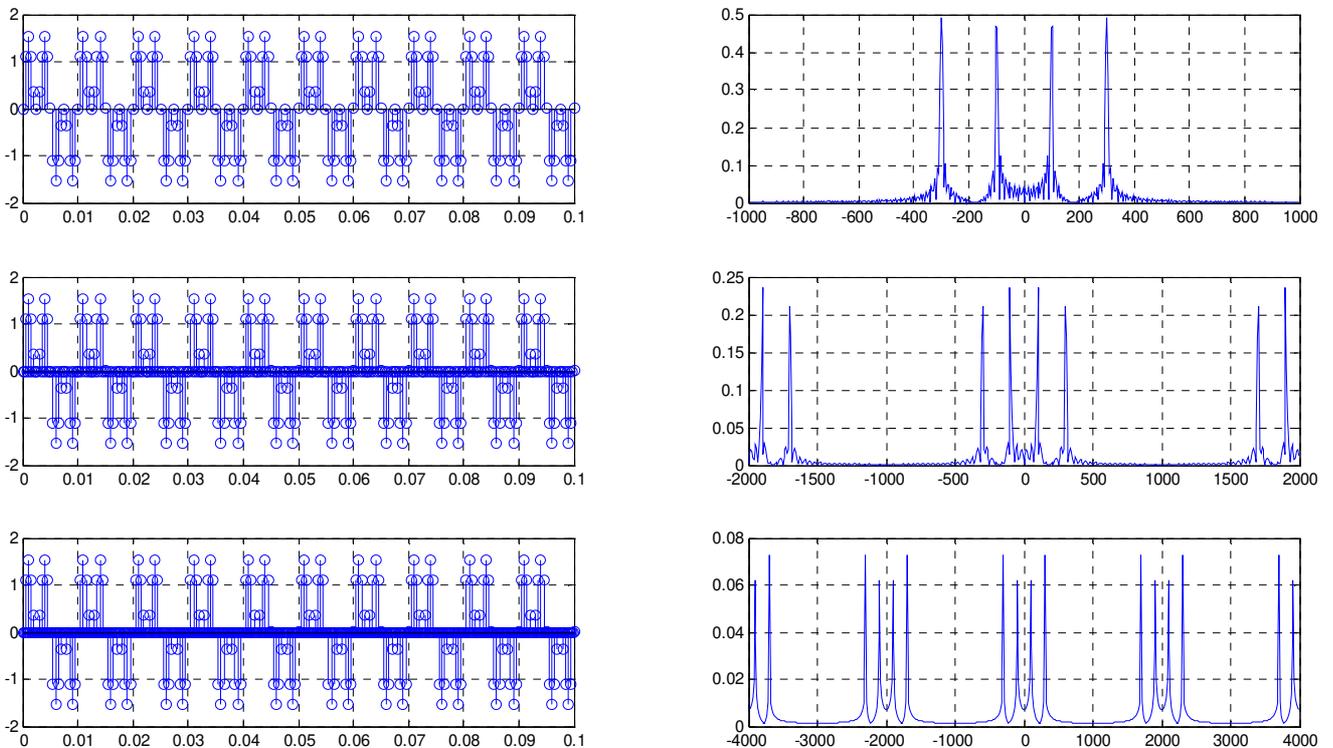
Le signal d'origine n'a pas de contenu fréquentiel au-delà de  $f_c/2$ . Le processus d'interpolation est susceptible d'en ajouter. On le fait donc suivre d'un filtrage passe-bas à la coupure  $f_c/2$  pour supprimer le contenu ajouté. C'est la raison pour laquelle la méthode d'interpolation n'a pas beaucoup d'importance.

Exercice d'application : Vérifier que l'interpolation zéro préserve le spectre du signal.

$$X_D(z) = \sum_{-\infty}^{+\infty} x_D(m)z^{-m} = \sum_{-\infty}^{+\infty} x_D(nD)z^{-nD} = \sum_{-\infty}^{+\infty} x(n)(z^{-n})^D = X(z^D)$$

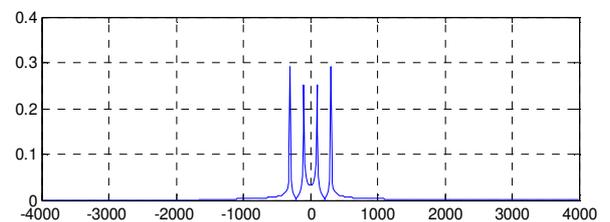
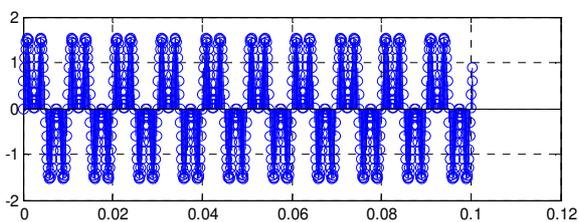
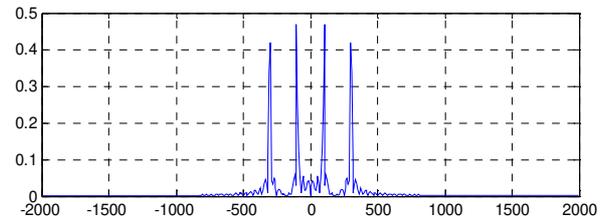
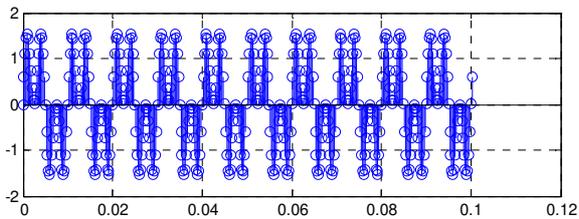
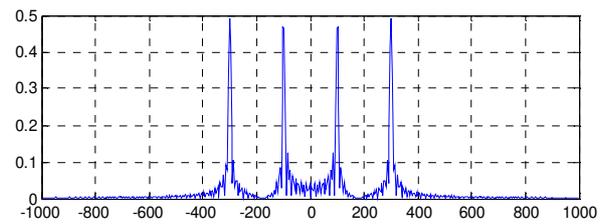
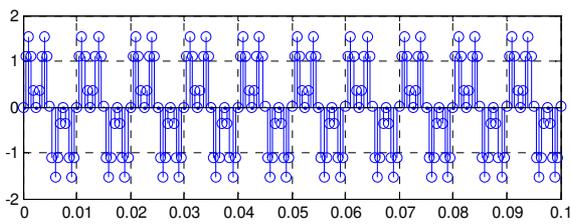
Ce qui en fréquence nous donne :  $X_D(f) = X(z = e^{2\pi jfDT} = e^{2\pi jfD/f_c})$  ce qui signifie que le signal original et sur-échantillonné ont le même spectre, on ne fait que parcourir le cercle D fois.

Exemple : On reprend le même signal que l'on interpole de 2 puis de 4.

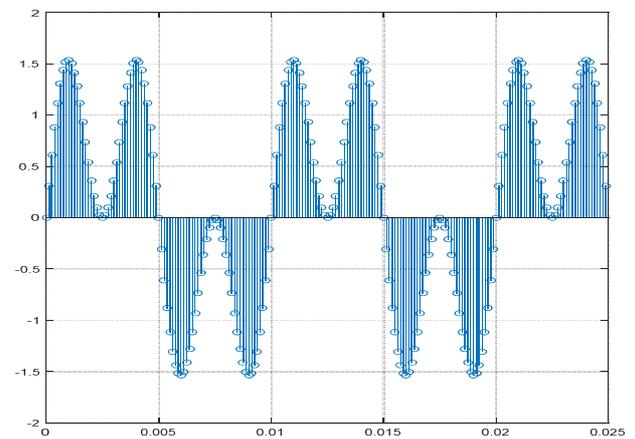
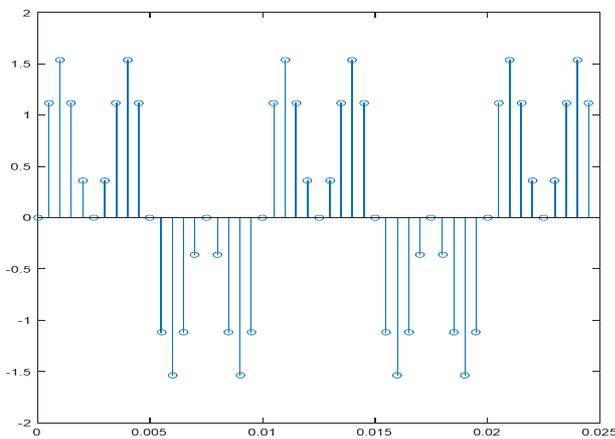


Après sur-échantillonnage, la forme du spectre reste la même aucune énergie n'a été ajoutée au signal), mais la fréquence d'échantillonnage est maintenant de  $Df_c$ . Le sur-échantillonnage fait apparaître ce que l'on appelle des spectres miroirs et pour obtenir un signal dont la forme temporelle correspondrait au signal  $x(n)$  que l'on aurait échantillonné à une fréquence  $Df_c$ .

Il faut supprimer ces spectres miroirs et donc filtrer par un filtre passe-bas le signal sur-échantillonné  $y(m)$  avec un filtre ayant une bande atténuée à partir de  $f_c/2$



Nous pouvons observer l'effet l'interpolation suivie du filtrage passe-bas anti-miroirs (équivalent à un 0 padding dans le domaine fréquentiel)



Exemples:

- Interpolation de 4  $Y(z) = X(z^4)$
- Interpolation de 2  $Y(z) = X(z^2)$

2. Décomposition polyphases

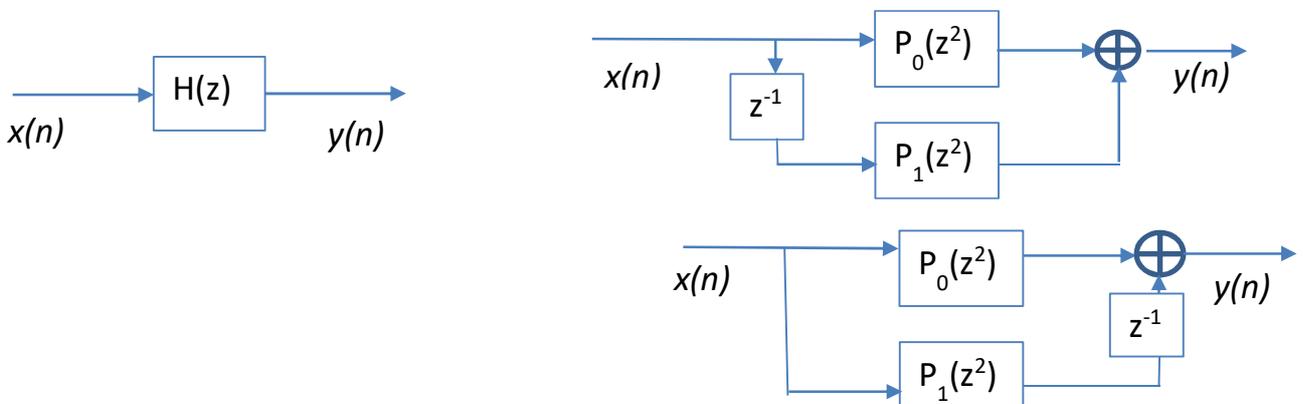
La décomposition polyphases intervient dans les bancs de filtres, elle permet une représentation simple pour le sur et sous échantillonnage [21]. Quand un signal subit une décimation d'un facteur 2, cela revient à séparer les échantillons avec un indice pair de ceux avec un indice impair, ou encore, à séparer deux phases distinctes ayant des délais (aussi appelé écarts de phase) différents dans le vecteur [22]. Si on considère une décimation d'un facteur M, le principe est exactement le même mais avec M phases distinctes.

Considérons un filtre  $h(n)$  dont la TZ est donnée par [suplec355]:  $H(z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n}$

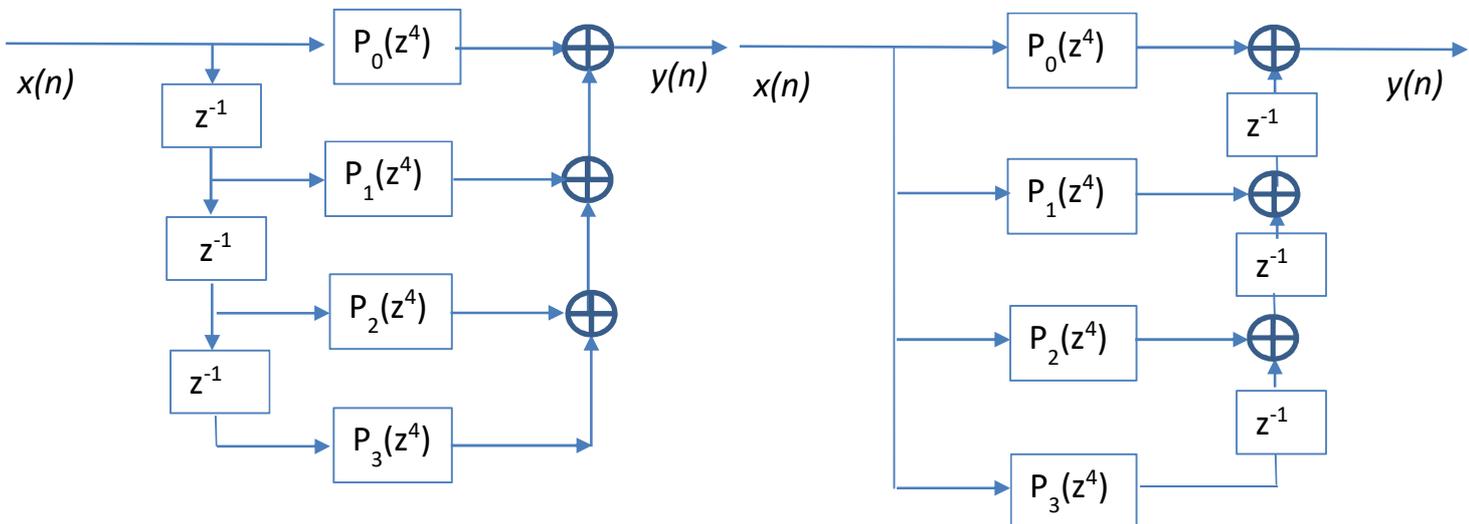
La séparation des coefficients d'indice pairs et impairs nous donnera :

$$H(z) = \sum_{n=-\infty}^{\infty} h(2n)z^{-2n} + z^{-1} \sum_{n=-\infty}^{\infty} h(2n + 1)z^{-2n}$$

En posant :  $P_0(z) = \sum_{n=-\infty}^{\infty} h(2n)z^{-n}$  et  $P_1(z) = \sum_{n=-\infty}^{\infty} h(2n + 1)z^{-n}$  alors  $H(z) = P_0(z^2) + z^{-1}P_1(z^2)$



Exemple : Décomposition polyphases pour M= 4 (structure directe et structure transposée)



D'une manière plus générale, et pour un entier M donné,  $H(z)$  peut être décomposé en :

$$H(z) = \sum_{k=0}^{M-1} z^{-k} P_k(z^M) \text{ où } P_k(z) = \sum_{n=-\infty}^{\infty} h(Mn + k)z^{-n}$$

Cette manière d'écrire la fonction de transfert  $H(z)$  est appelée la représentation polyphase du filtre  $H(z)$  et les filtres  $E_k(z)$  sont dit ses composantes polyphases. Elle permet de réaliser une fonction de transfert équivalente

avec un système dont la plupart des éléments fonctionnent à **une cadence réduite**.

**Décimation:** Prenons le cas du décimateur [20], si on considère un filtre FIR de longueur N multiple de M (Rajouter des zéros si nécessaire pour cela) ayant pour réponse impulsionnelle h(n), dont la transformée en Z est donnée par  $H_M(z) = \sum_{n=0}^{N-1} h(n)z^{-n} = \sum_{k=0}^{M-1} \sum_{l=0}^{\frac{N}{M}-1} h(lM+k) z^{-(lM+k)}$

$$H_M(z) = \sum_{k=0}^{M-1} z^{-k} \sum_{l=0}^{\frac{N}{M}-1} h(lM+k) (z^M)^{-l} = \sum_{k=0}^{M-1} z^{-k} P_k(z^M)$$

Où  $P_k(z) = \sum_{l=0}^{\frac{N}{M}-1} h(lM+k) z^{-l}$

En considérant le schéma bloc d'un banc de filtres, on voit que le filtrage est classiquement appliqué avant la décimation. Cela signifie donc que le signal est filtré par les M filtres et que l'on ne garde ensuite qu'une phase parmi les M de chaque résultat de filtrage. Cette manière de faire la transformation n'est pas efficace car elle entraîne beaucoup de calculs qui ne seront pas exploités par la suite. La notion de représentation polyphase revient à inverser ce schéma en faisant en premier lieu l'opération de sous-échantillonnage, puis une opération de filtrage.

Identité noble 1 [section 10]



Identité noble 2



Exemple illustratif pour la décimation

Soit le signal x(n) à décimer de 3 avec un filtre passe-bas h(n) dont la fonction de transfert est

$$H(z) = h(0) + h(1)z^{-1} + h(2)z^{-2} + h(3)z^{-3} + \dots + h(11)z^{-11}$$

La première étape est de décomposer le filtre h(n) en trois filtres P<sub>0</sub>(z<sup>3</sup>), P<sub>1</sub>(z<sup>3</sup>) et P<sub>2</sub>(z<sup>3</sup>). Considérons la première composante du filtre polyphases P<sub>0</sub>(z<sup>3</sup>) = h(0) + h(3)z<sup>-3</sup> + h(6)z<sup>-6</sup> + h(9)z<sup>-9</sup>



On a  $X(z) = x(0) + x(1)z^{-1} + x(2)z^{-2} + x(3)z^{-3} + \dots + x(n)z^{-n} + \dots$

Ainsi

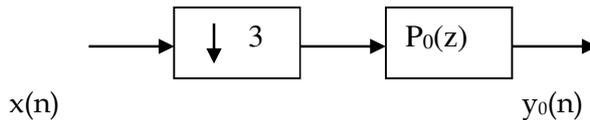
$$y_0(n) = x(0)h(0) + x(1)h(0)z^{-1} + x(2)h(0)z^{-2} + [x(3)h(0) + x(0)h(3)]z^{-3} + [x(4)h(0) + x(1)h(3)]z^{-4} + [x(5)h(0) + x(2)h(3)]z^{-5} + [x(6)h(0) + x(3)h(3) + x(0)h(6)]z^{-6} + [x(7)h(0) + x(4)h(3) + x(1)h(6)]z^{-7} + [x(8)h(0) + x(5)h(3) + x(2)h(6)]z^{-8} + [x(9)h(0) + x(6)h(3) + x(3)h(6) + x(0)h(9)]z^{-9} + [x(10)h(0) + x(7)h(3) + x(4)h(6) + x(1)h(9)]z^{-10} + [x(11)h(0) + x(8)h(3) + x(5)h(6) + x(2)h(9)]z^{-11} + [x(12)h(0) + x(9)h(3) + x(6)h(6) + x(3)h(9)]z^{-12} + \dots;$$

Puisque, le filtre est suivi d'une décimation par 3, on ne retiendra que les éléments de puissance multiple de 3 (en rouge) ce qui au final revient à

$$y_0(n) = x(0)h(0) + [x(3)h(0) + x(0)h(3)]z^{-1} + [x(6)h(0) + x(3)h(3) + x(0)h(6)]z^{-2} + [x(9)h(0) + x(6)h(3) + x(3)h(6) + x(0)h(9)]z^{-3} + [x(12)h(0) + x(9)h(3) + x(6)h(6) + x(3)h(9)]z^{-4} + \dots;$$

Qui n'est rien d'autre que le résultat de la convolution du signal  $x(n)$  décimé par 3  $\{x(0), x(3), x(6), \dots\}$  par  $P_0(z) = h(0) + h(3)z^{-1} + h(6)z^{-2} + h(9)z^{-3}$

Ce qui correspond au schéma suivant :

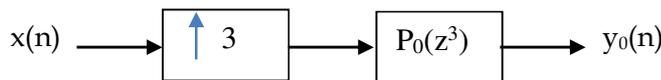


Exemple illustratif pour l'interpolation

Soit le  $x(n)$  à interpoler de 3 avec un filtre passe-bas  $h(n)$  ayant pour fonction de transfert:

$$H(z) = h(0) + h(1)z^{-1} + h(2)z^{-2} + h(3)z^{-3} + \dots + h(11)z^{-11} = P_0(z^3) + z^{-1}P_1(z^3) + z^{-2}P_2(z^3)$$

La première étape est de décomposer le filtre  $h(n)$  en trois filtres  $P_0(z^3)$ ,  $P_1(z^3)$  et  $P_2(z^3)$ . Considérons la première composante du filtre polyphase  $P_0(z^3) = h(0) + h(3)z^{-3} + h(6)z^{-6} + h(9)z^{-9}$



On a  $x(n) = \{x(0), x(1), x(2), x(3), \dots\} \Rightarrow X(z) = x(0) + x(1)z^{-1} + x(2)z^{-2} + x(3)z^{-3} + \dots + x(n)z^{-n} + \dots$

La décomposition polyphases du filtre nous donne

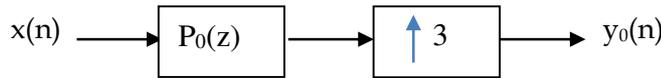
$$H(z) = h(0) + h(1)z^{-1} + h(2)z^{-2} + h(3)z^{-3} + \dots + h(11)z^{-11} = P_0(z^3) + z^{-1}P_1(z^3) + z^{-2}P_2(z^3)$$

Avec  $P_0(z^3) = h(0) + h(3)z^{-3} + h(6)z^{-6} + h(9)z^{-9}$

Après interpolation  $\Rightarrow x_0(n) = \{x(0), 0, 0, x(1), 0, 0, x(2), 0, 0, x(3)\} \Rightarrow X_0(z) = x(0) + x(1)z^{-3} + x(2)z^{-6} + x(3)z^{-9} + x(4)z^{-12} + \dots$

D'où  $y_0(n) = x(0)h(0) + [x(1)h(0) + x(0)h(3)]z^{-3} + [x(2)h(0) + x(1)h(3) + x(0)h(6)]z^{-6} + [x(3)h(0) + x(2)h(3) + x(1)h(6) + x(0)h(9)]z^{-9} + [x(4)h(0) + x(3)h(3) + x(2)h(6) + x(1)h(9)]z^{-12} + \dots$

Reprenons le schéma équivalent



Sachant que  $P_0(z^3) = h(0) + h(3)z^{-3} + h(6)z^{-6} + h(9)z^{-9} \Rightarrow P_0(z) = h(0) + h(3)z^{-1} + h(6)z^{-2} + h(9)z^{-3}$

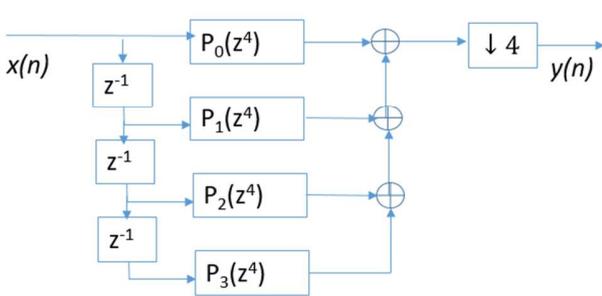
Et que  $X(z) = x(0) + x(1)z^{-1} + x(2)z^{-2} + x(3)z^{-3} + \dots + x(n)z^{-n} + \dots$

Alors  $x_0(n) = x(0)h(0) + [x(1)h(0) + x(0)h(3)]z^{-1} + [x(2)h(0) + x(1)h(3) + x(0)h(6)]z^{-2} + [x(3)h(0) + x(2)h(3) + x(1)h(6) + x(0)h(9)]z^{-3} + [x(4)h(0) + x(3)h(3) + x(2)h(6) + x(1)h(9)]z^{-4} + \dots$

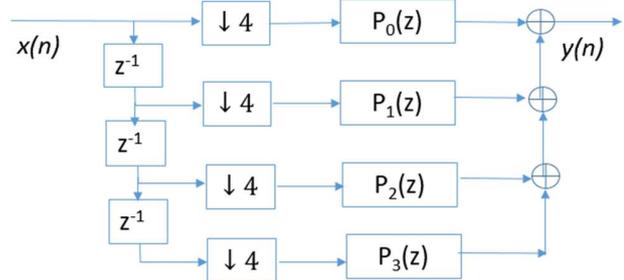
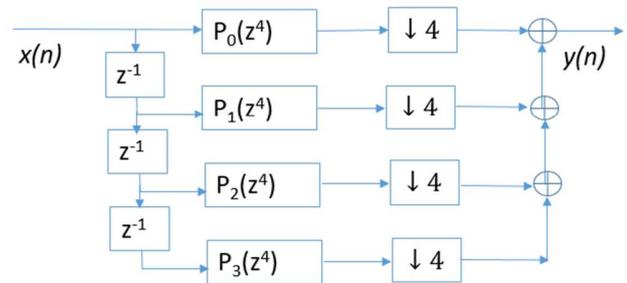
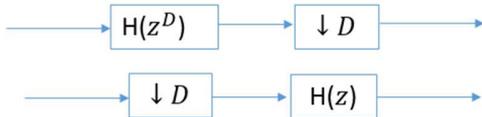
Après Interpolation par 3  $\Rightarrow y_0(n) = x(0)h(0) + 0z^{-1} + 0z^{-2} + [x(1)h(0) + x(0)h(3)]z^{-3} + 0z^{-4} + 0z^{-5} + [x(2)h(0) + x(1)h(3) + x(0)h(6)]z^{-6} + 0z^{-7} + 0z^{-8} + [x(3)h(0) + x(2)h(3) + x(1)h(6) + x(0)h(9)]z^{-9} + 0z^{-10} + 0z^{-11} + [x(4)h(0) + x(3)h(3) + x(2)h(6) + x(1)h(9)]z^{-12} + \dots$

**Exemple** : Décomposition polyphases pour une décimation M=4

Ainsi, considérons un décimateur de 4 équipé d'un filtre anti-recouvrement H(z) auquel on applique la décomposition polyphases :



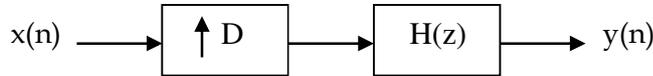
**Structure Directe**



En appliquant la première identité noble, on obtient une forme permutée permettant d'effectuer le filtrage à des taux plus faibles

Ainsi, chaque ligne de la décomposition polyphases constitue une réponse à un taux M fois plus faible, déphasée d'un échantillon par rapport à la ligne précédente.

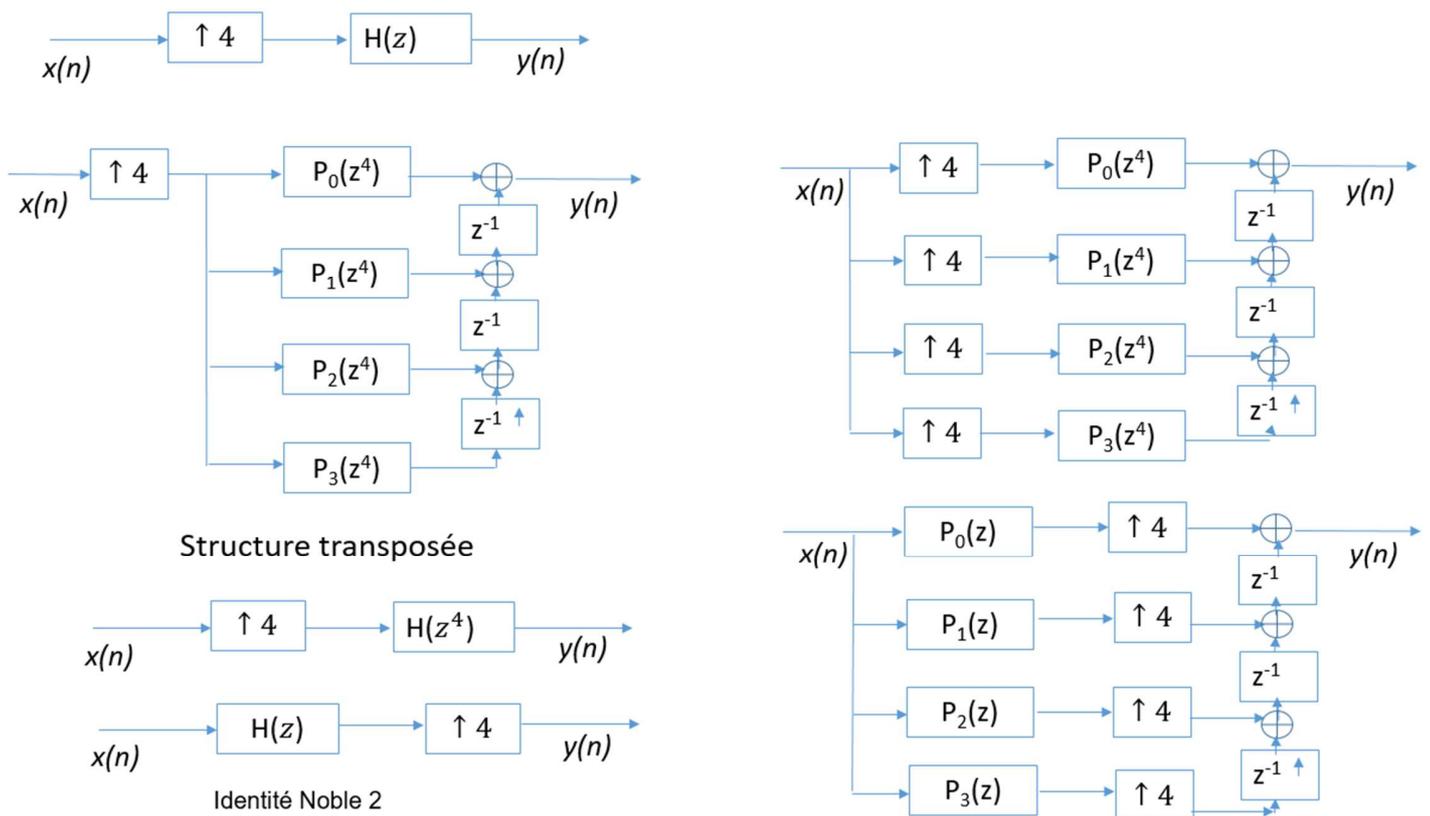
**Interpolation** : En suivant une démarche similaire on peut construire la matrice polyphases d'un banc de filtres de synthèse. Dans ce cas, l'opération de sur échantillonnage qui précède le filtrage est remplacée par une étape de filtrage suivie d'un sur-échantillonnage. On opère de la même façon pour l'interpolateur équipé d'un filtre d'interpolation  $H(z)$ .



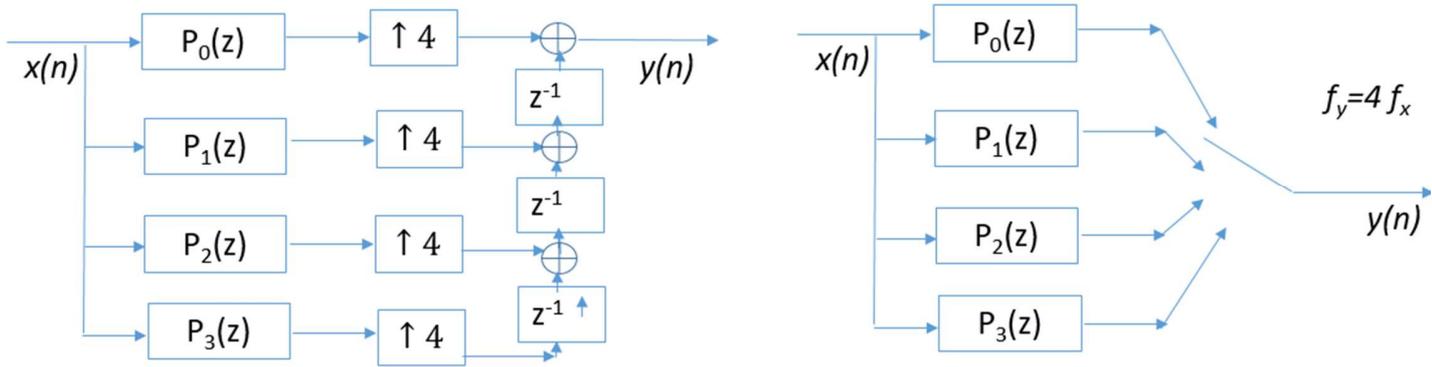
**Exemple** : Décomposition polyphases pour une interpolation  $D=4$ . On utilisera la structure transposée plus pratique dans ce cas.

En appliquant la deuxième identité noble, on gagne en termes de temps de traitement puisque tout le filtrage s'effectue au taux le plus faible.

Ce type de matrice polyphases, correspondant à un banc de filtres de synthèse, est parfois appelé matrice polyphases de type 2 par opposition avec les matrices polyphases de type 1 qui correspondent à un banc de filtres d'analyse [22].



**Remarque** : A chaque instant d'échantillonnage en sortie du système une seule composante polyphase est non nulle. Cela conduit au modèle de commutateur où on ne considère à chaque instant d'échantillonnage que la composante non nulle [24]. Un commutateur en sortie peut jouer le rôle des délais et additionneurs. En effet, l'interpolation par  $D = 4$ , dans cet exemple, insère 3 zéros sur 4. A un instant donné, une seule branche contribue à la sommation et les autres sont à zéro.

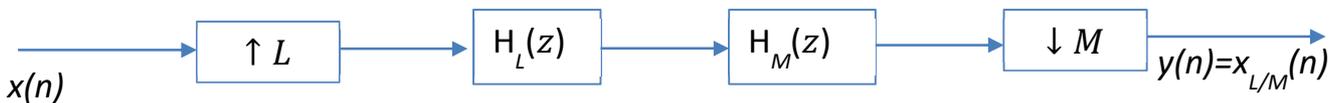


### 3. Changement de fréquence rationnel

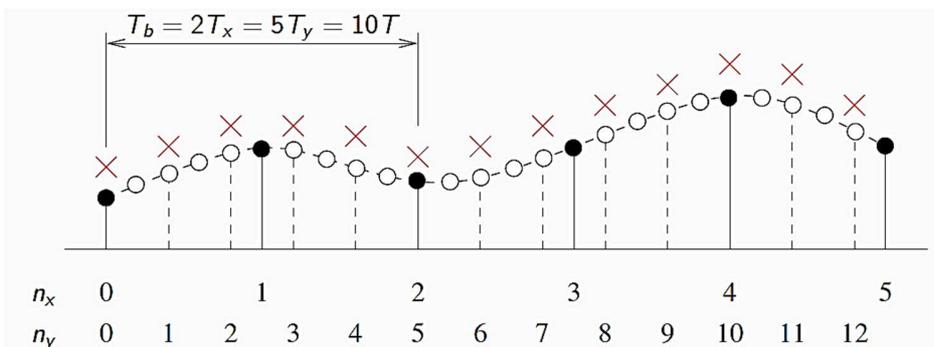
Supposons que l'on veuille changer numériquement la fréquence d'échantillonnage d'un signal  $x(n)$  et passer d'une fréquence d'échantillonnage  $f_1$  à une fréquence d'échantillonnage  $f_2$  où  $f_2/f_1=L/M$ . Cela conduit donc à sur-échantillonner le signal  $x(n)$  d'un facteur  $L$  puis à le décimer d'un facteur  $M$ .

- Le sur-échantillonnage par  $L$  doit être suivie d'un filtre passe-bas coupant à  $f_c/2L$  pour supprimer les spectres dus au sur-échantillonnage où  $f_c$  est la fréquence d'échantillonnage du signal de sortie.
- La décimation par  $M$  doit être précédée d'un filtrage anti-repliement ayant une fréquence de coupure de  $f_c/2M$ , si  $f_c$  est la fréquence d'échantillonnage du signal d'entrée.

On ne conservera alors qu'un seul des deux filtres passe-bas : celui dont la fréquence de coupure est la plus basse



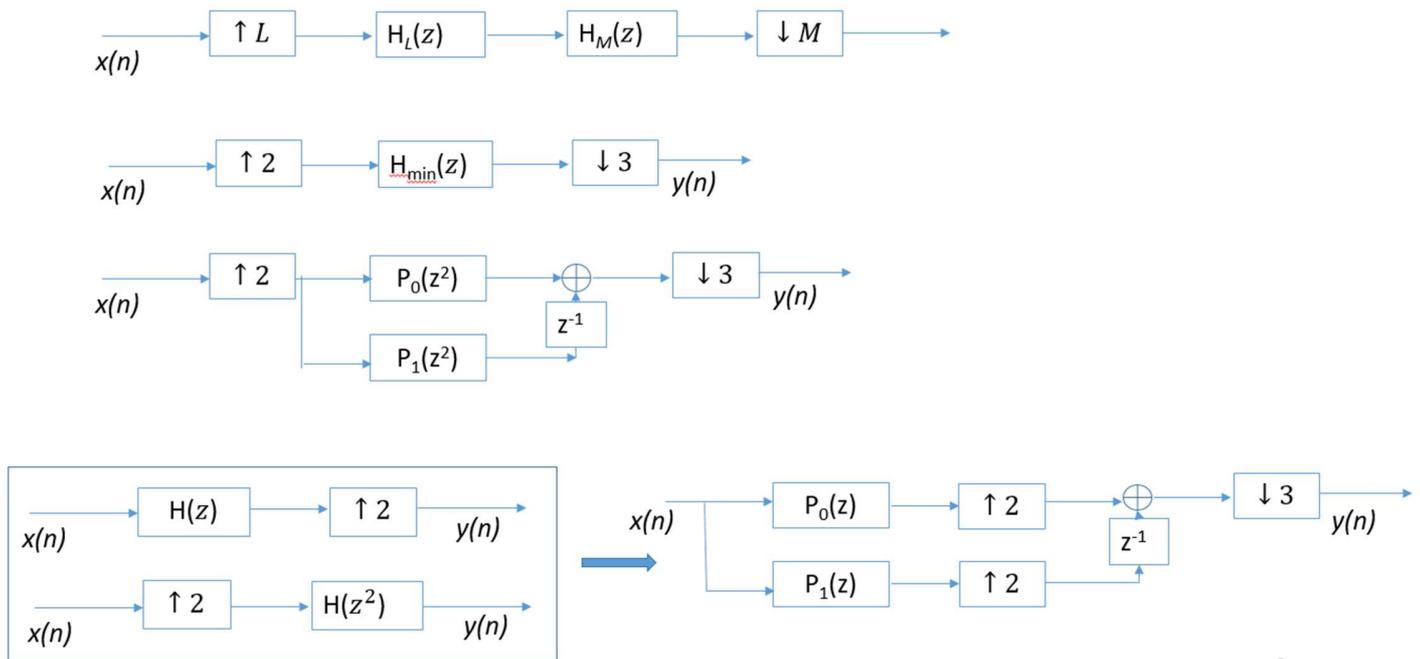
Remarque : Si  $L$  et  $M$  sont premiers entre eux, décimation et interpolation sont commutatifs, ci-dessous un exemple : de conversion où  $L = 5$  et  $M = 2$  [31]



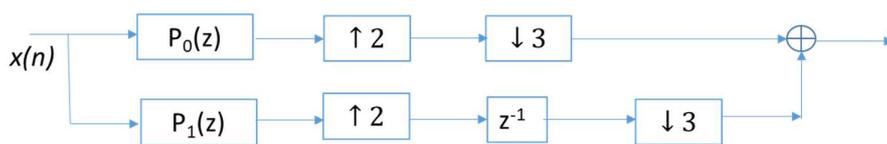
En outre, d'après la théorie des nombres, si  $L$  et  $M$  sont premiers entre eux alors ils existent deux entiers  $l_0$  et  $m_0$  que  $l_0L - m_0M = -1 \rightarrow z^{-1} = z^{-(l_0L - m_0M)} = z^{-l_0L} z^{m_0M}$

**Exemple :** On souhaite sur une réduction de fréquence d'échantillonnage d'un signal audio d'un facteur 2/3 (passage de 48kHz à 32kHz). Voici la structure polyphase conduisant à évaluer le filtre à la fréquence la plus basse possible. En appliquant la décomposition polyphase pour le sur-échantillonnage, on obtient la structure ci-contre). A noter que décimation et interpolation ne sont pas commutatives sauf si L et M sont premiers entre eux [20].

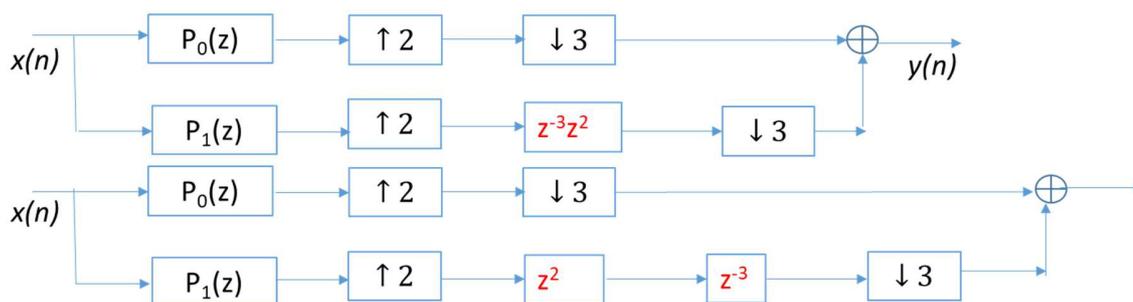
On commence par garder le filtre passe-bas dont la fréquence de coupure est la plus petite. Puis on décompose le filtre suivant l'interpolation.



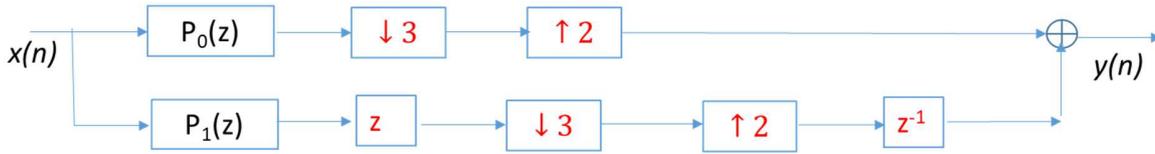
On introduit la décimation avant les additionneurs ce qui nous donne :



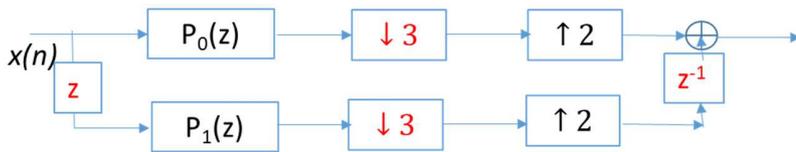
On utilise  $z^{-1} = z^{-(l_0L - m_0M)} = z^{-l_0L} z^{m_0M}$  ce qui nous donne  $z^{-1} = z^{-3} z^2$



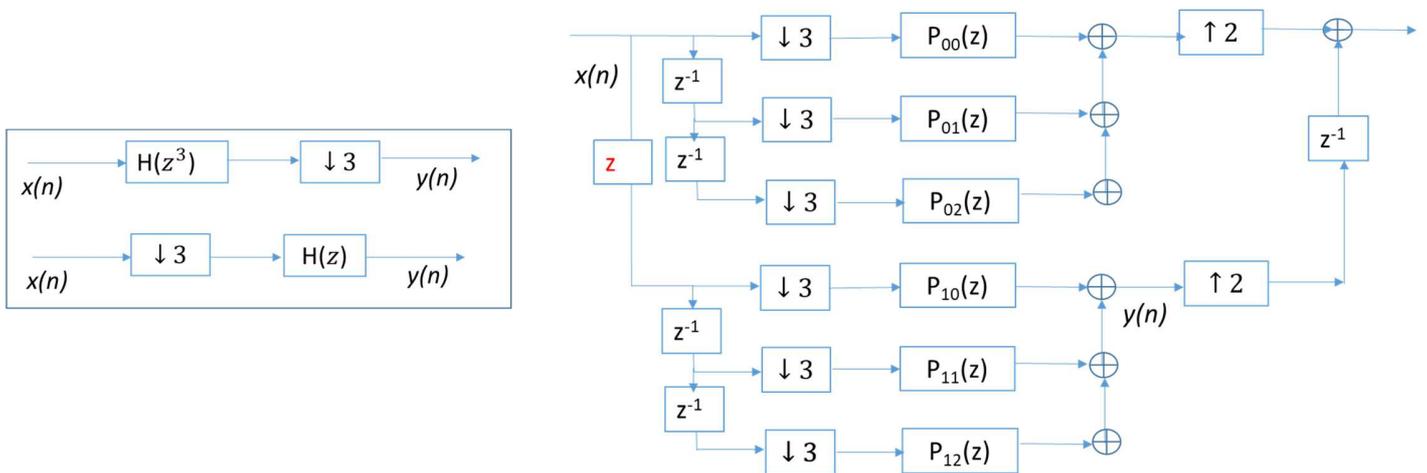
On applique les identités noble et on permute décimation et interpolation



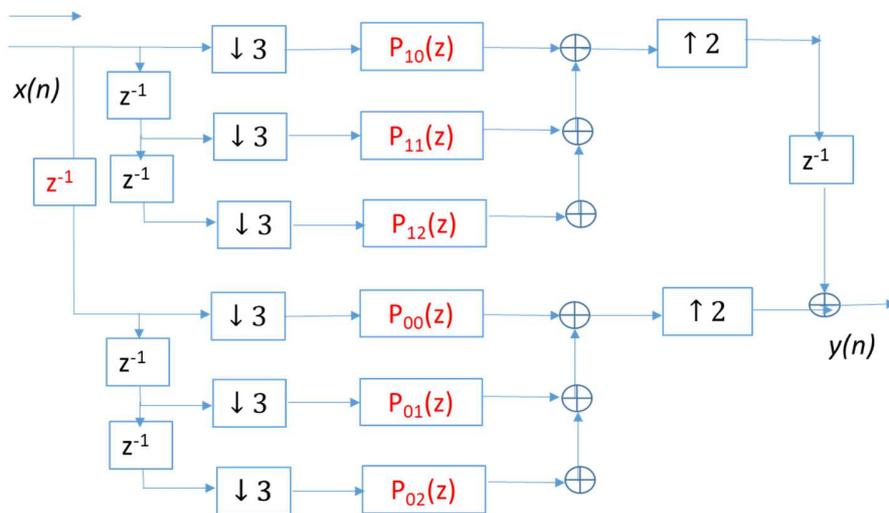
Que l'on peut schématiser aussi



Puis on développe la décomposition polyphases pour la partie décimation en exploitant la deuxième identité noble.



On peut observer que le deuxième bloc est en avance par rapport au premier, autrement dit le premier est en retard par rapport au deuxième ce qui nous permet de donner la décomposition polyphases ci-dessous (composée uniquement de retardateurs)



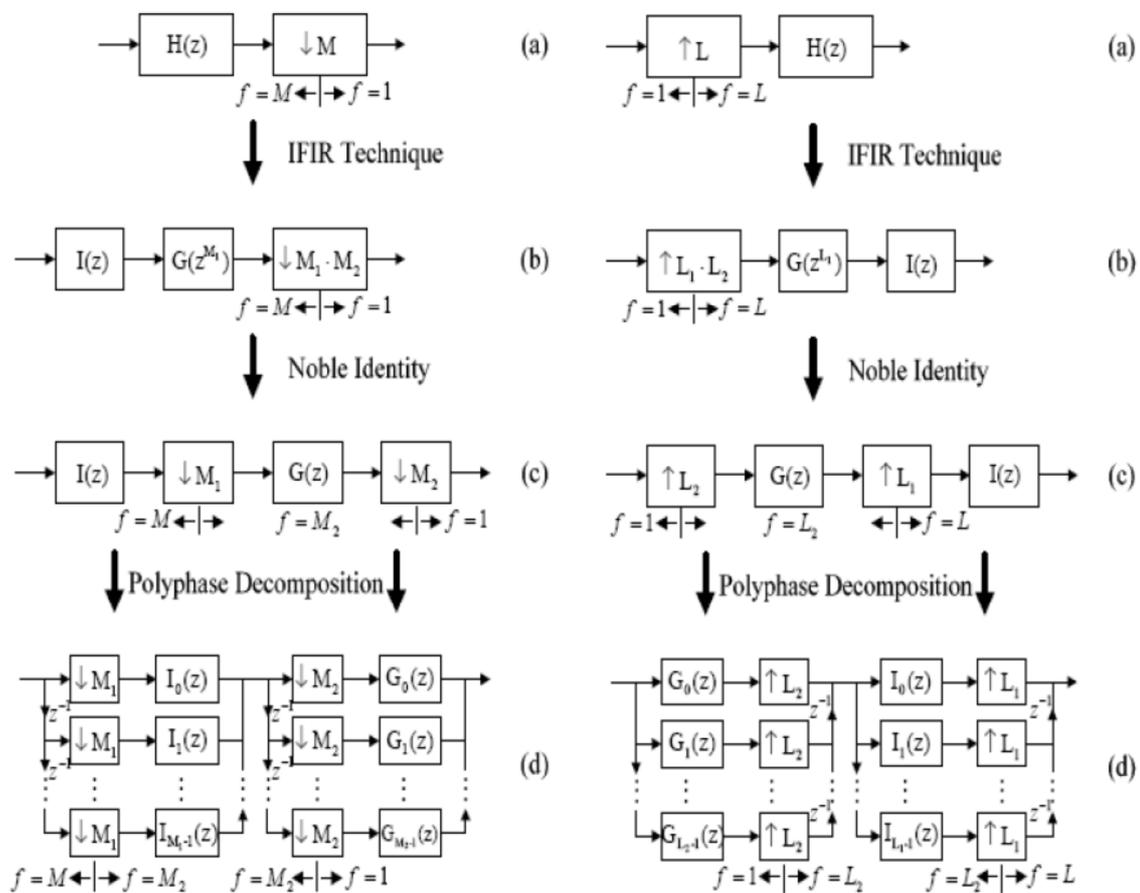
Remarques

Un filtre de décimation/interpolation à bande passante très étroite est complexe à mettre en œuvre (Ordre élevé)

. Il peut être remplacé par la mise en cascade de filtres plus simples. Les spécifications des filtres individuels sont considérablement assouplies puisque la spécification globale du filtre est partagée entre plusieurs filtres d'ordre inférieur.

Ainsi, pour un facteur de décimation ou interpolation, il est plus judicieux d'opérer par paliers (multi-étages).

Si on doit décimer de 15, cela nécessite un filtre d'ordre élevé. On peut alors procéder en décimant par 3 puis par 5. Ce qui nous permettra d'employer 2 filtres de plus faible ordre. De même, deux décompositions polyphases successives de 3 et 5 sont moins coûteuses qu'une décomposition polyphases de 15.



Ainsi, si l'on doit réaliser un Filtre FIR avec des spécifications :  $f_e = 40$  MHz,  $f_c = 300$  KHz,  $f_p=400$ KHz,  $\delta_p = 0,1$ db,  $\delta_s = 60$ db.

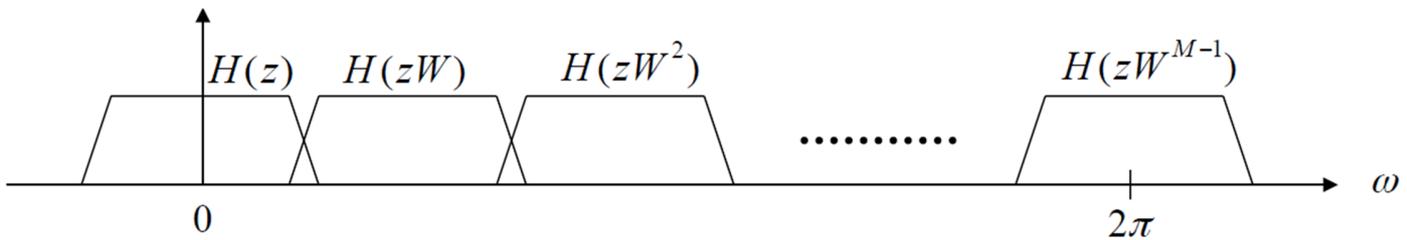
Pour une décimation de 64, la meilleure factorisation multi-étages est S (nombre d'étages) = 3, ou 6, et chaque étape du facteur de décimation de 4 ou 2, respectivement.

$f_s$	$f_c$	$f_{st}$	M	S	$\delta_p$	$\delta_s$	Order of FIR N	
40MHz	300 KHz	400 KHz	64	1	0.1	60 dB	1087	
	2.4MHz, 300KHz	3.2MHz, 400KHz	8*	2	0.05	60 dB	147	
			8		0.05		147	
	4MHz, 1MHz, 300KHz	3.2MHz, 0.8MHz, 400KHz	4*	3	0.03	60 dB	77	
			4*		0.03		77	
			4		0.03		77	
	9.6MHz, 4.8MHz, 2.4MHz, 1.2MHz, 600KHz, 300KHz	12.8MHz, 6.4MHz, 3.2MHz, 1.6MHz, 800KHz, 400KHz	2*	6	6	0.017	60 dB	40
			2*			0.017		40
			2*			0.017		40
			2*			0.017		40
		2*	0.017			40		
		2	0.017			40		

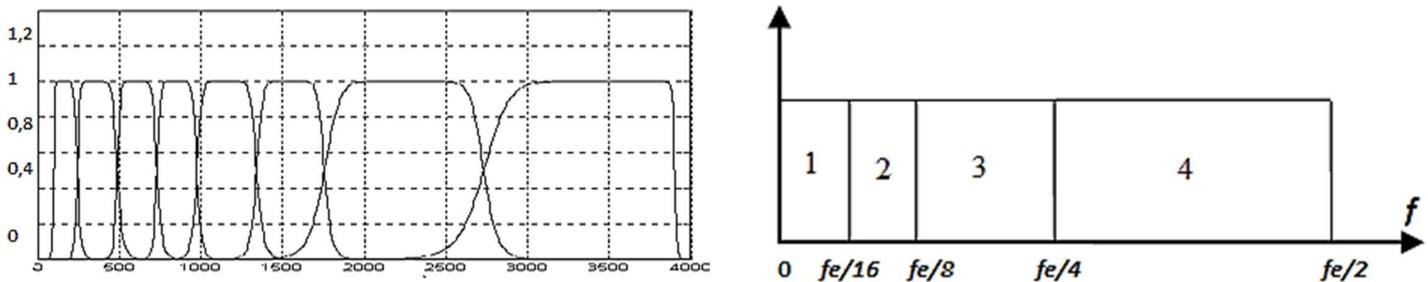
### 3. Bancs de filtres

Un banc de filtre est un ensemble de filtres numériques travaillant en parallèle et découpant la bande de fréquence en K sous bandes. Le traitement du signal numérique traditionnel utilise des blocs comme : l'additionneur, multiplicateurs et des retardateurs. Dans les systèmes banc de filtres, nous avons deux nouveaux blocs que sont le décimateur et l'interpolateur [23].

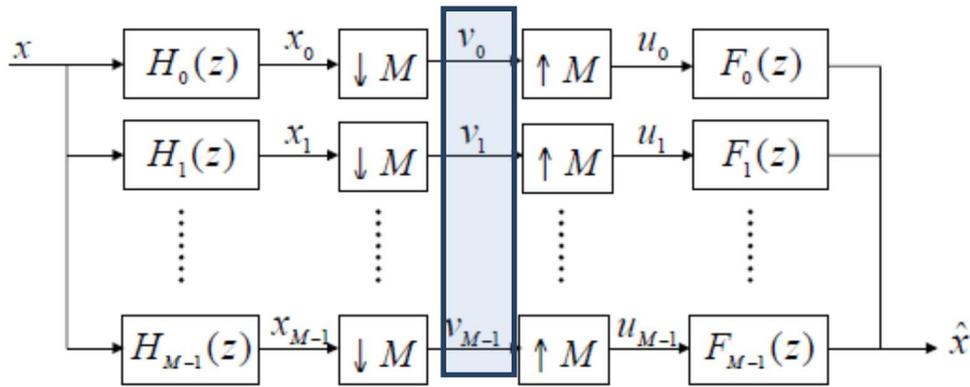
- Un banc de filtres est un ensemble de filtres, avec une entrée ou une sortie commune
- Pour constituer un banc de filtres, on peut créer et décaler un filtre générateur
- Un banc de filtres est un ensemble de filtres  $(H_0(z); \dots; H_{M-1}(z))$  tel que tel que les bandes passantes des filtres forment (+/-) une partition de l'ensemble des fréquences et leurs réponses en fréquence se déduisent les unes des autres par certaines relations telles que  $H_k(z) = H_0\left(ze^{-\frac{2\pi jk}{M}}\right) = H_0(zW^k)$



- Autres décompositions : échelle de Mel (psychoacoustique), Ondelettes etc



- Chaque sortie du filtre d'un banc est à bande étroite, on peut donc la décimer pour la traiter plus rapidement.
- Un banc de filtres est dit "à reconstruction parfaite" s'il existe un ensemble de filtres (dits "de synthèse") qui, lorsqu'on les applique aux sorties du banc initial et qu'on somme leurs sorties, reconstruit parfaitement le signal d'entrée (à un retard près).

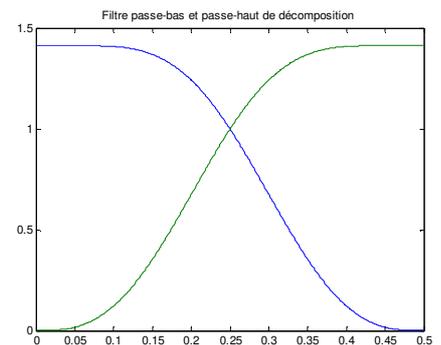
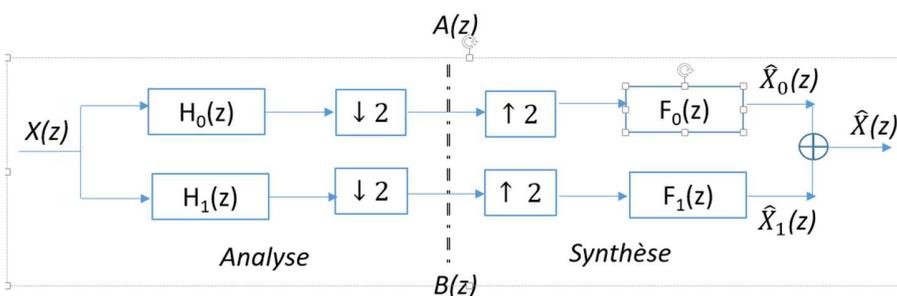


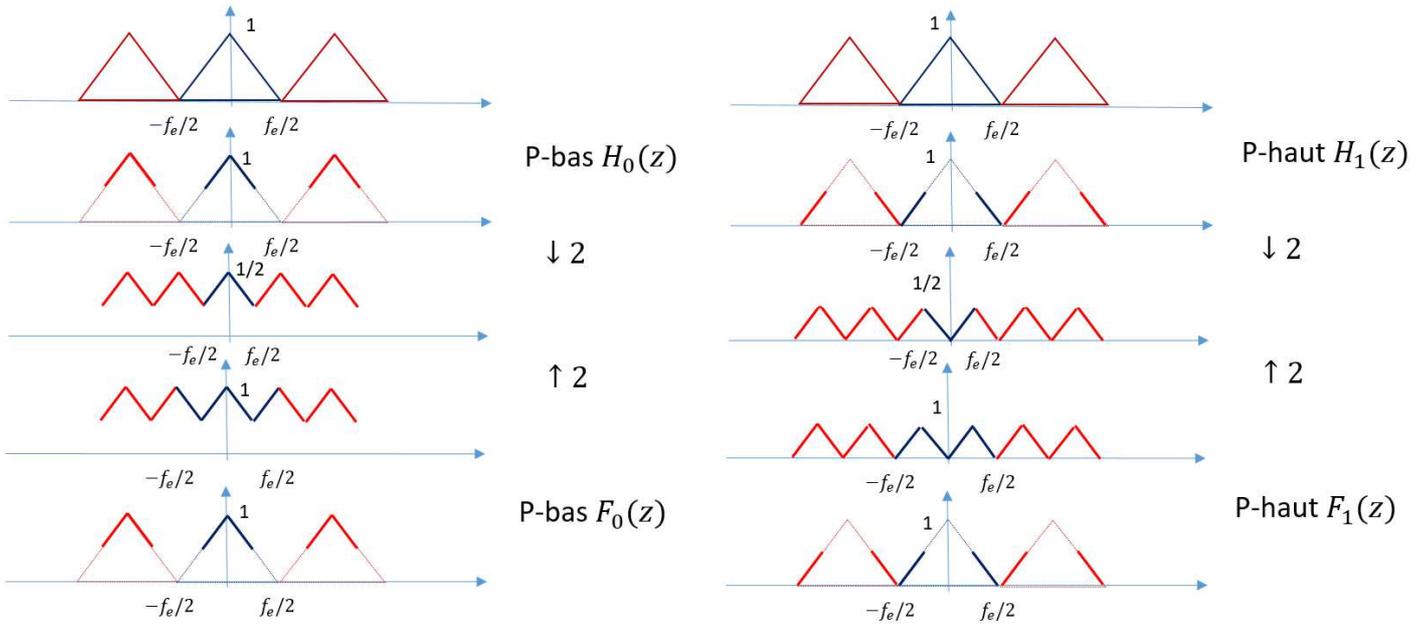
Ces deux cas sont représentés sur la figure ci-dessus. Ce système est appelé banc de filtres d'analyse, et les filtres  $H_k(z)$  sont les filtres d'analyse. Ce banc décompose le signal  $x(n)$  en  $M$  signaux  $v_i(n)$  appelés signaux de sous-bandes. Le système de gauche est appelé banc de filtres de synthèse et les filtres  $F_k(z)$  sont les filtres de synthèse. Il combine les  $M$  signaux  $w_k(n)$  en un seul signal  $y(n)$ [21]

Les bancs de filtres d'analyse et de synthèse sont généralement associés, le premier décompose un signal pour appliquer un traitement à chaque signal de sous-bande et le second re-combine les signaux de sous-bandes traités pour construire le signal modifié. Un tel système d'analyse/synthèse est appelé banc de filtres à reconstruction parfaite quand en l'absence de tout traitement dans les sous-bandes, le signal de sortie  $y(n)=x(n-k)$ . Ces filtres sont alors dits bi-orthogonaux, ils sont étroitement liés à la théorie des ondelettes objet du prochain chapitre.

Exemple d'application : Banc de filtres à deux canaux [20]

Dans un banc de filtres à 2 canaux on sépare le signal d'entrée en un signal passe-haut et un signal passe-bas, respectivement  $d(n)$  et  $a(n)$ . On appelle  $H_0(z)$  et  $H_1(z)$  les filtres passe bas et passe haut du banc d'analyse et  $F_0(z)$  et  $F_1(z)$  les filtres passe bas et passe haut du banc de synthèse. On appelle  $X(z)$  la transformée en  $Z$  du signal  $x(n)$ . L'inconvénient de cette structure est que le nombre d'échantillon est multiplié par 2 en sortie du banc d'analyse. Si les filtres du banc d'analyse  $H_0(z)$  et  $H_1(z)$  ont des gabarits correspondant à ceux représentés ci-contre, il est alors possible d'introduire un échantillonnage critique en décimant par deux en sortie des filtres du banc du d'analyse.





On remarquera qu'il n'y a pas de filtre anti-repliement lors de la décimation, en fait on se sert du fait que le spectre soit à bande limitée (ici la moitié de la bande de fréquence). C'est un cas particulier où le signal est à bande limité égale à  $f_e/M$ . Ainsi on peut alors récupérer le signal original à partir du signal sous échantillonné. De même pour le sur échantillonnage puisque on se sert de filtres de reconstruction qui eux aussi occupent la moitié de la bande de fréquence.

On peut vérifier que pour  $H_1(z) = H_0(-z^{-1})$ , on obtient  $H_1(f) = H_0(\frac{f_e}{2} - f)$

►  $A(z) = \left(\frac{1}{2} \sum_{k=0}^1 H_0 \left(W_2^k z^{\frac{1}{2}}\right) X \left(W_2^k z^{\frac{1}{2}}\right)\right)$  avec  $W_2^1 = e^{-\frac{2\pi j}{2}} = -1$

$\Rightarrow A(z) = \frac{1}{2} \left( H_0 \left(z^{\frac{1}{2}}\right) X \left(z^{\frac{1}{2}}\right) + H_0 \left(-z^{\frac{1}{2}}\right) X \left(-z^{\frac{1}{2}}\right) \right)$

►  $\hat{X}_0(z) = A(z^2)F_0(z) \Rightarrow \hat{X}_0(z) = F_0(z) \frac{1}{2} \left( H_0(z)X(z) + H_0(-z)X(-z) \right)$

►  $\hat{X}_1(z) = F_1(z) \frac{1}{2} \left( H_1(z)X(z) + H_1(-z)X(-z) \right)$

$\Rightarrow \hat{X}(z) = \frac{1}{2} \left( F_0(z)H_0(z) + F_1(z)H_1(z) \right) X(z) + \frac{1}{2} \left( F_0(z)H_0(-z) + F_1(z)H_1(-z) \right) X(-z)$

Rappelons que l'expression de la sortie du banc de filtre est donnée par [20,21]:

$$\hat{X}(z) = \frac{1}{2} \left( F_0(z)H_0(z) + F_1(z)H_1(z) \right) X(z) + \frac{1}{2} \left( F_0(z)H_0(-z) + F_1(z)H_1(-z) \right) X(-z)$$

Il y aura reconstruction sera si  $\hat{x}(n) = x(n)$  à un retard près, soit :

$$F_0(z)H_0(z) + F_1(z)H_1(z) = 2 z^{-k}$$

$$F_0(z)H_0(-z) + F_1(z)H_1(-z) = 0$$

La deuxième condition sera réalisée si :  $F_0(z)H_0(-z) = -F_1(z)H_1(-z) \Rightarrow \frac{F_0(z)}{F_1(z)} = -\frac{H_1(-z)}{H_0(-z)}$

Soit  $F_0(z) = -H_1(-z)$  et  $F_1(z) = H_0(-z)$  ou  $F_0(z) = H_1(-z)$  et  $F_1(z) = -H_0(-z)$

- Une première famille de filtres dite Q.M.F. (Quadrature Mirror Filter) permet d'éliminer le terme en  $X(-z)$

En imposant  $H_1(z) = H_0(-z)$  soit  $h_1(n) = (-1)^n h_0(n)$  soit  $H_1(e^{2\pi j f T_e}) = H_0(-e^{2\pi j f T_e}) = H_0(e^{2\pi j (f \pm \frac{f_e}{2}) T_e})$

On obtient :  $F_0(z) = H_0(z)$  soit  $f_0(n) = h_0(n)$   
 $F_1(z) = -H_0(-z)$  soit  $f_1(n) = -(-1)^n h_0(n)$

- La deuxième famille de filtres dits C.Q.F. (Conjugate Quadrature Filter) permettent une reconstruction rigoureusement exacte du signal d'entrée contrairement aux filtres Q.M.F. mais ils possèdent une complexité calculatoire plus importante.

Cette fois-ci on veut obtenir  $H_1(z)$  tel que les relations d'orthogonalité suivantes soient vérifiées [30]:

$$H_0(z)H_0(z^{-1}) + H_0(-z)H_0(-z^{-1}) = 2$$

$$H_1(z)H_1(z^{-1}) + H_1(-z)H_1(-z^{-1}) = 2$$

$$H_0(z)H_1(z^{-1}) + H_0(-z)H_1(-z^{-1}) = 0$$

$\Rightarrow H_1(z) = \pm H_0(-z^{-1})z^{-(L-1)}$  soit  $h_1(n) = \mp (-1)^n h_0(L-1-n)$  et ce pour un nombre de coefficients L pair.

$\Rightarrow H_1(e^{2\pi j f T_e}) = \pm H_0(-e^{-2\pi j f T_e}) e^{-2\pi j f T_e(L-1)} = \pm H_0(e^{2\pi j (\frac{f_e}{2} - f) T_e}) e^{-2\pi j f T_e(L-1)} \Rightarrow |H_1(f)| = |H_0(\frac{f_e}{2} - f)|$

On obtient :  $F_0(z) = H_1(-z) = H_0(z^{-1})z^{-(L-1)}$  soit  $f_0(n) = h_0(L-1-n)$   
 $F_1(z) = -H_0(-z)$  soit  $f_1(n) = -(-1)^n h_0(n)$

En passant de la TZ vers la TF ( $z = e^{2\pi j f T_e}$ ), on aboutit aux relations suivantes :

- Symétrie en puissance  $|H_0(f)|^2 + |H_0(f \pm f_e/2)|^2 = 2$  et  $|H_1(f)|^2 + |H_1(f \pm f_e/2)|^2 = 2$
- Complémentarité en puissance  $|H_0(f)|^2 + |H_1(f)|^2 = 2$
- Orthogonalité  $H_0(f)H_1^*(f) + H_0(f \pm \frac{f_e}{2})H_1^*(f \pm \frac{f_e}{2}) = 0$

Remarque : Pour un gain de temps, on peut appliquer la décomposition polyphases aux filtres de synthèse et d'analyse.

**Filtres QMF** :  $H_1(z) = H_0(-z)$ ,  $F_0(z) = H_0(z)$ ,  $F_1(z) = -H_0(-z)$

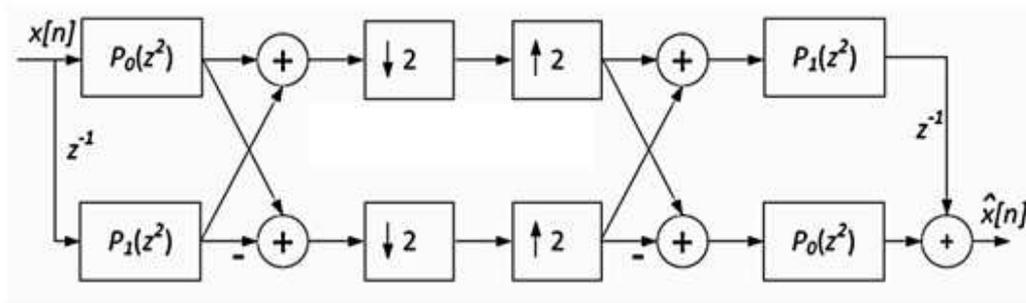
$$H_0(z) = P_0(z^2) + z^{-1}P_1(z^2)$$

$$H_1(z) = P_0(z^2) - z^{-1}P_1(z^2)$$

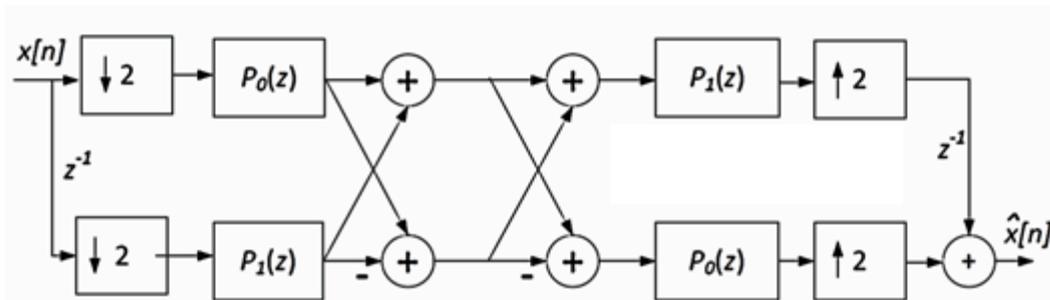
$$F_0(z) = P_0(z^2) + z^{-1}P_1(z^2)$$

$$F_1(z) = -[P_0(z^2) - z^{-1}P_1(z^2)]$$

Soit donc la structure ci-dessous



Après application des identité nobles:



#### 4. Applications de traitement multi-cadences [ 21]

Conception de déphaseurs, interfaçage de systèmes numériques à taux différents, implémentation de filtres LP à bande étroite, implémentation de bancs de filtres numériques, codage sous-bande de signaux de parole, filtres miroirs en quadrature, transmultiplexeurs, sur-échantillonnage dans les CA/N et CN/A .

Série n°5

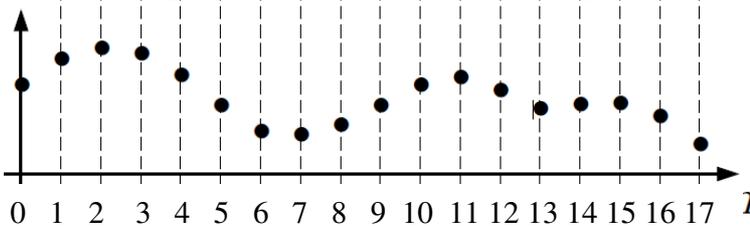
1. Soit le signal  $x(n) = n U(n)$ , donner et tracer le signal résultant si :

- On le décime par 2, par 3 et par 4
  - On l'interpole de 2, de 4
  - On le décime par 2 puis qu'on l'interpole par 2 et si on inverse les 2 opérations
  - On le décime par 3 et on l'interpole de 2 et si on inverse les 2 opérations
- Vérifier que la décimation n'est pas invariante à la translation contrairement à l'interpolation;

**Solution**

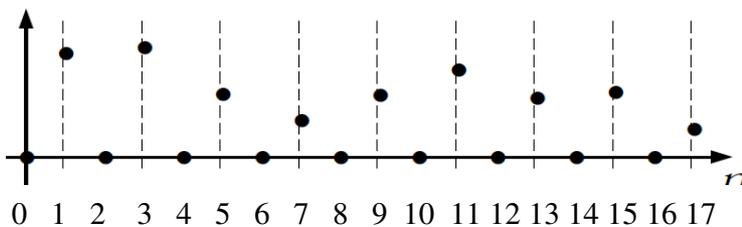
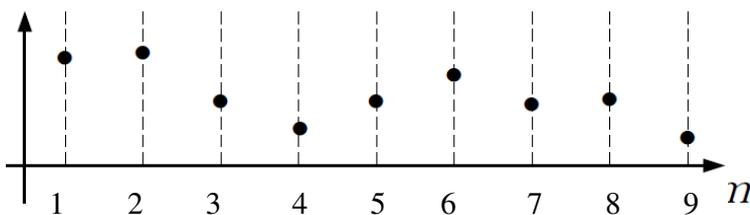
0,2,4,6,8,...      0,3,6,9,12,...      0,4,8,12,16,...  
 0,0,1,0,2,0,3,0,4,0,... ou 0,0,1,1,2,2,3,3,4,4,... ou 0,0,5,1,1.5,2,2.5,3,3.5,4,...  
 0,2,4,6,8,... puis 0,0,2,0,4,0,6,0,8,8,... En inversant 0,0,1,0,2,0,3,0,4,0,... puis 0,1,2,3,4,...  
 0,3,6,9,12,... puis 0,0,3,0,6,0,9,0,12,0,... En inversant  
 0,0,1,0,2,0,3,0,4,0,5,0,6,0,7,0,8,0,9,0,10,0,11,0 12,0,... puis 0,0, 3,0,6,0,9,0,12,..  
 Non Invariance de la décimation 0,2,4,6,8,... Ou 1,3,5,7,9,...

2. Soit le signal suivant, on veut le décimer par 2 puis l'interpoler par 2 également



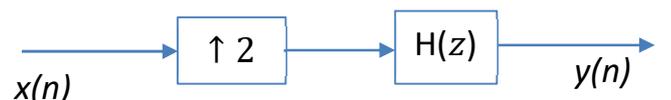
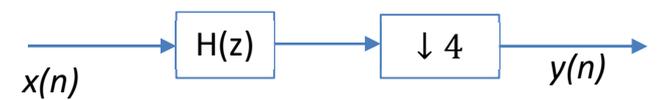
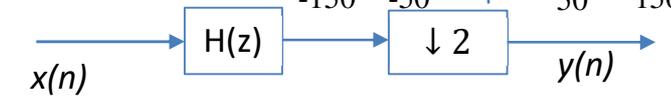
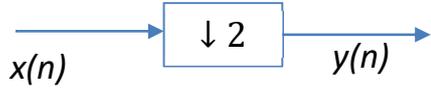
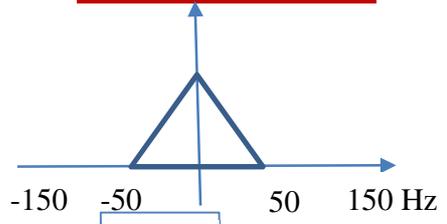
- Donner le tracé du signal décimé puis interpolé
- On considère que la fréquence du signal original est 10kHz, donner la fréquence à la sortie du décimateur puis celle de l'interpolateur.

**Solution**



Après décimation  $f_e = 5kHz$       Après interpolation  $f_e = 10kHz$

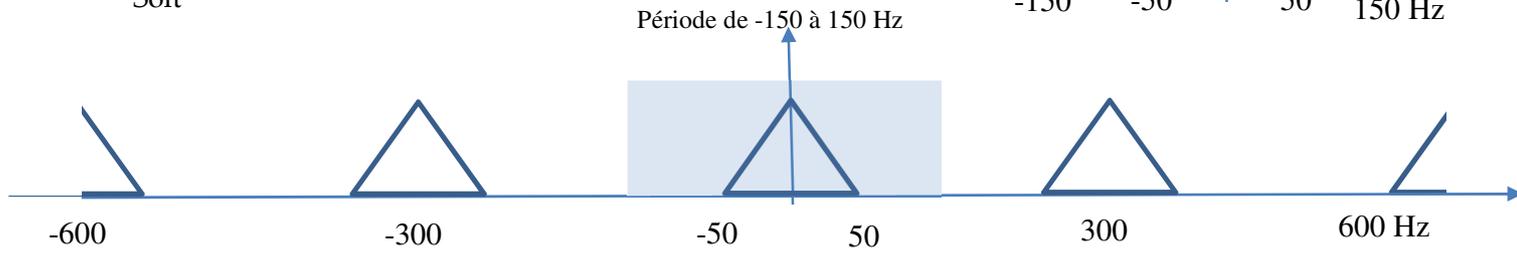
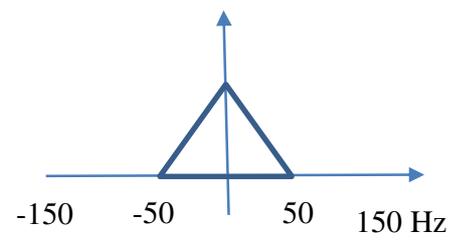
3. On considère le signal  $x(n)$  dont la TFTD est donnée comme ci-contre ,
- Tracer la TFTD de la sortie correspondante pour les 8 cas :
  - Donner la fréquence de coupure du filtre passe-bas  $H(z)$  pour chaque cas
  - Préciser le rôle de  $H(z)$  dans chaque cas



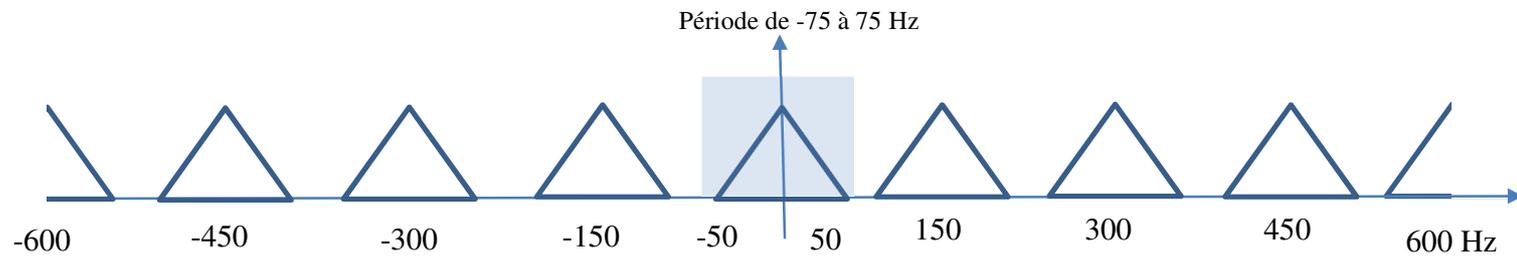
**Solution**

Le signal original est périodique de période 300Hz

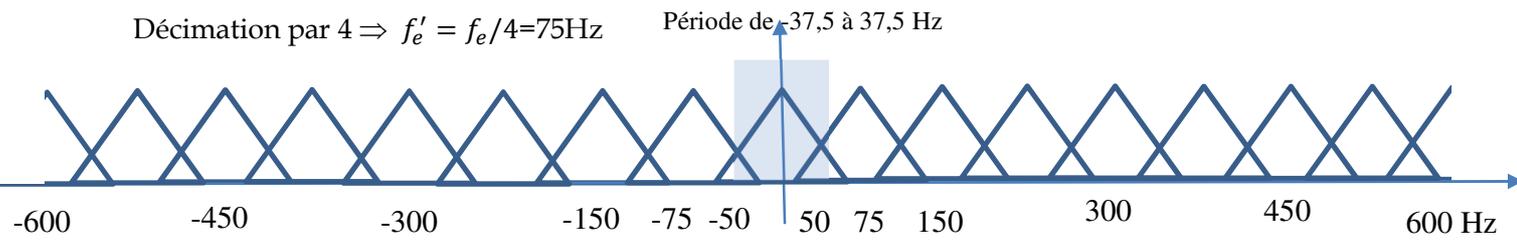
Soit



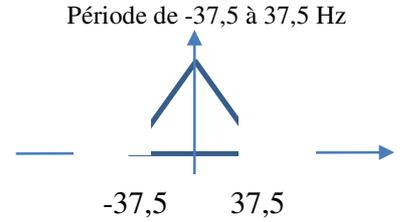
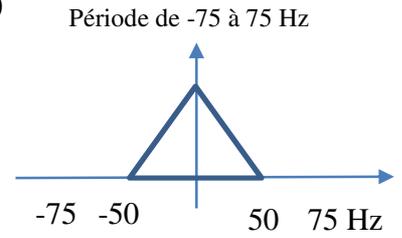
Décimation par 2  $\Rightarrow f'_e = f_e/2=150\text{Hz}$



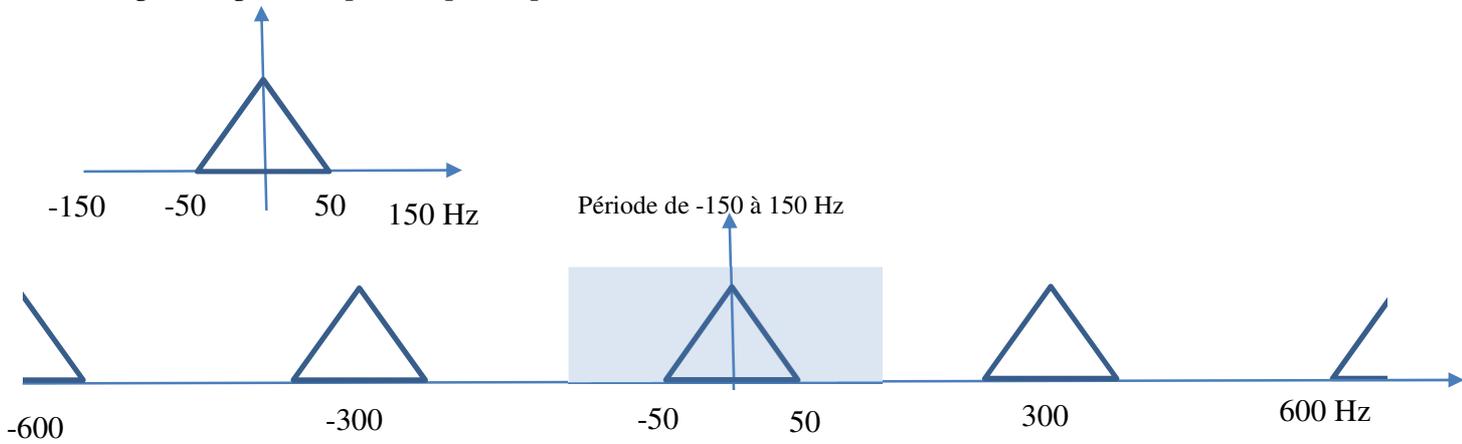
Décimation par 4  $\Rightarrow f'_e = f_e/4=75\text{Hz}$



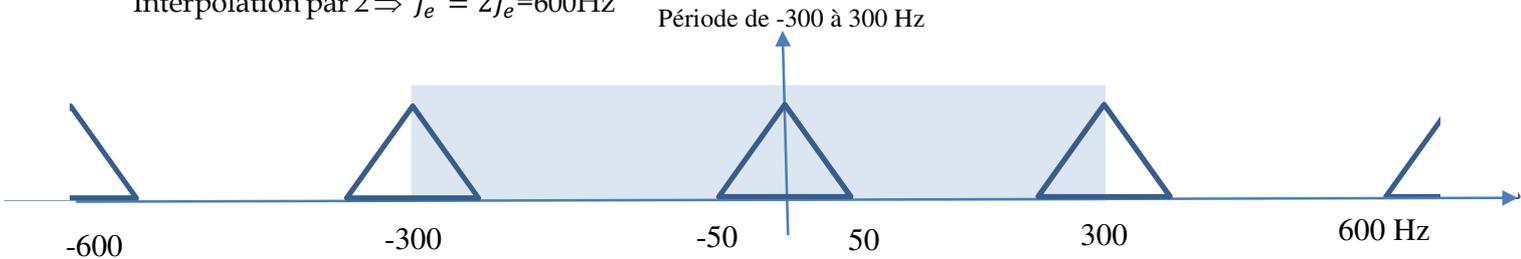
C'est pour cette raison qu'on place un filtre passe-bas à  $f'_e/2 = f_e/(2M)$



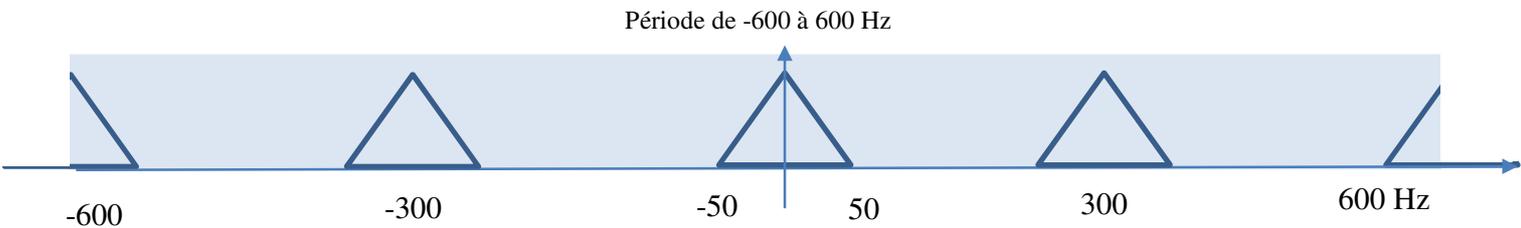
Le signal original est périodique de période 300Hz



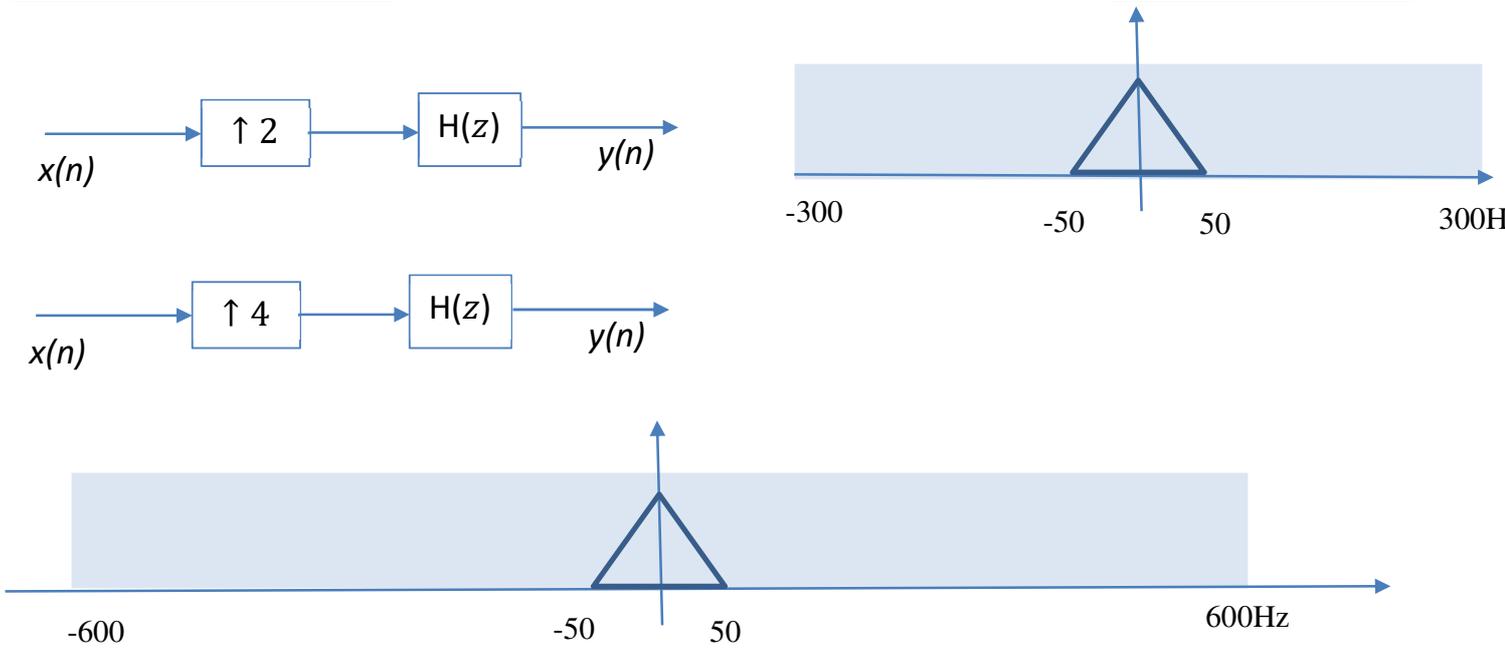
Interpolation par 2  $\Rightarrow f'_e = 2f_e = 600\text{Hz}$



Interpolation par 4  $\Rightarrow f'_e = 4f_e = 1200\text{Hz}$



Apparition de spectres miroirs  $\Rightarrow$  Filtre passe-bas à  $f_e/2$



$$f_c = \frac{300}{2.2} = 75, f_c = \frac{300}{2.4} = 37,5$$

$$f_c = \frac{300}{2} = 150, f_c = \frac{300}{2} = 150$$

Décimation: Anti-repliement

Interpolation: Anti-miroirs (Voir cours)

4. Donner l'expression des filtres polyphases pour les 4 cas de l'exercice précédent ainsi que la décomposition polyphases initiale et finale pour chaque cas.

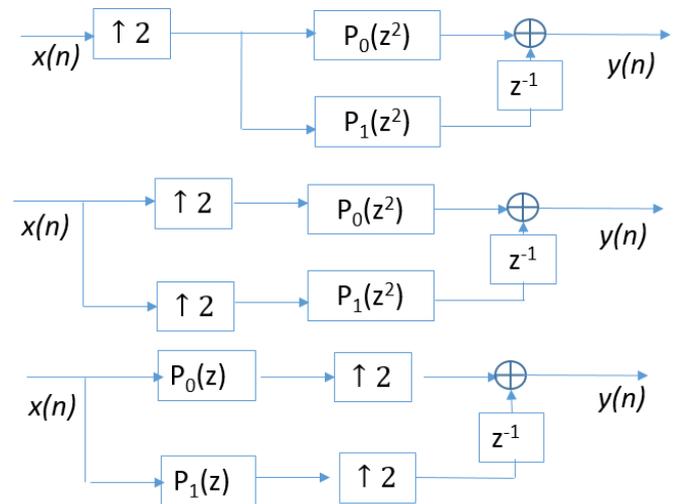
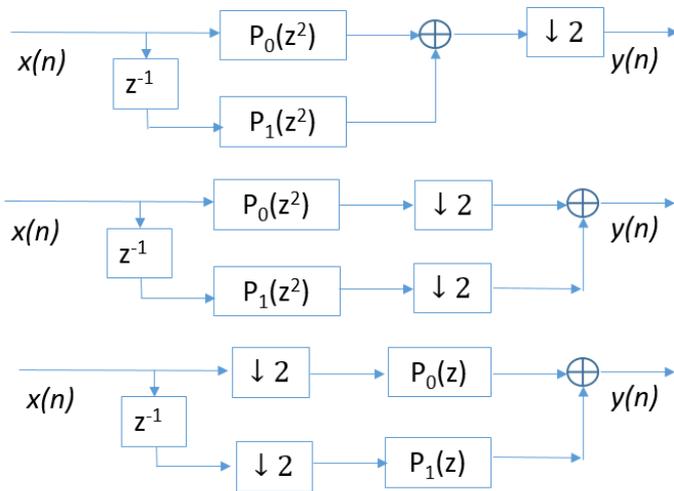
### Solution

$$- M=2 \Rightarrow H(z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n} = \sum_{n=-\infty}^{\infty} h(2n)z^{-2n} + z^{-1} \sum_{n=-\infty}^{\infty} h(2n+1)z^{-2n}$$

$$\text{On pose } P_0(z) = \sum_{n=-\infty}^{\infty} h(2n)z^{-n} \quad P_1(z) = \sum_{n=-\infty}^{\infty} h(2n+1)z^{-n} \Rightarrow H(z) = P_0(z^2) + z^{-1}P_1(z^2)$$

Structure polyphases pour une Décimation par 2

Structure polyphases pour une Interpolation par 2

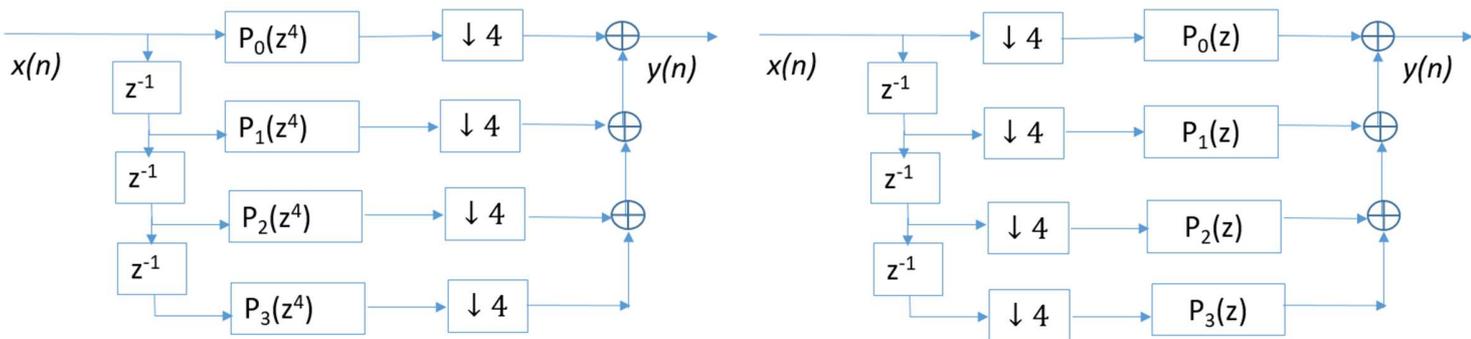


- M=4

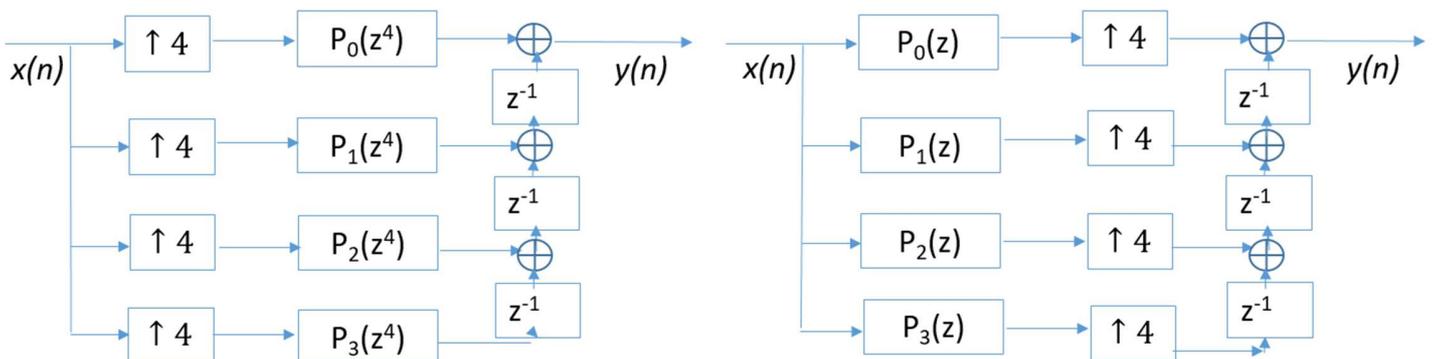
$$H(z) = \sum_{n=-\infty}^{\infty} h(4n)z^{-4n} + \sum_{n=-\infty}^{\infty} h(4n+1)z^{-4n-1} + \sum_{n=-\infty}^{\infty} h(4n+2)z^{-4n-2} + \sum_{n=-\infty}^{\infty} h(4n+3)z^{-4n-3}$$

On pose  $P_0(z) = \sum_{n=-\infty}^{\infty} h(4n)z^{-n}$      $P_1(z) = \sum_{n=-\infty}^{\infty} h(4n+1)z^{-n}$      $P_2(z) = \sum_{n=-\infty}^{\infty} h(4n+2)z^{-n}$   
 $P_3(z) = \sum_{n=-\infty}^{\infty} h(4n+3)z^{-n} \Rightarrow H(z) = P_0(z^4) + z^{-1}P_1(z^4) + z^{-2}P_2(z^4) + z^{-3}P_3(z^4)$

- Décimation par 4 (décomposition polyphases initiale et finale)



- Interpolation de 4 (décomposition polyphases initiale et finale)

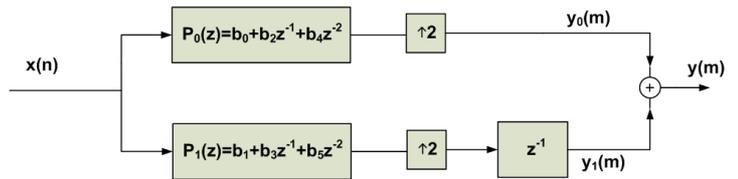


5. Soit le filtre  $H(z) = 1 + 0.5 z^{-1} - 0.5 z^{-2} + 0.9 z^{-3} + 0.8 z^{-4} + 1.2 z^{-5}$

- Donner l'expression numérique des filtres polyphases pour une décimation de 2
- Donner l'expression numérique des filtres polyphases pour une interpolation de 2
- Donner l'expression numérique des filtres polyphases pour une décimation de 3
- Donner l'expression numérique des filtres polyphases pour une interpolation de 3

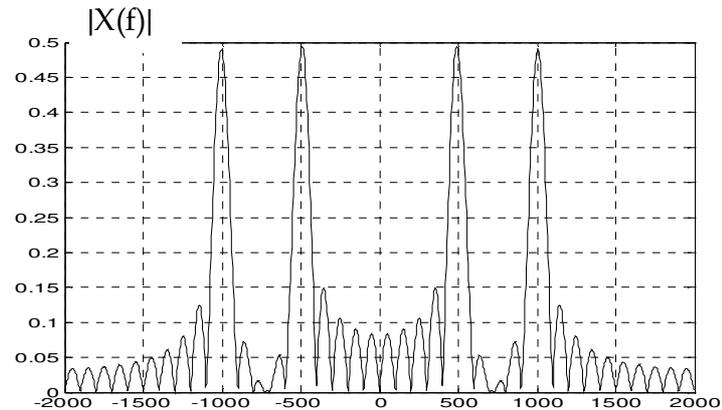
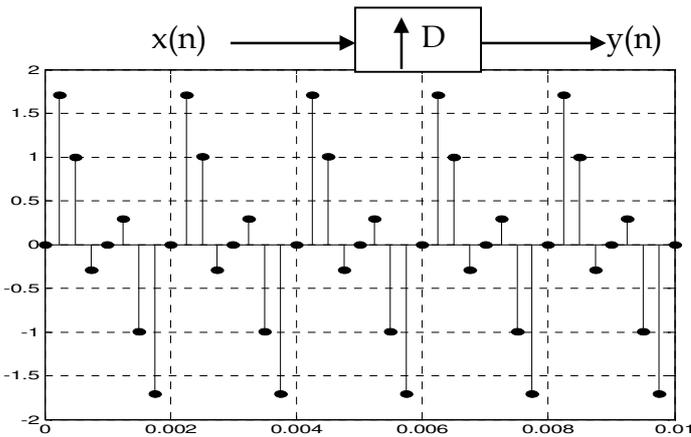
**Solution**

-  $H(z) = P_0(z^2) + z^{-1}P_1(z^2)$   
 $P_0(z) = \sum_{n=-\infty}^{\infty} h(2n)z^{-n} = 1 - 0.5 z^{-1} + 0.8 z^{-2}$   
 $P_1(z) = \sum_{n=-\infty}^{\infty} h(2n + 1)z^{-n} = 0.5 + 0.9 z^{-1} + 1.2 z^{-2}$



-  $H(z) = P_0(z^3) + z^{-1}P_1(z^3) + z^{-2}P_2(z^3)$   
 $P_0(z) = \sum_{n=-\infty}^{\infty} h(3n)z^{-n} = 1 + 0.9 z^{-1}$   
 $P_1(z) = \sum_{n=-\infty}^{\infty} h(3n + 1)z^{-n} = 0.5 + 0.8 z^{-1}$   
 $P_2(z) = \sum_{n=-\infty}^{\infty} h(3n + 2)z^{-n} = -0.5 + 1.2 z^{-1}$

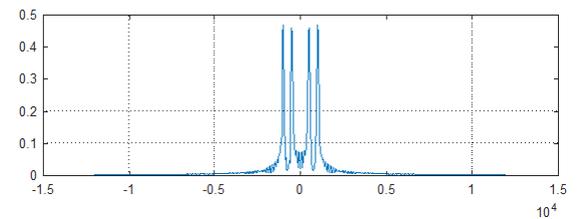
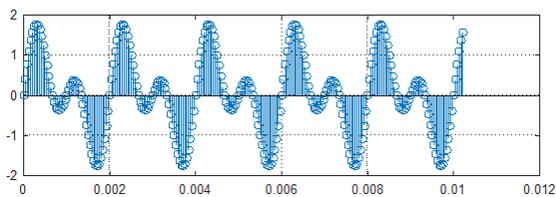
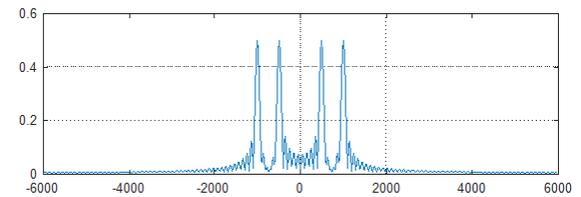
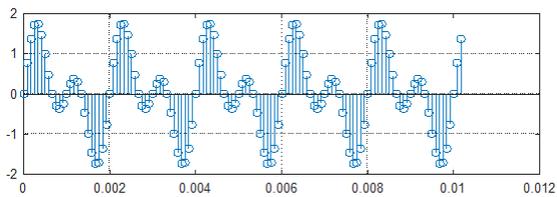
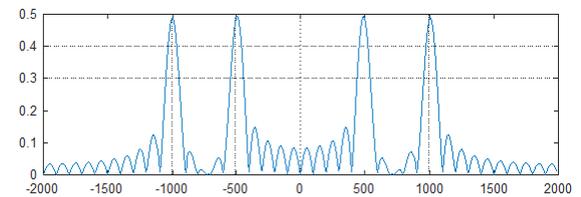
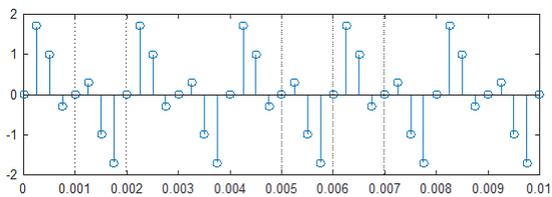
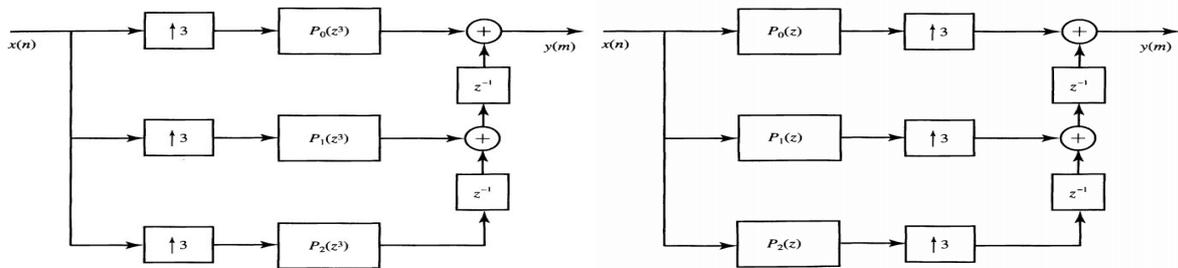
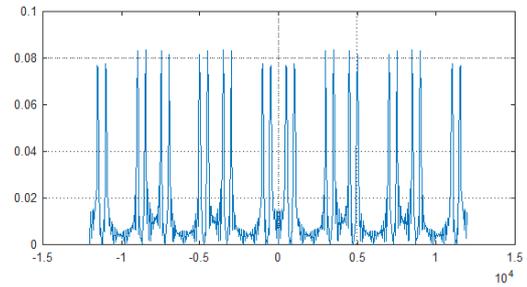
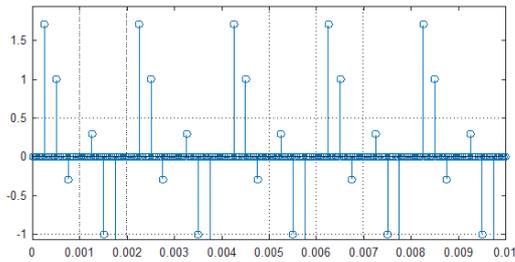
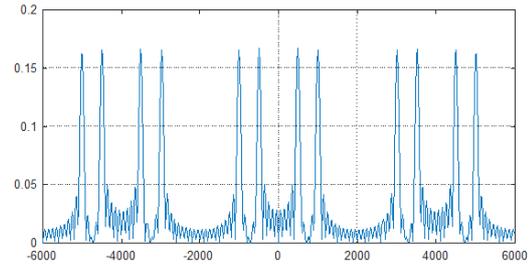
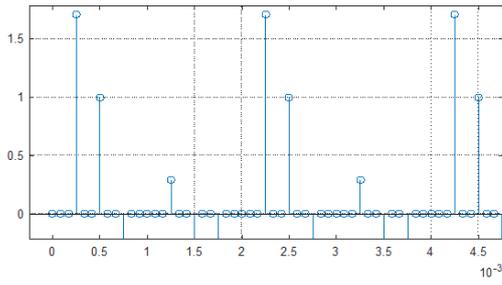
6. Avant de convertir un signal numérique x(n) (ci-dessous signal et sa TFTD), en signal analogique, on décide de l'interpoler par D.



- Quelle est l'intérêt de l'interpolation dans ce cas ?
- Déterminer l'expression du signal x(n).
- Tracer le signal interpolé pour D=3 et sa TFTD.
- On considère que D=6, tracer à nouveau le signal interpolé et sa TFTD.
- Donner la décomposition polyphases initiale et finale pour D=3
- Tracer le signal et sa TFTD dans le cas où l'on fait suivre l'interpolation d'un filtre passe-bas à  $f_c/2$

**Solution**

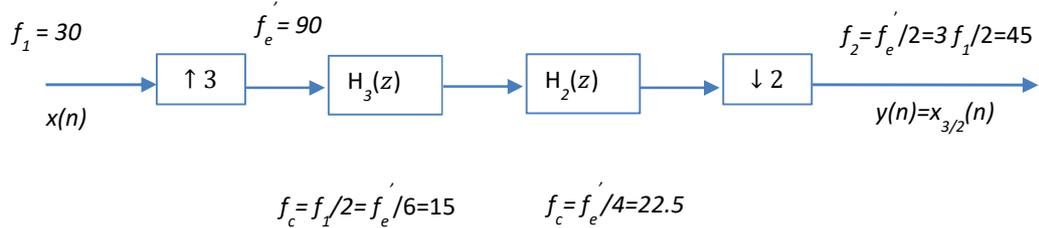
- Gagner en précision avant la conversion numérique analogique
- $h(n) = \sin(2\pi \cdot 500 \cdot n) + \sin(2\pi \cdot 1000 \cdot n) \quad f_c = 4\text{kHz}$
- $D=3 \Rightarrow$  Insérer 2 zéros entre 2 échantillons consécutifs. TFTD sur 3 périodes de -6kHz à 6kHz
- $D=6 \Rightarrow$  Insérer 5 zéros entre 2 échantillons consécutifs. TFTD sur 6 périodes de -12kHz à 12kHz



- Un signal audio est enregistré à été transmis à une fréquence de 30 kHz, à la réception avant de le convertir en analogique, on opère un changement de fréquence telle que la nouvelle fréquence soit 45Hz.

- Donner le schéma général de cette opération en employant la décimation et l'interpolation
- Donner la fréquence de coupure de chaque filtre passe-bas, lequel faudra-t-il garder
- Donner la décomposition polyphases finale

**Solution :**



8. Considérer une porte de largeur 5, vérifier les propriétés liant les fonctions de transfert pour la décimation et l'interpolation par 2

Décimation par M  $\Rightarrow Y(z) = \frac{1}{M} \sum_{k=0}^{M-1} X\left(W_M^k z^{\frac{1}{M}}\right)$  avec  $W_M = e^{-\frac{2\pi j}{M}}$

Interpolation par M  $\Rightarrow Y(z) = X(z^M)$

**Solution**

$x(n) = \{1, 1, \underline{1}, 1, 1\} \Rightarrow$  Décimation par 2  $\Rightarrow y(n) = x(2n) = \{1, \underline{1}, 1\}$ ,

$x(n) = \{1, 1, \underline{1}, 1, 1\} \Rightarrow$  Interpolation par 2  $\Rightarrow y(n) = x\left(\frac{n}{2}\right) = \{1, 0, 1, 0, \underline{1}, 0, 1, 0, 1\}$

$X(z) = z^2 + z^1 + 1 + z^{-1} + z^{-2} \Rightarrow$  Décimation par 2  $\Rightarrow Y(z) = \frac{1}{2} \left( X\left(z^{\frac{1}{2}}\right) + X\left(-z^{\frac{1}{2}}\right) \right)$

$Y(z) = \frac{1}{2} \left[ (z^1 + z^{1/2} + 1 + z^{-1/2} + z^{-1}) + (z^1 - z^{1/2} + 1 - z^{-1/2} + z^{-1}) \right] = z^1 + 1 + z^{-1}$

$X(z) = z^2 + z^1 + 1 + z^{-1} + z^{-2} \Rightarrow$  Interpolation par 2  $\Rightarrow Y(z) = X(z^2) = z^4 + z^2 + 1 + z^{-2} + z^{-4}$

9. Vérifier pour les filtres CQF que si  $H_1(z) = H_0(-z^{-1})$  alors  $H_1(f) = H_0\left(\frac{f_e}{2} - f\right)$

Vérifier pour les filtres QMF que si  $H_1(z) = H_0(-z)$ , soit  $H_1(f) = H_0\left(f - \frac{f_e}{2}\right)$

**Solution**

-  $H_0(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + \dots \Rightarrow H_1(z) = b_0 - b_1 z^1 + b_2 z^2 - b_3 z^3 + \dots$

$H_0(f) = b_0 + b_1 e^{-2\pi j f T_e} + b_2 e^{-4\pi j f T_e} + b_3 e^{-6\pi j f T_e} + \dots$

$H_1(f) = b_0 - b_1 e^{2\pi j f T_e} + b_2 e^{4\pi j f T_e} - b_3 e^{6\pi j f T_e} + \dots$

$H_0\left(\frac{f_e}{2} - f\right) = b_0 + b_1 e^{-2\pi j \left(\frac{f_e}{2} - f\right) T_e} + b_2 e^{-4\pi j \left(\frac{f_e}{2} - f\right) T_e} + b_3 e^{-6\pi j \left(\frac{f_e}{2} - f\right) T_e} + \dots = H_1(f)$

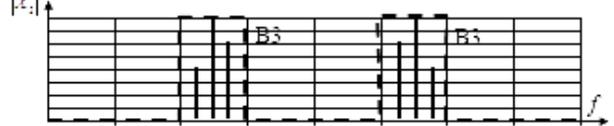
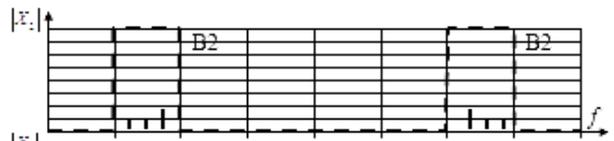
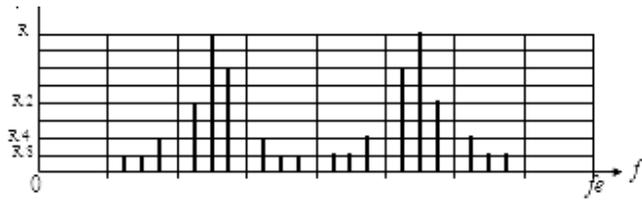
$$- H_0(z) = b_0 + b_1z^{-1} + b_2z^{-2} + b_3z^{-3} + \dots \Rightarrow H_1(z) = b_0 - b_1z^{-1} + b_2z^{-2} - b_3z^{-3} + \dots$$

$$H_0(f) = b_0 + b_1e^{-2\pi jfT_e} + b_2e^{-4\pi jfT_e} + b_3e^{-6\pi jfT_e} + \dots$$

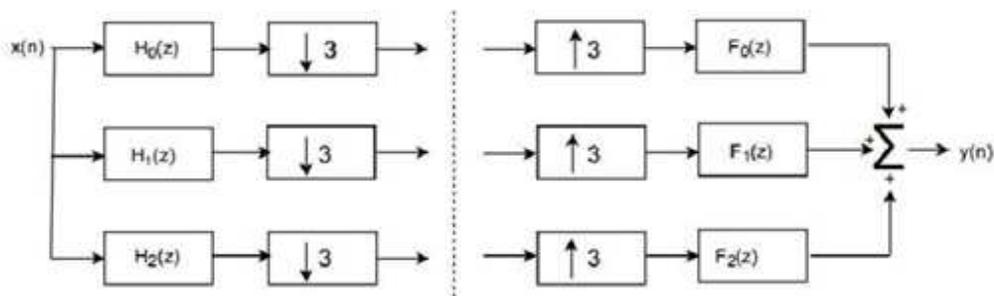
$$H_1(f) = b_0 - b_1e^{-2\pi jfT_e} + b_2e^{-4\pi jfT_e} - b_3e^{-6\pi jfT_e} + \dots$$

$$H_0\left(f - \frac{f_e}{2}\right) = b_0 + b_1e^{-2\pi j\left(f - \frac{f_e}{2}\right)T_e} + b_2e^{-4\pi j\left(f - \frac{f_e}{2}\right)T_e} + b_3e^{-6\pi j\left(f - \frac{f_e}{2}\right)T_e} + \dots = H_1(f)$$

10. Appliquer graphiquement un banc de 4 filtres du signal dont la Tf est la suivante :



11. On souhaite réaliser un banc de filtres pour M=3, donner la structure d'analyse et reconstruction correspondante.



**Exercices supplémentaires**

1. On dispose d'un signal audio échantillonné à  $f_1=48$  kHz que l'on souhaite convertir au standard CD de

$$f_2=44,1 \text{ kHz par décimation et interpolation. Solution } \frac{f_1}{f_2} = \frac{2^5 \times 5}{7^2 \times 3}$$

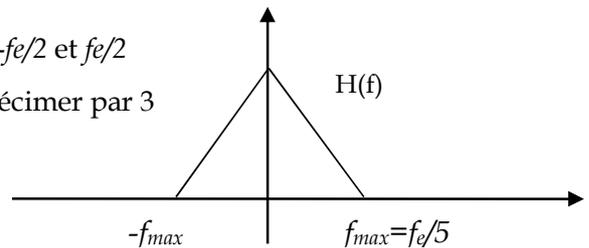
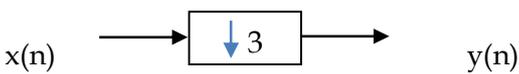
La conversion peut alors être réalisée en six étapes d'interpolation et de décimation :

- interpolation par 7 de 48 000 à 336 000 ;
- décimation par 10 de 336 000 à 33 600 ;

- interpolation par 7 de 33 600 à 235 200 ;
- décimation par 4 de 235 200 à 58 800 ;
- interpolation par 3 de 58 800 à 176 400 ;
- décimation par 4 de 176 400 à 44 100.

2. Soit le signal suivant :  $h(n) = \sin(2\pi \cdot 150 \cdot n) + \sin(2\pi \cdot 350 \cdot n)$  et le module de la TFD dont les tracés respectifs sont donnés ci-dessus :
- Déterminer  $T_e$ .
  - Quel est l'intérêt de l'interpolation ?
  - Tracer le signal (les 20 premières valeurs) et la TF obtenue pour une interpolation de 2 puis de 4 (sans filtrage passe-bas)
  - Pourquoi emploie-t-on un filtre passe-bas lors l'interpolation, où le place-t-on ?
  - Quel est le type du filtre de Tchebeychev employé dans ce cas et pourquoi ?
  - Donner la décomposition polyphases finale pour une interpolation de 2 en donnant l'expression des filtres polyphases

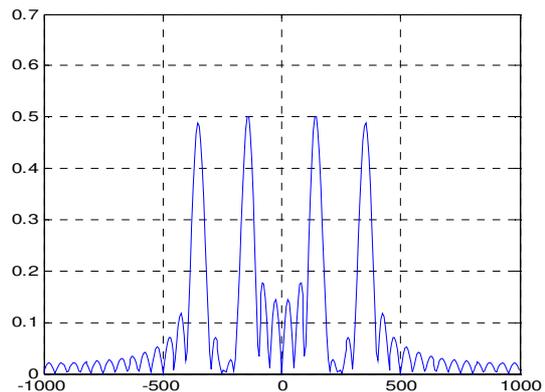
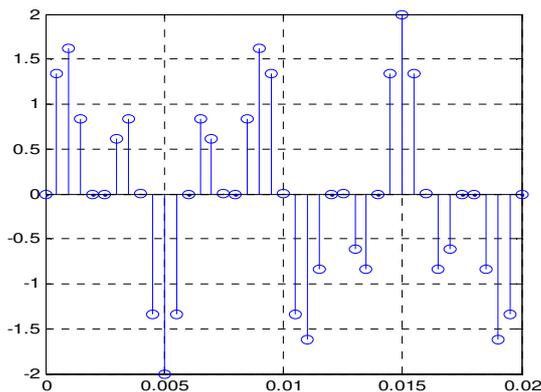
3. Afin de transmettre un signal  $x(n)$  (dont la TTFD entre  $-f_e/2$  et  $f_e/2$  est donné ci-contre) plus rapidement, on décide de le décimer par 3



- Tracer la TTFD obtenue après décimation ( $f_e=30\text{kHz}$ )
- On veut placer un filtre anti-repliement, ou doit-on le placer ? (justifier), tracer la TTFD à nouveau
- Donner la décomposition polyphases initiale et finale en donnant l'expression de chaque filtre.

**Solution : Examen 18/19**

4. Soit le signal suivant :  $h(n) = \sin(2\pi \cdot 150 \cdot n) + \sin(2\pi \cdot 350 \cdot n)$  et le module de la TFD dont les tracés respectifs sont donnés ci-après :



- Tracer le signal et la TF obtenue pour une décimation de 2 puis de 4 (avec filtrage passe-bas)
- Donner la décomposition polyphases finale pour une décimation de 4 en donnant l'expression des filtres polyphases
- Quel est l'intérêt de la décomposition polyphases
- Pourquoi emploie-t-on un filtre passe-bas lors de la décimation
- **Solution : Examen 17/18**

TP n°5 (Sous Matlab)

1. Interpolation et Décimation

```
clc;clear all; close all;
Te=0.0005; Fe=1/Te; t = 0:Te:0.1; N=length(t); NF=1024;
x = sin(2*pi*100*t) + sin(2*pi*300*t);

figure; subplot(3,2,1); stem(t,x); grid
y1 = x(1:2:N); t1 = t(1:2:N); subplot(3,2,3); stem(t1,y1); grid
y2 = x(1:6:N); t2 = t(1:6:N); subplot(3,2,5); stem(t2,y2); grid
X = fft(x,NF);X=fftshift(X); axef=Fe*(-1/2:1/NF:1/2-1/NF);
subplot(3,2,2); plot(axef,abs(X/N)); grid
Y1 = fft(y1,NF);Y1=fftshift(Y1);axef=Fe/2*(-1/2:1/NF:1/2-1/NF);
subplot(3,2,4); plot(axef,abs(2*Y1/N)); grid
Y2 = fft(y2,NF); Y2=fftshift(Y2);axef=Fe/6*(-1/2:1/NF:1/2-1/NF);
subplot(3,2,6); plot(axef,abs(6*Y2/N)); grid
```

```
figure; subplot(3,2,1); stem(t,x); grid
y1 = decimate(x,2); t1 = t(1:2:N); subplot(3,2,3); stem(t1,y1); grid
y2 = decimate(x,6); t2 = t(1:6:N); subplot(3,2,5); stem(t2,y2); grid
X = fft(x,NF);X=fftshift(X); axef=Fe*(-1/2:1/NF:1/2-1/NF);
subplot(3,2,2); plot(axef,abs(X/N)); grid
Y1 = fft(y1,NF);Y1=fftshift(Y1);axef=Fe/2*(-1/2:1/NF:1/2-1/NF);
subplot(3,2,4); plot(axef,abs(2*Y1/N)); grid
Y2 = fft(y2,NF); Y2=fftshift(Y2);axef=Fe/6*(-1/2:1/NF:1/2-1/NF);
subplot(3,2,6); plot(axef,abs(6*Y2/N)); grid
```

1. Etablir le lien entre période du signal original et celui sous-échantillonné, faire de même pour les fréquences.
2. Expliquer la différence entre `decimate(x, 6)` ; entre `x(1:6:N)` ;

Rajouter les lignes suivantes

```
figure;subplot(3,2,1); stem(t,x); grid
L=2; t1 = 0:Te/L:0.1;N1=length(t1);x1= zeros(N1, 1); x1(1:L:N*L) = x;
subplot(3,2,3); stem(t1,x1); grid
L=4 ; t2 = 0:Te/L:0.1;N2=length(t2);x2= zeros(N2, 1); x2(1:L:N*L)= x;
subplot(3,2,5); stem(t2,x2); grid
X = fft(x,NF);X=fftshift(X); axef=Fe*(-1/2:1/NF:1/2-1/NF);
subplot(3,2,2); plot(axef,abs(X/N)); grid
L=2;Y1 = fft(x1,NF);Y1=fftshift(Y1);axef=L*Fe*(-1/2:1/NF:1/2-1/NF);
subplot(3,2,4); plot(axef,abs(Y1/N1)); grid
L=4 ;Y2 = fft(x2,NF); Y2=fftshift(Y2);axef=L*Fe*(-1/2:1/NF:1/2-1/NF);
subplot(3,2,6); plot(axef,abs(Y2/N2)); grid
```

3. Commenter les TFD obtenues
4. Rajouter les lignes suivantes et comparer (justifier les différences observées sur la TFD

```
figure;subplot(3,2,1); stem(t,x); grid
L=2; t1 = 0:Te/L:0.1+(L-1)*Te/L;N1=length(t1);x1=interp(x, L);
subplot(3,2,3); stem(t1,x1); grid
L=4; t2 = 0:Te/L:0.1+(L-1)*Te/L;N2=length(t2);x2=interp(x, L);
subplot(3,2,5); stem(t2,x2); grid
X = fft(x,NF);X=fftshift(X); axef=Fe*(-1/2:1/NF:1/2-1/NF);
subplot(3,2,2); plot(axef,abs(X/N)); grid
L=2;Y1 = fft(x1,NF);Y1=fftshift(Y1);axef=L*Fe*(-1/2:1/NF:1/2-1/NF);
subplot(3,2,4); plot(axef,abs(Y1/N1)); grid
```

```
L=4 ; Y2 = fft(x2,NF); Y2=fftshift(Y2); axef=L*Fe*(-1/2:1/NF:1/2-1/NF);  
subplot(3,2,6); plot(axef,abs(Y2/N2)); grid
```

5. Pour expérimenter avec le changement de  $f_e$  et ses conséquences : Téléchargez le son Vous avez du courrier en attente.wav.
  - Écoutez et Observez son spectre.
  - Décimez-le (sans filtre passe-bas), écoutez et observez le spectre du résultat.
  - Interpolez-le de deux manières différentes (zéro et échelon). Écoutez et observez son spectre.

## 2. Décomposition polyphases

## 3. Bancs de filtres

## TP n°5 (Sous Python)

### 1. Interpolation et Décimation : Démo

```
# -*- coding: utf-8 -*-
import numpy as np; import matplotlib.pyplot as plt; import scipy.signal as sp;

N = 30; f0=100; f1=300; fe=2000; Te=1/fe; NF=1024;
t = np.arange(0.0, N*Te, Te); x = np.sin(2.0*np.pi*f0*t) + np.sin(2.0*np.pi*f1*t) ;
TFx = np.fft.fft(x, NF); TFx = np.fft.fftshift(TFx); freq = np.arange(-NF/2, NF/2)*fe/NF;
plt.figure(1)
plt.subplot(321); plt.stem(t, x); plt.subplot(322); plt.plot(freq, np.abs(TFx));
x1 = np.zeros(2*N); x1[:,2]=x[:,:] ; t1 = np.arange(0.0, N*Te, Te/2); TFx1 =
np.fft.fft(x1, NF);
TFx1 = np.fft.fftshift(TFx1); freq = np.arange(-NF/2, NF/2)*2*fe/NF;
plt.subplot(323); plt.stem(t1, x1); plt.subplot(324); plt.plot(freq, np.abs(TFx1));
x2 = np.zeros(4*N); x2[:,4]=x[:,:] ; t2 = np.arange(0.0, N*Te, Te/4); TFx2 = np.fft.fft(x2, NF);
TFx2 = np.fft.fftshift(TFx2); freq = np.arange(-NF/2, NF/2)*4*fe/NF;
plt.subplot(325); plt.stem(t2, x2); plt.subplot(326); plt.plot(freq, np.abs(TFx2));

plt.figure(2)
freq = np.arange(-NF/2, NF/2)*fe/NF;
plt.subplot(321); plt.stem(t, x); plt.subplot(322); plt.plot(freq, np.abs(TFx));
x1 = sp.resample(x, 2*N); t1 = np.arange(0.0, N*Te, Te/2); TFx1 = np.fft.fft(x1, NF);
TFx1 = np.fft.fftshift(TFx1); freq = np.arange(-NF/2, NF/2)*2*fe/NF;
plt.subplot(323); plt.stem(t1, x1); plt.subplot(324); plt.plot(freq, np.abs(TFx1));
x2 = sp.resample(x, 4*N); t2 = np.arange(0.0, N*Te, Te/4); TFx2 = np.fft.fft(x2, NF);
TFx2 = np.fft.fftshift(TFx2); freq = np.arange(-NF/2, NF/2)*4*fe/NF;
plt.subplot(325); plt.stem(t2, x2); plt.subplot(326); plt.plot(freq, np.abs(TFx2));

N = 100; f0=100; f1=300; fe=2000; Te=1/fe; NF=1024;
t = np.arange(0.0, N*Te, Te); x = np.sin(2.0*np.pi*f0*t) + np.sin(2.0*np.pi*f1*t) ;
TFx = np.fft.fft(x, NF); TFx = np.fft.fftshift(TFx); freq = np.arange(-NF/2, NF/2)*fe/NF;
plt.figure(3)
plt.subplot(321); plt.stem(t, x); plt.subplot(322); plt.plot(freq, np.abs(TFx));
x1 = x[:,2]; t1 = t[:,2]; TFx1 = np.fft.fft(x1, NF); TFx1 = np.fft.fftshift(TFx1);
freq = np.arange(-NF/2, NF/2)*fe/(2*NF);
plt.subplot(323); plt.stem(t1, x1); plt.subplot(324); plt.plot(freq, np.abs(TFx1));
x2 = x[:,6]; t2 = t[:,6]; TFx2 = np.fft.fft(x2, NF); TFx2 = np.fft.fftshift(TFx2);
freq = np.arange(-NF/2, NF/2)*fe/(6*NF);
plt.subplot(325); plt.stem(t2, x2); plt.subplot(326); plt.plot(freq, np.abs(TFx2));

plt.figure(4)
freq = np.arange(-NF/2, NF/2)*fe/NF;
plt.subplot(321); plt.stem(t, x); plt.subplot(322); plt.plot(freq, np.abs(TFx));
x1 = sp.decimate(x, 2); t1 = t[:,2]; TFx1 = np.fft.fft(x1, NF); TFx1 = np.fft.fftshift(TFx1);
freq = np.arange(-NF/2, NF/2)*fe/(2*NF);
plt.subplot(323); plt.stem(t1, x1); plt.subplot(324); plt.plot(freq, np.abs(TFx1));
x2 = sp.decimate(x, 6); t2 = t[:,6]; TFx2 = np.fft.fft(x2, NF); TFx2 = np.fft.fftshift(TFx2);
freq = np.arange(-NF/2, NF/2)*fe/(6*NF);
plt.subplot(325); plt.stem(t2, x2); plt.subplot(326); plt.plot(freq, np.abs(TFx2));
```

## 2. Interpolation et Décimation : A faire

# -\*- coding: utf-8 -\*-

```
import numpy as np; import matplotlib.pyplot as plt
#from zp_plot import zplane
import scipy.signal as sp;
```

```
from scipy.io import wavfile as wf; import winsound ;
fname = 'vousavezducourrierenattente.wav';
winsound.PlaySound(fname, winsound.SND_FILENAME)
fe, x = wf.read(fname);
Te=1/fe; N=len(x); t = np.arange(0,N)*Te;
```

```
# plt.figure(1);plt.subplot(211); plt.plot(t,x); plt.grid(True);
# plt.xlabel('temps'); plt.ylabel('Amp'); plt.title('Phrase');
# #Ecrire dans un fichier audio
# fnameb='Copie du signal.wav';
# Xb=X
# Xb=np.int8(Xb) ; wf.write(fnameb, fe, Xb);
# winsound.PlaySound(fnameb,winsound.SND_FILENAME) ;
```

#-----A faire-----

# 1

```
# Tester le sous échantillonnage avec et sans filtrage passe-bas
# Afficher le signal et sa TFD dans chaque cas et zoomer sur une meme zone d'une durée de 50 ms et commenter
# Ecouter le signa obtenu à chaque fois (wf.wite avec la nouvelle fe en divisant // par le facteur de décimation)
# Perçoit-on une déformation du son ? A quoi est-elle due ?
# Jusqu'à quel facteur de décimation peut-on aller sans déformation
# Que se passet-il si on garde la même fe (originale) à l'écoute?
```

# 2

```
# Tester le sur échantillonnage avec et sans filtrage passe-bas
# Afficher le signal et sa TFD dans chaque cas et zoomer sur une meme zone d'une durée de 50 ms ou 20ms et commenter
# Ecouter le signa obtenu à chaque fois (wf.wite avec la nouvelle fe en multipliant par facteur d'Interp)
# Perçoit-on une déformation du son ? pourquoi?
# Que se passet-il si vous garde la même fe à l'écoute?
```

# 3

```
# On veut passer à un signal dont la fréquence d'échantillonnage est 1.5*fe
# Proposer 2 solutions
# Laquelle vous permet de conserver le plus les valeurs originelles?
```

### 3. Décomposition polyphases

`scipy.signal.resample_poly`

### 4. Bancs de filtres : Démo

#[https://github.com/mshakeelt/8-Bands\\_Filter\\_Bank](https://github.com/mshakeelt/8-Bands_Filter_Bank)

```
import numpy as np
import scipy.signal as sp
import sounddevice as snd
import scipy.io.wavfile as wav
import matplotlib.pyplot as plt

#-----Lecture du signal audio-----#
fe,x = wav.read('vousavezducourrierenattente.wav')
snd.play(x, fe)

#-----Génération du banc de filtre-----#
N = 8
f1 = sp.remez(8*N, [0,0.0625, 0.0627, 0.5], [1,0], [1, 100])
f2 = sp.remez(8*N, [0, 0.0625, 0.0627, 0.125, 0.127, 0.5], [0, 1, 0], [100, 1, 100])
f3 = sp.remez(8*N, [0, 0.125, 0.127, 0.1875, 0.1877, 0.5], [0, 1, 0], [100, 1, 100])
f4 = sp.remez(8*N, [0, 0.1875, 0.1877, 0.25, 0.27, 0.5], [0, 1, 0], [100, 1, 100])
f5 = sp.remez(8*N, [0, 0.25, 0.27, 0.3125, 0.3127, 0.5], [0, 1, 0], [100, 1, 100])
f6 = sp.remez(8*N, [0, 0.3125, 0.3127, 0.375, 0.377, 0.5], [0, 1, 0], [100, 1, 100])
f7 = sp.remez(8*N, [0, 0.375, 0.377, 0.4375, 0.4377, 0.5], [0, 1, 0], [100, 1, 100])
f8 = sp.remez(8*N, [0, 0.4375, 0.4377, 0.5], [0, 1], [100, 1])

## Calcul des réponses en fréquence
w1,H1 = sp.freqz(f1,1,fs=fe); w2,H2 = sp.freqz(f2,1,fs=fe)
w3,H3 = sp.freqz(f3,1,fs=fe); w4,H4 = sp.freqz(f4, 1,fs=fe)
w5,H5 = sp.freqz(f5, 1,fs=fe); w6,H6 = sp.freqz(f6, 1,fs=fe)
w7,H7 = sp.freqz(f7, 1,fs=fe); w8,H8 = sp.freqz(f8,1,fs=fe)

## Visualisation des reponses impulsionnelles et fréquentielles
plt.figure(1)
plt.plot(f1, 'b', f2, 'g',f3, 'r',f4, 'c',f5, 'm',f6, 'y',f7, 'k',f8, 'b--')
plt.legend(('Lowpass', 'Bandpass 1', 'Bandpass 2', 'Bandpass 3', 'Bandpass 4', 'Bandpass 5', 'Bandpass 6',
'Highpass'), loc='upper right')
plt.title('Réponse impulsionnelle')

plt.figure(2)
plt.plot(w1,20*np.log10(np.abs(H1))); plt.plot(w2,20*np.log10(np.abs(H2)))
plt.plot(w3,20*np.log10(np.abs(H3))); plt.plot(w4,20*np.log10(np.abs(H4)))
plt.plot(w5,20*np.log10(np.abs(H5))); plt.plot(w6,20*np.log10(np.abs(H6)))
plt.plot(w7,20*np.log10(np.abs(H7)));plt.plot(w8,20*np.log10(np.abs(H8)))
plt.legend(('Lowpass', 'Bandpass 1', 'Bandpass 2', 'Bandpass 3', 'Bandpass 4', 'Bandpass 5', 'Bandpass 6',
'Highpass'), loc='upper right')
plt.xlabel('Fréquence');plt.ylabel('Amplitude en dB') ;plt.title('Réponses fréquentielles')
```

```

#-----Analyse par bancs de filtre-----#
## Filtrage par le banc de filtres
filtered1 = sp.lfilter(f1,1,x)
filtered2 = sp.lfilter(f2,1,x)
filtered3 = sp.lfilter(f3,1,x)
filtered4 = sp.lfilter(f4,1,x)
filtered5 = sp.lfilter(f5,1,x)
filtered6 = sp.lfilter(f6,1,x)
filtered7 = sp.lfilter(f7,1,x)
filtered8 = sp.lfilter(f8,1,x)

## Ecoute du son filtré par les 8 filtres
# a=input('appuyer sur une touche')
# snd.play(filtered1, fe); a=input('appuyer sur une touche')
# snd.play(filtered2, fe); a=input('appuyer sur une touche')
# snd.play(filtered3, fe); a=input('appuyer sur une touche')
# snd.play(filtered4, fe); a=input('appuyer sur une touche')
# snd.play(filtered5, fe); a=input('appuyer sur une touche')
# snd.play(filtered6, fe); a=input('appuyer sur une touche')
# snd.play(filtered7, fe); a=input('appuyer sur une touche')
# snd.play(filtered8, fe); a=input('appuyer sur une touche')

## Décimation
filteredds1 = filtered1[::N]
filteredds2 = filtered2[::N]
filteredds3 = filtered3[::N]
filteredds4 = filtered4[::N]
filteredds5 = filtered5[::N]
filteredds6 = filtered6[::N]
filteredds7 = filtered7[::N]
filteredds8 = filtered8[::N]

#-----Reconstruction par bancs de filtre-----#
##Interpolation
filteredus1 = np.zeros(len(x))
filteredus2 = np.zeros(len(x))
filteredus3 = np.zeros(len(x))
filteredus4 = np.zeros(len(x))
filteredus5 = np.zeros(len(x))
filteredus6 = np.zeros(len(x))
filteredus7 = np.zeros(len(x))
filteredus8 = np.zeros(len(x))

filteredus1[::N] = filteredds1
filteredus2[::N] = filteredds2
filteredus3[::N] = filteredds3
filteredus4[::N] = filteredds4
filteredus5[::N] = filteredds5
filteredus6[::N] = filteredds6
filteredus7[::N] = filteredds7

```

```
filteredus8[:,N] = filteredds8
```

```
# Filtrage par le banc de filtres
filteredyn1 = sp.lfilter(f1,1,filteredus1)
filteredyn2 = sp.lfilter(f2,1,filteredus2)
filteredyn3 = sp.lfilter(f3,1,filteredus3)
filteredyn4 = sp.lfilter(f4,1,filteredus4)
filteredyn5 = sp.lfilter(f5,1,filteredus5)
filteredyn6 = sp.lfilter(f6,1,filteredus6)
filteredyn7 = sp.lfilter(f7,1,filteredus7)
filteredyn8 = sp.lfilter(f8,1,filteredus8)
```

```
## Sommation des réponses
```

```
recons =
filteredyn1+filteredyn2+filteredyn3+filteredyn4+filteredyn5+filteredyn6+filteredyn7+filteredyn8
```

```
a=input('appuyer sur une touche')
snd.play(recons, fe)
```

```
plt.figure(3)
plt.plot(x)
plt.plot(recons)
plt.show()
```

#### 4. Bancs de filtres : A faire

```
#-----A faire-----#
# Visualiser les TFD des filtres dans l'échelle linéaire (pas en décibels)
# Visualiser la TFD du signal audio original ainsi que la TFD du Signal reconstruit
# Ne pas oublier de diviser les 2 TFD par le nombre total de points (TFD normalisée)
# Calculer l'erreur quadratique moyenne (somme des carrés des différences)
# Calculer l'énergie de signaux décimés
# Ne pas considérer lors de la reconstruction les bandes de fréquence les moins porteuses d'énergie
# Faire l'essai en enlevant une seule bande puis 2 puis 3, Jusqu'où peut-on aller?
# On veut faire de la compression. proposer une solution
# Rajouter du bruit dans une des bandes et reconstruire sans cette bande. Commenter?
```

## VI. Transformation en ondelettes discrètes (DWT) (En cours d'élaboration)

La plupart des signaux du monde réel ne sont pas stationnaires, et c'est justement dans l'évolution de leurs caractéristiques (statistiques, fréquentielles, temporelles, spatiales) que réside l'essentiel de l'information qu'ils contiennent. Les signaux vocaux et les images en sont des exemples courants [27]. Rappelons que l'analyse de Fourier permet une caractérisation globale du signal (on intègre de  $-\infty$  à  $+\infty$ ), on perd toute localisation temporelle ou spatiale, l'idéal est de faire appel à une transformation qui nous apporte l'information sur le contenu fréquentiel tout en préservant la localisation (temporelle ou spatiale) afin d'obtenir une représentation temps/fréquence ou espace/échelle du signal. C'est ainsi que deux théories ont été élaborées, la transformée de Fourier à fenêtre puis la transformée continue par ondelettes.

Les ondelettes, famille de fonctions déduites d'une même fonction, appelée ondelette mère, par opérations de translations et dilatations, ont trouvé, de par la puissance de leur théorie, des applications dans de nombreux domaines aussi variés que les mathématiques (analyse, probabilités, fractales), le traitement du signal (compression, astronomie, sismique), la physique (mécanique quantique, turbulence).

### 1. Dualité temps-fréquence et transformée de Fourier à court terme. Inconvénients.

La transformée de Fourier à court terme (TFCT) et ses dérivées (notamment le spectrogramme) sont les méthodes temps-fréquence les plus utilisées dans les applications pratiques. Ainsi, cette classe de méthodes représente la solution la plus répandue pour éliminer les limitations de la transformée de Fourier. L'idée de base est très simple et efficace : on décompose le signal en petits segments et on applique, sur chacune des sections, la transformée de Fourier en obtenant ainsi le spectre « local ». La totalité des spectres « locaux » indique alors comment le spectre varie au cours du temps.

$$TFCT(t, f) = \int_{-\infty}^{+\infty} x(\tau)h^*(\tau - t)e^{-j2\pi f\tau} d\tau$$

Où  $h(n)$  est une fenêtre de pondération (Rectangulaire, Bartlett, Hanning, Hamming, Gaussienne etc.)

Le spectrogramme est le module au carré de la TFCT

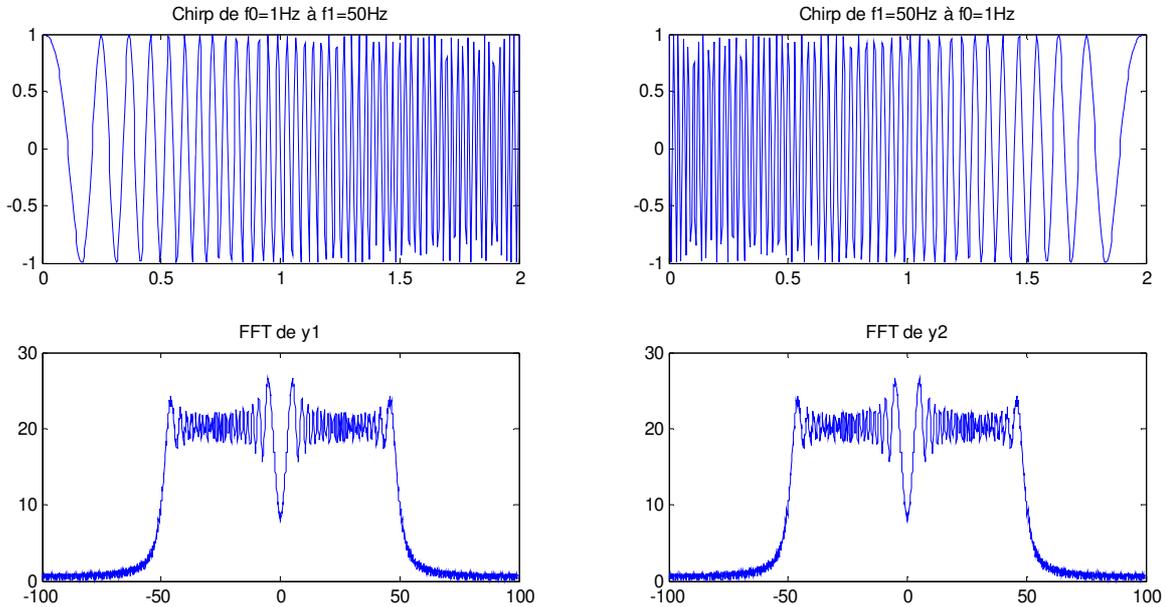
$$S_x(t, f) = \left| \int_{-\infty}^{+\infty} x(\tau)h^*(\tau - t)e^{-j2\pi f\tau} d\tau \right|^2$$

Il faut relever n'existe pas qu'une seule TFCT puisqu'elle dépend de : La durée de la fenêtre (choisie pour que le signal soit supposé stationnaire sur cette durée), la forme de la fenêtre (compromis largeur-hauteur des lobes), le taux de recouvrement entre les fenêtres.

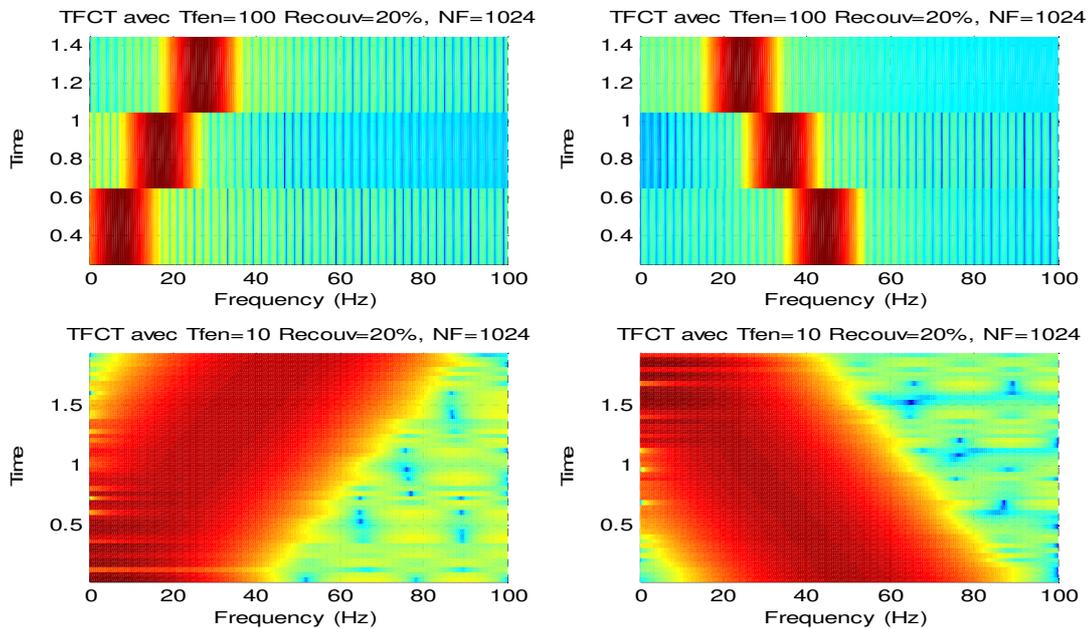
Le rôle de la fenêtre  $h(t)$ (dont l'énergie doit valoir 1) est de découper un voisinage de longueur  $L$  du point  $t$ , dans lequel le contenu fréquentiel est analysé. On conçoit qu'il y a un compromis entre la longueur  $L$  de  $h(t)$  qui représente la résolution temporelle, qui induit une résolution fréquentielle en  $f/L$ , et la capacité de la TFCT à suivre des modulations plus ou moins rapides. Ces 2 résolutions évoluent en inverse l'une de l'autre. Il a été

montré (principe d'incertitude d'Heisenberg) que la fenêtre qui réalise le meilleur compromis temps-fréquence est la fenêtre gaussienne.

Exemple : Analyse d'un signal chirp allant de  $f_1$  puis  $f_2$  et inversement (voir TP n°6)



Comme illustré ci-haut, on perd toute localisation temporelle puisqu'on obtient la même TF pour les deux signaux. Pour y remédier on calcule la TFCT en effectuant la TFD pour différents intervalles et recouvrements



Les TFDs sur chaque fenêtre glissante fournissent le spectrogramme qui permet d'adapter la TF à la caractérisation des signaux non stationnaires. On obtient, alors, une représentation temps-fréquence permettant de localiser la distribution de l'énergie simultanément en temps et fréquence. Rappelons que la longueur de la fenêtre choisie va conditionner le nombre de points fréquentiels (résolution fréquentielle) et la résolution

temporelle (spectre moyen sur la fenêtre).

Il est clair que pour ce cas, la TFCT opérant avec une taille de fenêtre unique ne permet pas de localiser précisément chaque fréquence. Celle-ci devrait s'adapter en fonction de l'évolution du signal. L'idéal serait de pouvoir choisir une fenêtre et une forme d'onde (signal oscillant dans une fenêtre temporelle donnée) que l'on pourrait dilater (pour les basses fréquences) et contracter (fréquences élevées) à volonté.

### 2. Ondelettes continues

Elle a été introduite par Jean Morlet en 1981 pour résoudre des problèmes de signaux sismiques en recherche pétrolière.

Partant d'une fenêtre  $h$  (dite fonction mère) ayant pour symbole  $\psi$  dépendant de  $t$ , on peut générer un ensemble de fonctions de base similaire par dilatation (indice  $a$ ) et translation (indice  $b$ ) d'un seul prototype  $\psi_{a,b}(t)$  :

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \quad a > 0$$

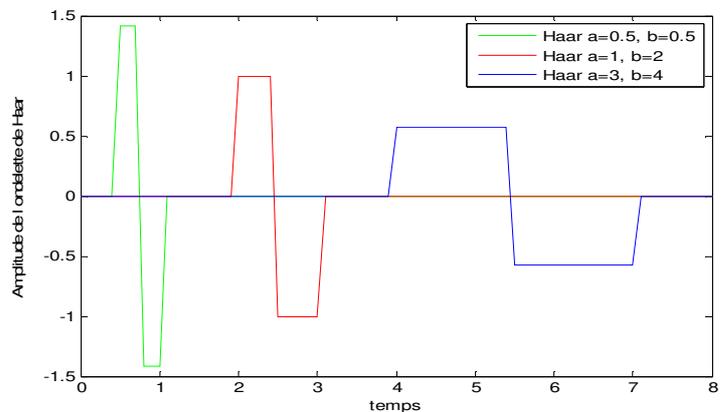
Où  $a > 0$  est un paramètre d'échelle de contraction ( $a < 1$ ) ou de dilatation ( $a > 1$ ) de la fenêtre et  $b$  une translation de la fenêtre. On notera que la norme de  $\psi_{a,b}$  est conservée lors du changement de facteur d'échelle (voir figure ci-dessous):

$$\|\psi_{a,b}\| = \int_{-\infty}^{+\infty} \frac{1}{a} \left| \psi\left(\frac{t-b}{a}\right) \right|^2 dt = \|\psi\|^2$$

Exemple :

La fonction de Haar  $\psi(t) = \begin{cases} 1 & \text{si } 0 \leq t \leq \frac{1}{2} \\ -1 & \text{si } \frac{1}{2} \leq t \leq 1 \\ 0 & \text{ailleurs} \end{cases}$

ainsi  $\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) = \begin{cases} \frac{1}{\sqrt{a}} & \text{si } b \leq t \leq b + \frac{a}{2} \\ -\frac{1}{\sqrt{a}} & \text{si } b + \frac{a}{2} \leq t \leq b + a \\ 0 & \text{ailleurs} \end{cases}$



### Exemples d'ondelettes

- L'ondelette chapeau mexicain (Paul à l'ordre 2)  $\psi(t) = \frac{1}{\sigma\sqrt{2\pi}} \left(1 - \frac{t^2}{\sigma^2}\right) e^{-\frac{t^2}{2\sigma^2}}$

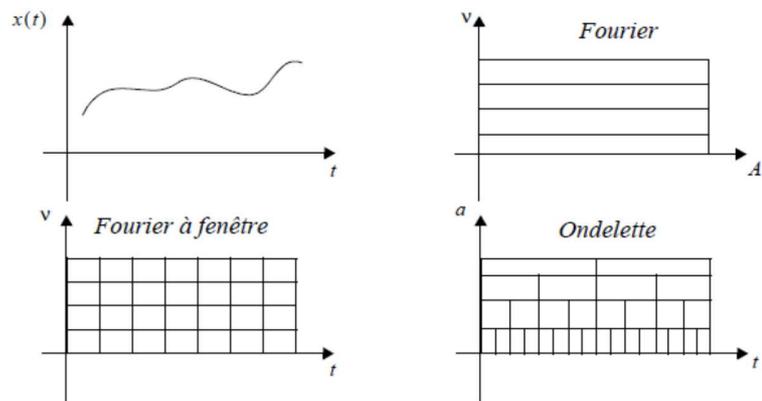
- L'ondelette de Morlet  $\psi(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{t^2}{2\sigma^2}} e^{-2\pi jft}$

L'ondelette de Morlet est une sinusoïde complexe modulée par une gaussienne. L'ondelette de Paul décroît plus vite que celle de Morlet et autorise des localisations en temps plus précises. L'ondelette est obtenue par dérivée d'une gaussienne et permet des localisations temporelles d'une qualité légèrement inférieure à celle de Paul [25]

En décomposant le signal  $x(t)$  sur cette famille, on obtient ainsi les coefficients d'ondelettes  $WT_{x,\psi}(a, b)$  qui caractérisent le coefficient de la décomposition du signal  $x(t)$  dans cette base, soit :

$$WT_{x,\psi}(a, b) = \int_{-\infty}^{\infty} x(t)\psi_{a,b}^*(t) dt = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t)\psi\left(\frac{t-b}{a}\right)^* dt$$

L'analyse par ondelettes commence avec une fenêtre d'analyse  $a$  de largeur très fine, la translate sur tout le signal puis recommence en augmentant l'échelle [25]. Ses coefficients mesurent, en un certain sens, les fluctuations du signal  $x(t)$  autour du point  $t = b$ , à l'échelle fournie par  $a$ . En diminuant  $a$ , le support de  $\psi_{a,b}$  réduit en temps et donc couvre une plage en fréquence plus grande et vice versa. Donc  $1/a$  est proportionnel à une fréquence.



Pour un facteur d'échelle assez grand, la représentation des coefficients d'ondelettes en fonction de  $b$ , la position, donne une représentation de "la forme générale de la fonction". Par contre un facteur d'échelle faible correspond à une représentation des singularités.

La transformée en ondelettes est un opérateur linéaire, invariant par translation, et par dilatation. Quelle que soit l'échelle et quel que soit l'endroit, l'analyse du signal se fait avec la même fonction. La transformée en ondelettes d'un signal n'est pas unique, elle dépend de l'ondelette mère utilisée. En effet, l'ondelette mère  $\psi(t)$  devra avoir une bonne localisation (nulle en dehors d'un certain intervalle), et devra être oscillante le nombre de moments nuls correspond au nombre d'oscillations).

On peut montrer que si la fonction analysante (l'ondelette) est correctement choisie, la transformation en ondelettes est inversible. Le signal  $x(t)$  peut être reconstruit après double intégration suivant le facteur d'échelle  $a$  et le paramètre de translation  $b$  :

$$x(t) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{1}{a^2} WT_{x,\psi}(a, b)\psi_{a,b}^*(t) da db$$

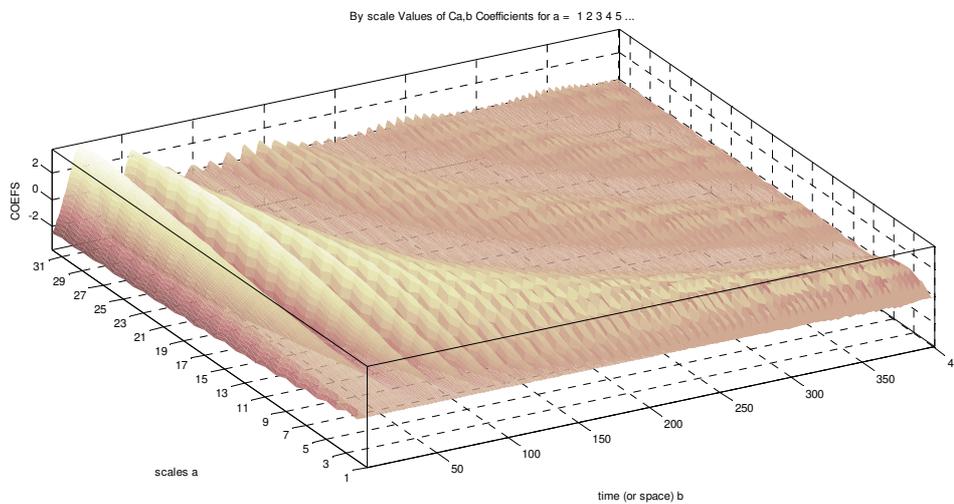
Exemple d'application : Analyse du signal chirp avec l'ondelette de Haar (Voir TP n°6)

Absolute Values of Ca,b Coefficients for a = 1 2 3 4 5 ...



Les coefficients de l'ondelette sont représentés en fonction du temps où les valeurs les plus élevées sont de couleur claire. On note qu'un coefficient a une amplitude d'autant plus grande que l'ondelette ressemble au signal sur la portion analysée. Lorsque la fenêtre est étroite (onde étroite), on observe les hautes fréquences et lorsqu'elle est large, ce sont les coefficients des basses fréquences qui sont élevés.

Toutefois bien que la transformation continue en ondelettes permette d'obtenir une bonne localisation temps-fréquence, toutefois, l'information engendrée par la transformation est infiniment redondante (a et b sont continues engendrant une sur analyse). Pour y remédier, on a recours à l'emploi d'ondelettes discrètes dyadiques.



### 3. Ondelettes discrètes (DWT), ondelettes dyadiques et analyse multi-résolution

Dans les applications, comme le traitement des images, la quantité d'information à traiter peut être importante, il est nécessaire d'optimiser les calculs et la taille des données. La multi-résolution apporte une solution à ce type de problème. Celle-ci est basée sur une transformation discrète qui diffère de la précédente d'un facteur 2 (transformation dyadique). Pour limiter la quantité d'information générée, l'ondelette encode uniquement la différence d'information entre deux résolutions successives. Cette technique, se prête bien à une décomposition/composition par banc de filtre [25]

Dans la plupart des formulations de la DWT (transformée en ondelettes discrètes), les bases respectives sont donc dérivées par une échelle dyadique des bases avec  $a=2^{-j}$  et un changement d'unité  $b=k 2^{-j}$ :

Les valeurs discrètes du facteur d'échelle  $a$  et du paramètre de translation  $b$  seront considérées sous la forme :

$$a = 2^{-j} \text{ et } b = k 2^{-j}$$

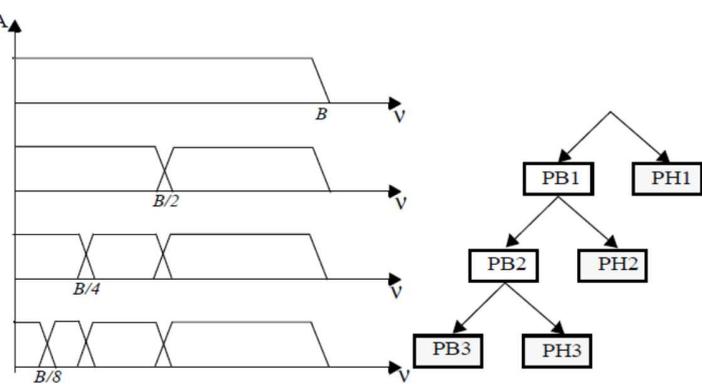
Avec ces valeurs de  $a$  et  $b$ , l'équation devient :  $C_x(2^{-j}, k 2^{-j}) = 2^{\frac{j}{2}} \int_{-\infty}^{+\infty} x(t) \psi(2^j t - k) dt$

Si la fonction  $x(t)$  est discrétisée, en supposant une période d'échantillonnage égale à 1, pour des raisons de simplicité, l'équation s'écrit alors :  $C_x(2^{-j}, k 2^{-j}) = 2^{\frac{j}{2}} \sum_n x(n) \psi(2^j n - k)$

La construction de telles ondelettes peut s'aborder comme un problème de choix de base de décomposition de signal, mais aussi comme un problème de décomposition en sous-bandes (voir chapitre précédent).

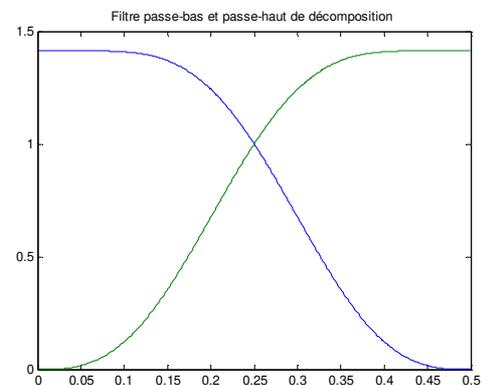
L'idée de l'analyse multirésolution développée par Meyer et Mallat permet d'analyser un signal à différentes échelles à travers des opérateurs linéaires à des niveaux de résolutions correspondant à différentes bandes de fréquences spatiales.

Le signal est décomposé de manière itérative par un banc de filtres. A chaque niveau, le spectre du signal est partagé en deux bandes par un filtre passe-bas et un par filtre passe-haut. Le filtre passe-bas (fonction d'échelle) donne les informations grossières tandis que le filtre passe-haut (ondelettes) encode les détails. Le processus est répété jusqu'à obtenir la résolution souhaitée. L'avantage de cette méthode est de n'utiliser que deux filtres. Quand le processus est arrêté, le signal peut être défini par le jeu de coefficients correspondant [25]: {PB3, PH3,PH2,PH1}.

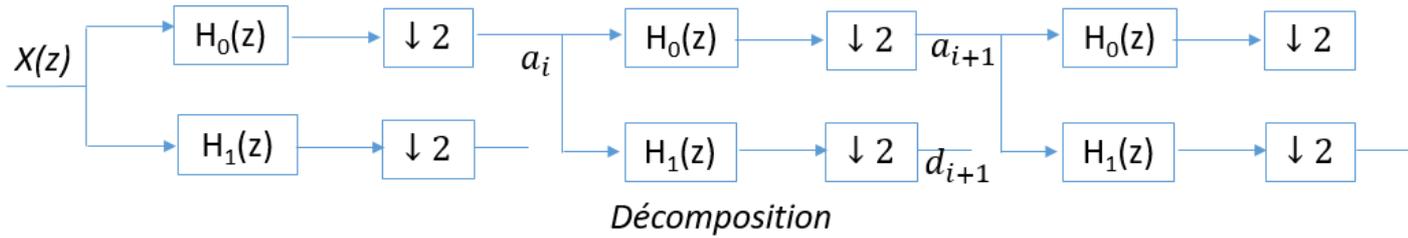


L'ondelette doit être très compacte ainsi, ses coefficients doivent être, pour la plupart, proche de zéro. Cette condition dépend de quelques paramètres comme la régularité de la fonction, le nombre de moments nuls et la taille de son support. La compacité du support impose que les filtres ( $h_0$ ) qui vont engendrer la fonction d'échelle (donnant une approximation grossière du signal) et l'ondelette ( $h_1$  fournissant les détails) soient à réponse impulsionnelle finie.

Cette condition impose d'utiliser des filtres miroirs conjugués, Les deux filtres sont symétriques conjugués autour de la fréquence centrale. (Passe-bas et passe haut ont la même fréquence de coupure). Les plus connus sont les filtres de Daubechies à support compact. Les filtres en quadrature ont une réponse impulsionnelle finie. Ils utilisent un filtre passe-bas et un filtre passe-haut. Le terme quadrature vient du fait que la somme des modules au carré des deux filtres est constante (2 ou 1) dite condition d'orthogonalité. La bande passante utilisée, par chacun des filtres, est la moitié de la bande initiale. Ils se coupent en  $f_c/4$ . On peut dans ce cas, utiliser un décimateur pour réduire la quantité d'information d'un facteur 2 [25].



La transformée en ondelettes discrète sur des bases orthonormées se ramène à des opérations de filtrage numérique suivies de sous échantillonnage. La reconstruction est parfaite et s'effectue également par des filtrages numériques précédés de sur-échantillonnage [27]



La décomposition suivant l’algorithme de Mallat est obtenue [27]:

$$a_{j+1}[n] = \sum_k h_0[2n - k]a_j[k]$$

$$d_{j+1}[n] = \sum_k h_1[2n - k]a_j[k]$$

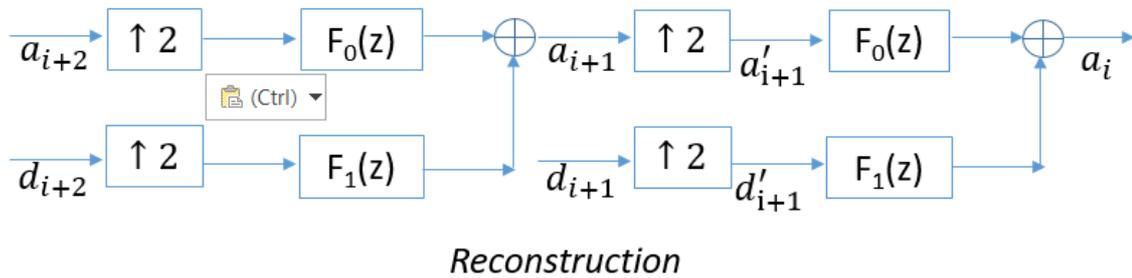
Et la reconstruction suivant l’algorithme de Mallat est obtenue par:

$$a_j[n] = \sum_k \{f_0[n - 2k]a_{j+1}[k] + f_1[n - 2k]d_{j+1}[k]\}$$

Remarque

Si on appelle  $a'_{j+1}$  le signal  $a_{j+1}$  après interpolation (insertion d'un 0 entre les 2 échantillons successifs) et  $d'_{j+1}$  le signal  $d_{j+1}$  après interpolation alors [27]:

$$a_j[n] = \sum_k \{f_0[n - k]a'_{j+1}[k] + f_1[n - k]d'_{j+1}[k]\}$$



Nous avons vu au chapitre précédent qu'un banc de filtres à reconstruction parfaite pouvait s'obtenir via les filtres CQF. Pour connaître les filtres  $h_0(n), h_1(n), f_0(n), et f_1(n)$ , dans le cas de bases orthogonales, il suffit de connaître seulement le filtre  $h_0(n)$  puisque les trois autres filtres se calculent à partir de ce dernier (filtres de reconstruction identiques aux filtres d’analyse mais retournés dans le temps).

$$H_1(z) = -H_0(-z^{-1})z^{-(L-1)} \quad \text{soit} \quad h_1(n) = (-1)^n h_0(L-1-n)$$

$$F_0(z) = H_1(-z) = H_0(z^{-1})z^{-(L-1)} \quad \text{soit} \quad f_0(n) = h_0(L-1-n)$$

$$F_1(z) = -H_0(-z) \quad \text{soit} \quad f_1(n) = -(-1)^n h_0(n)$$

**Exemple 1**

$$H_0(z) = b_0 + b_1z^{-1} + b_2z^{-2} + b_3z^{-3}$$

$$\Rightarrow H_1(z) = -(b_0 - b_1z^1 + b_2z^2 - b_3z^3)z^{-3} = b_3 - b_2z^{-1} + b_1z^{-2} - b_0z^{-3}$$

$$\Rightarrow F_0(z) = (b_0 + b_1z^1 + b_2z^2 + b_3z^3)z^{-3} = b_3 + b_2z^{-1} + b_1z^{-2} + b_0z^{-3}$$

$$\Rightarrow F_1(z) = -(b_0 - b_1z^{-1} + b_2z^{-2} - b_3z^{-3}) = -b_0 + b_1z^{-1} - b_2z^{-2} + b_3z^{-3}$$

**Remarque : Dans ce qui suit nous adopterons la convention suivante le premier élément correspond à l'indice 1 (et non pas 0) pour être en conformité avec les TPs fait sous matlab**

Ainsi, considérons par exemple l'ondelette de Daubechies à 6 coefficients telle que la fonction d'échelle  $h$  soit :

$$h_0 = [0.2352 \ 0.5706 \ 0.3252 \ -0.0955 \ -0.0604 \ 0.0249],$$

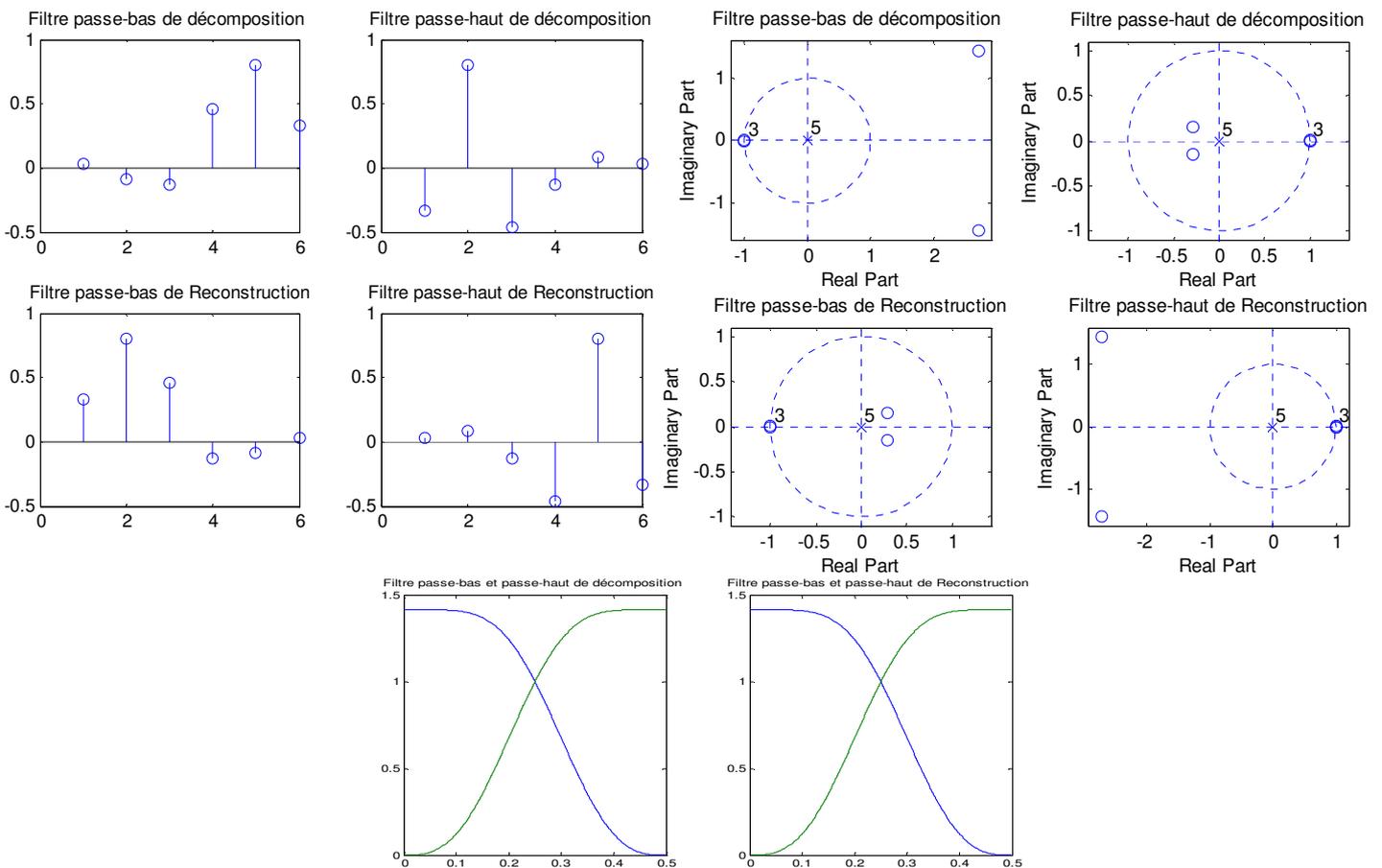
$h_1$  est obtenu en retournant  $h$  dans le temps et en inversant le signe des coefficients impairs, ce qui nous donne :

$$h_1 = [-0.0249 \ -0.0604 \ 0.0955 \ 0.3252 \ -0.5706 \ 0.2352]$$

Filtres de reconstruction identiques aux filtres d'analyse mais retournés dans le temps):

$$f_0 = [0.0249 \ -0.0604 \ -0.0955 \ 0.3252 \ 0.5706 \ 0.2352], \quad f_1 = [0.2352 \ -0.5706 \ 0.3252 \ 0.0955 \ -0.0604 \ -0.0249]$$

Ci - dessous sont donnés les coefficients d'ondelettes de Daubechies 3 avec le tracé des pôles et zéros ainsi que des réponses en fréquence (Voir TP n°6)



Remarquons que ce sont filtre CQF (vor chapitre précédent ) et donc sont orthogonaux [28] :

$$|H_0(f)|^2 + |H_0(f \pm f_e/2)|^2 = 2, \quad |H_1(f)|^2 + |H_1(f \pm f_e/2)|^2 = 2$$

$$|H_0(f)|^2 + |H_1(f)|^2 = 2 \quad H_0(f)H_1^*(f) + H_0(f \pm f_e/2)H_1^*(f \pm f_e/2) = 0$$

Comme les coefficients ne sont pas symétriques, le déphasage ne sera pas linéaire. Rappelons que la seule ondelette orthogonale qui soit symétrique est l'ondelette de Haar.

Exemple d'application 1 : On suppose que le signal à analyser est une rampe et que l'ondelette est celle de Haar.

Filtres de **Décomposition**  $h_0 = \left[ \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \right]$ ,  $h_1 = \left[ -\frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \right]$ , Filtres de **Reconstruction**  $f_0 = \left[ \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \right]$ ,  $f_1 = \left[ \frac{1}{\sqrt{2}} \quad -\frac{1}{\sqrt{2}} \right]$

Décomposition

$$a_{j+1}[n] = \sum_k h_0[2n - k] a_j[k] = \sum_k f_0[k - 2n] a_j[k] \quad d_{j+1}[n] = \sum_k h_1[2n - k] a_j[k] = \sum_k f_1[k - 2n] a_j[k]$$

	Approximation	Détail
	$h_0[-k] = \left[ \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \right] = f_0[k]$	$h_1[-k] = \left[ \frac{1}{\sqrt{2}} \quad -\frac{1}{\sqrt{2}} \right] = f_1[k]$
Niveau 0	0,1,2,3,4,5,6,7	
Niveau 1	$\frac{1}{\sqrt{2}}, \frac{5}{\sqrt{2}}, \frac{9}{\sqrt{2}}, \frac{13}{\sqrt{2}}$	$-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}$

Reconstruction:

$$a_j[n] = \sum_k \{f_0[n - k] a_{j+1}[k] + f_1[n - k] d_{j+1}[k]\} = \sum_k \{h_0[k - n] a_{j+1}[k] + h_1[k - n] d_{j+1}[k]\}$$

	Approximation $f_0[-k] = \left[ \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \right] = h_0[k]$	Détail $f_1[-k] = \left[ -\frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \right] = h_1[k]$
Niveau 1	$\frac{1}{\sqrt{2}}, \frac{5}{\sqrt{2}}, \frac{9}{\sqrt{2}}, \frac{13}{\sqrt{2}}$  $0, \frac{1}{\sqrt{2}}, 0, \frac{5}{\sqrt{2}}, 0, \frac{9}{\sqrt{2}}, 0, \frac{13}{\sqrt{2}}, 0$  $\frac{1}{2}, \frac{1}{2}, \frac{5}{2}, \frac{5}{2}, \frac{9}{2}, \frac{9}{2}, \frac{13}{2}, \frac{13}{2}$	$-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}$  $0, -\frac{1}{\sqrt{2}}, 0, -\frac{1}{\sqrt{2}}, 0, -\frac{1}{\sqrt{2}}, 0, -\frac{1}{\sqrt{2}}, 0$  $-\frac{1}{2}, \frac{1}{2}, -\frac{1}{2}, \frac{1}{2}, -\frac{1}{2}, \frac{1}{2}, -\frac{1}{2}, \frac{1}{2}$
Niveau 0	0,1,2,3,4,5,6,7	

Exemple d'application 2 :

On suppose que le filtrage passe-bas  $h_0$  est un opérateur de moyennage entre 2 coefficients et que le passe-haut  $h_1$  est un opérateur qui encode la demi-différence entre 2 coefficients [cours 2013] tels que  $h_0 = \{0.5, 0.5\}$ ,  $h_1 = \{-0.5, 0.5\}$ ,  $f_0 = 2 * \{0.5, 0.5\}$ ,  $f_1 = 2 * \{0.5, -0.5\}$

Signal original : 11, 9, 5, 7

Décomposition

	Approximation $h_0[-k]=\{0.5, 0.5\}$	Détail $h_1[-k]=\{0.5, -0.5\}$
Niveau 1	10, 6	1, -1
Niveau 2	8	2
Signal décomposé	8, 2, 1, -1	

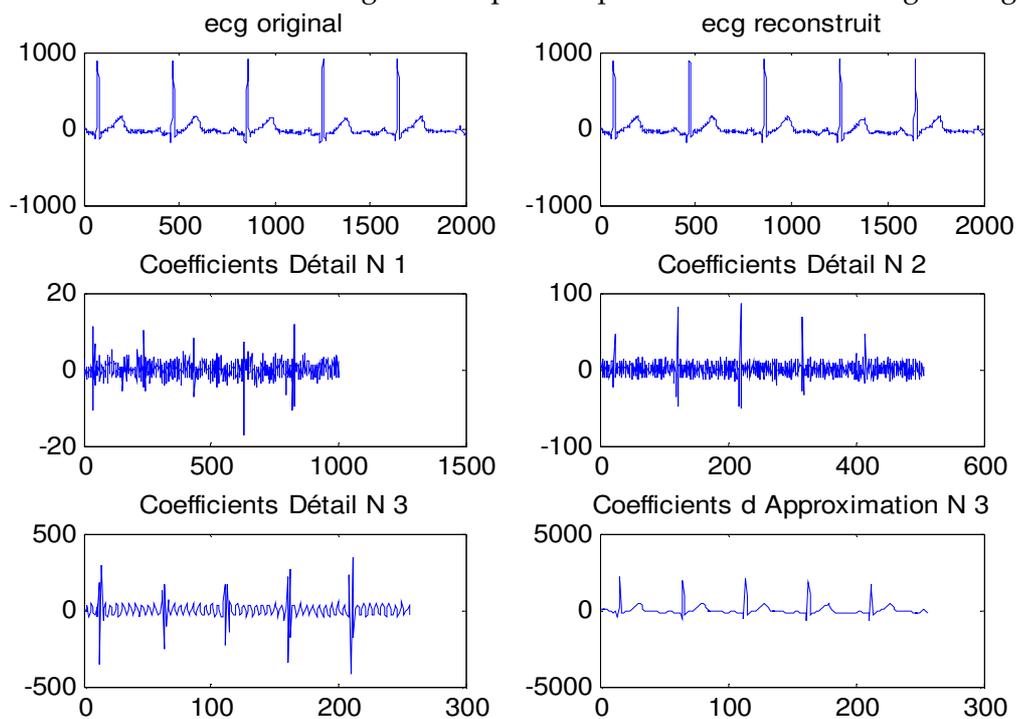
Reconstruction

	Approximation $f_0[-k]=\{1, 1\}$	Détail $f_1[-k]=\{-1, 1\}$
Niveau 2	8 0, 8, 0 [8, 8]	2 0,2,0 [2, -2]
Niveau 1	10, 6 0,10, 0, 6, 0 [10,10,6,6]	1, -1 0, 1, 0, -1 [1,-1,-1,1]
Signal reconstruit	11, 9, 5, 7	

L'opérateur  $h_0$  donne une représentation approximative alors que  $h_1$  donne une représentation plus précise (détail). A la fin du processus, on garde les détails et la dernière approximation, on commençant par le dernier niveau, on obtient :  $S_D = \{8,2,1,-1\}$ .

Il est possible de reconstruire le signal à partir des éléments de la décomposition en sommant le résultat de l'application des filtres  $f_0$  et  $f_1$ .

Exemple 3 Voici le résultat de la décomposition du signal ecg, ne sont gardés à la fin que la dernière approximation et tous les niveaux de détails grâce auxquels on pourra reconstruire le signal original (TP n°6)



**Remarques :**

- Les filtres de Daubechies conduisent à des filtres à réponse impulsionnelle finie (RIF) mais ne sont pas linéaires en phase.
- On peut construire des bases d'ondelettes conduisant à des filtres RIF linéaires en phase telles les B splines. Il s'agit des bases biorthogonales où familles de décomposition et reconstruction sont différentes et orthogonales uniquement entre elles [29]:

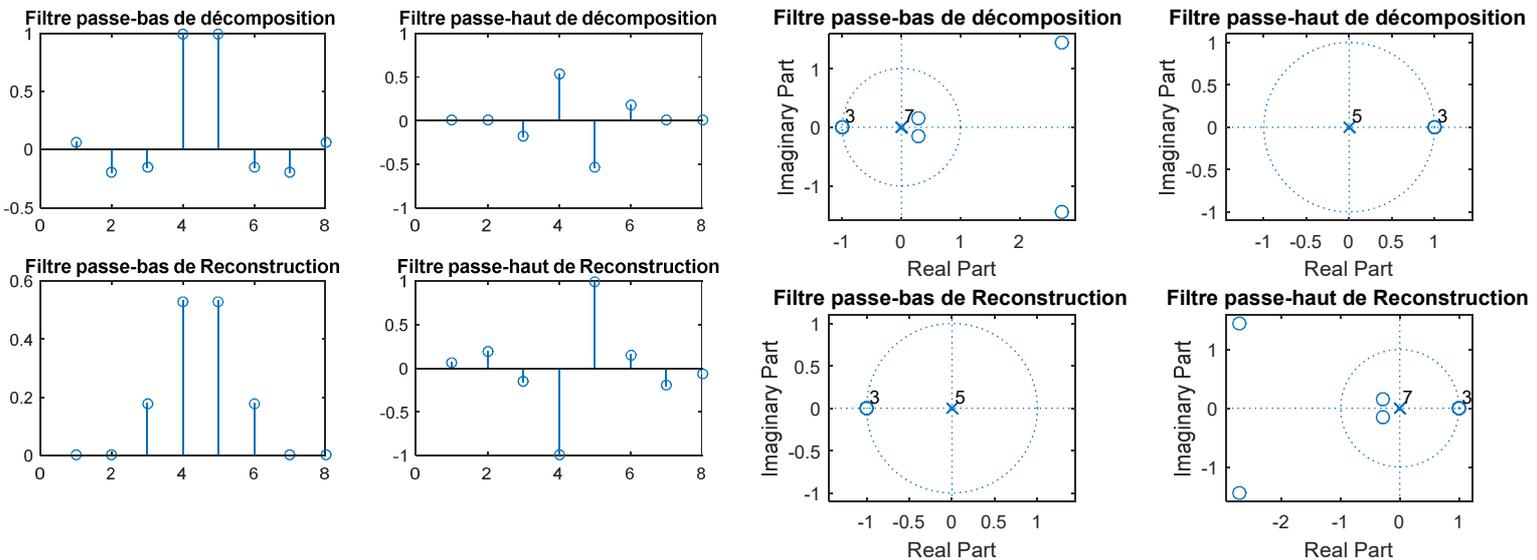
$$\begin{aligned}
 H_0(f)F_0^*(f) + H_0(f + f_e/2)F_0^*(f + f_e/2) &= 2 & \text{et} & & H_1(f)F_1^*(f) + H_1(f + f_e/2)F_1^*(f + f_e/2) &= 2 \\
 H_0(f)F_1^*(f) + H_0(f + f_e/2)F_1^*(f + f_e/2) &= 0 & \text{et} & & H_1(f)F_0^*(f) + H_1(f + f_e/2)F_0^*(f + f_e/2) &= 0
 \end{aligned}$$

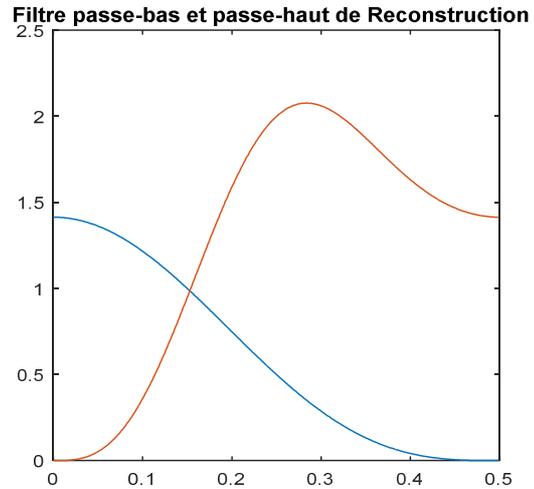
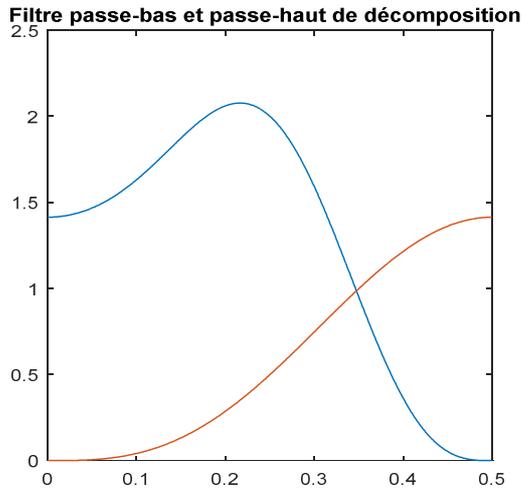
- En traitement d'images, on préférera les filtres bi-orthogonaux (JPEG2000).

Exemple d'ondelettes biorthogonales

Les filtres passe-bas et passe haut de décomposition sont différents tout comme les filtres de reconstruction. Cependant les filtres de décomposition et reconstruction sont duaux. En effet, on obtient le filtre Passe-haut de reconstruction  $F_1(z)$  en retournant dans le temps le passe-bas de décomposition  $H_0(z)$  et en inversant le signe d'un coefficient sur deux. De même pour le filtre passe-bas de reconstruction  $F_0(z)$  en retournant dans le temps le passe-bas de décomposition  $H_1(z)$  et en inversant le signe d'un coefficient sur deux, soient :

$$\begin{aligned}
 h_1(n) &= (-1)^n f_0(L - 1 - n) \\
 f_1(n) &= (-1)^n h_0(L - 1 - n)
 \end{aligned}$$





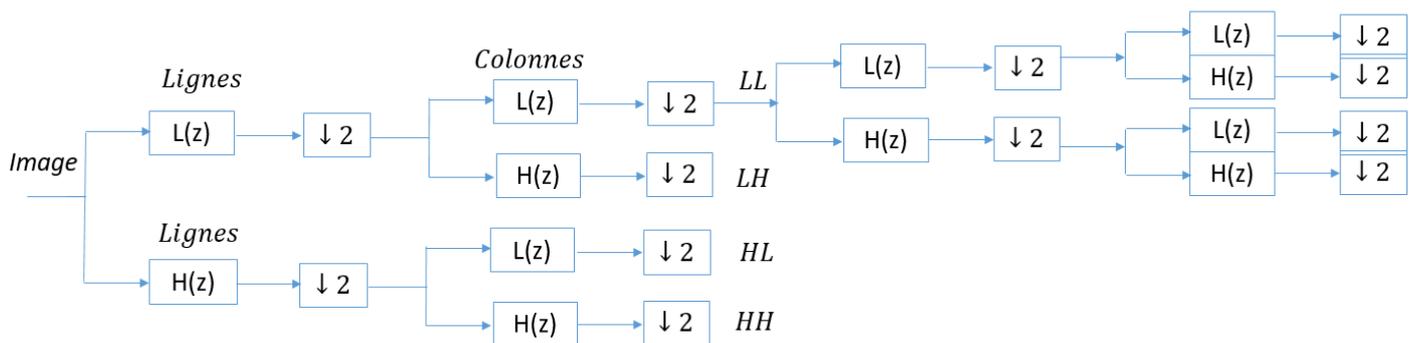
On notera la symétrie des réponses impulsionnelles des filtres passe-bas et l'anti-symétrie des filtres passe-haut d'analyse et de reconstruction conduisant donc à un déphasage linéaire. On notera également la dualité entre les filtres d'analyse et reconstruction.

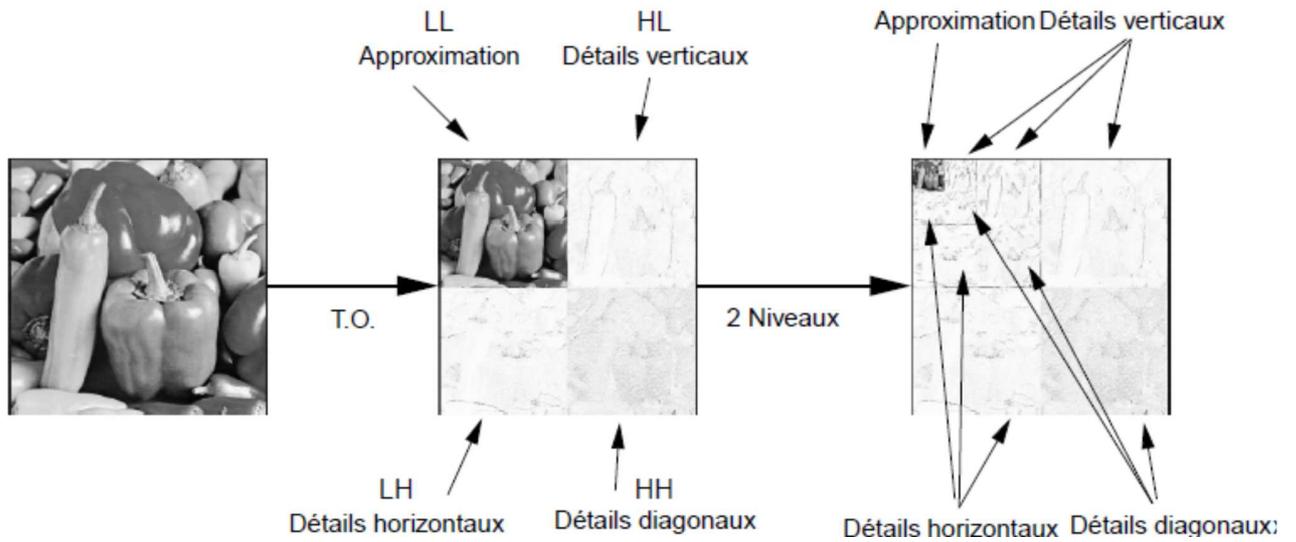
Remarque : Le lifting scheme permet une implantation très simple des décompositions en ondelettes et de leurs opérations inverses en employant la factorisation polyphases [21]

#### 4. Exemples d'application sur une image

Les ondelettes sont bien adaptées au débruitage des signaux. Le principe est simple, on décompose le signal bruité et l'on force à zéro les coefficients qui ne passent pas un seuil. Puis l'on reconstruit le signal (Voir TP n°6). De la même façon on peut compresser le signal.

Pour une image, on appliquera deux filtres le long des lignes l'un passe-haut (H), l'autre passe-bas (L), ensuite, on ne considère qu'une colonne sur deux et on applique à nouveau les 2 filtres. En ne considérant qu'une ligne sur deux, on obtient 4 images de taille réduite de moitié. En réitérant ce processus sur l'image d'approximation (LL), on obtient une pyramide d'images.





Cet exemple donne le résultat d'une compression jpeg et jpeg2000 (Ondelettes) avec un bpp de 0.2



Serie n°6 (en cours d'élaboration)

1. Déterminer la TFCT du signal suivant (h est une fenêtre porte de largeur T):

$$x(t) = \begin{cases} e^{2\pi j f_1 t} & \text{pour } t < t_0 \\ e^{2\pi j f_2 t} & \text{pour } t > t_0 \end{cases}$$

Solution :  $x(t) =$

$$\begin{cases} T \operatorname{sinc}((f - f_1)T) e^{-2\pi j(f - f_1)t} & \text{pour } t + \frac{T}{2} < t_0 \\ T \operatorname{sinc}((f - f_2)T) e^{-2\pi j(f - f_2)t} & \text{pour } t - \frac{T}{2} > t_0 \\ \left(t_0 - t + \frac{T}{2}\right) \operatorname{sinc}\left(\left(t_0 - t + \frac{T}{2}\right)(f - f_1)T\right) e^{-2\pi j(f - f_1)\left(\frac{t_0+t}{2} - \frac{T}{4}\right)} + \\ \left(\frac{T}{2} - t_0 + t\right) \operatorname{sinc}\left(\left(\frac{T}{2} - t_0 + t\right)(f - f_2)T\right) e^{-2\pi j(f - f_2)\left(\frac{t_0+t}{2} + \frac{T}{4}\right)} & \text{pour } -\frac{T}{2} < t - t_0 < \frac{T}{2} \end{cases}$$

2. Soient les ondelettes suivantes :

- La fonction de Haar  $\psi(t) = \begin{cases} 1 & \text{si } 0 \leq t \leq \frac{1}{2} \\ -1 & \text{si } \frac{1}{2} \leq t \leq 1 \\ 0 & \text{ailleurs} \end{cases}$

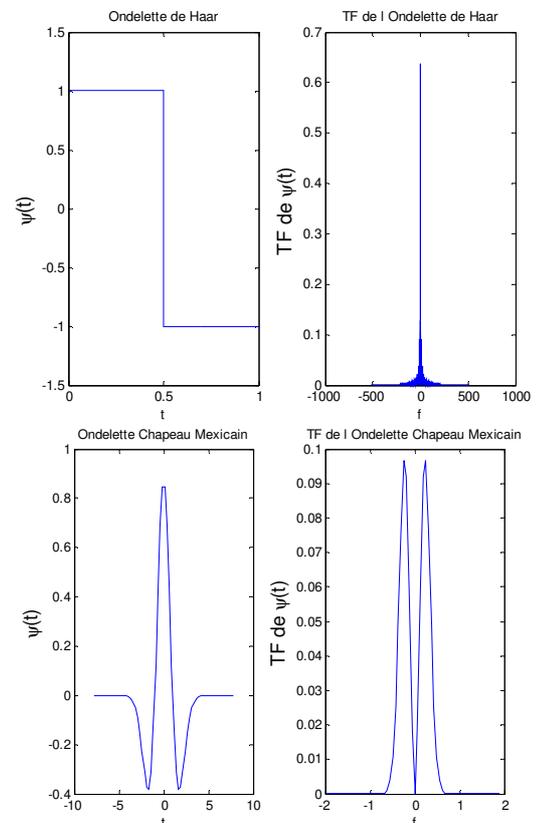
- L'ondelette chapeau mexicain  $\psi(t) = \alpha(1 - t^2)e^{-\frac{t^2}{2}}$

Calculer leur TF et tracer la

Solution :

Haar  $\hat{\psi}(f) = j \frac{\sin^2\left(\frac{\pi f}{2}\right)}{\frac{\pi f}{2}} e^{-\pi j f}$

Chapeau mexicain  $\hat{\psi}(f) = \beta 4\pi^2 f^2 e^{-\frac{f^2}{2}}$



3. Soit  $x(t) = A U(t-t_0)$ , montrer que les coefficients d'ondelettes sont nuls en dehors d'un domaine triangulaire du plan. On suppose que l'on utilise une ondelette de Haar de largeur T centrée en 0 ? Calculer Calculer alors les coefficients.

Solution

$$\begin{aligned} c_x(a, \tau) &= A\sqrt{a} \int_{-\frac{T}{2}}^{\frac{T}{2}} \psi(u) du = 0 & \text{si } \frac{t_0 - \tau}{a} < -\frac{T}{2} \text{ (condition d'admissibilité)} \\ c_x(a, \tau) &= A\sqrt{a} \int_{\frac{t_0 - \tau}{a}}^{\frac{T}{2}} \psi(u) du = -A\sqrt{a} I_\psi\left(\frac{t_0 - \tau}{a}\right) & \text{si } -\frac{T}{2} < \frac{t_0 - \tau}{a} < \frac{T}{2} \text{ cône de discontinuité} \\ c_x(a, \tau) &= 0 & \text{si } \frac{t_0 - \tau}{a} > \frac{T}{2} \end{aligned}$$

4. Soit une ondelette  $\psi(t)$  dont la TF est  $\Phi(f)$  possédant un maximum en  $f_M$  et de bande F. Calculer les coefficients pour  $x(t) = e^{2\pi j f_0 t}$

$$c_x(a, \tau) = \frac{A}{\sqrt{a}} \int_{\mathbb{R}} e^{j2\pi f_0 t} \psi\left(\frac{t-\tau}{a}\right) dt = \frac{A}{\sqrt{a}} TF^* \left[ \psi\left(\frac{t-\tau}{a}\right) \right]_{f=f_0}$$

$$\begin{matrix} \psi(t) & \xrightarrow{\text{dilatation}} & \psi\left(\frac{t}{a}\right) & \xrightarrow{\text{retard}} & \psi\left(\frac{t-\tau}{a}\right) \\ \downarrow & & \downarrow & & \downarrow \\ \Phi(f) & & |a| \Phi(af) & & |a| \Phi(af) e^{-j2\pi f \tau} \end{matrix}$$

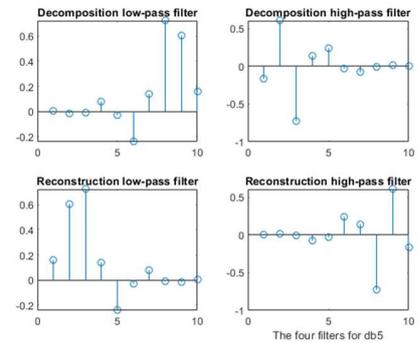
$$c_x(a, \tau) = \frac{A}{\sqrt{a}} |a| (\Phi(af_0) e^{-j2\pi f_0 \tau})^* = A\sqrt{a} \Phi^*(af_0) e^{j2\pi f_0 \tau}$$

$$\neq 0 \text{ pour } \frac{f_m}{f_0} - \frac{F}{2f_0} \leq a \leq \frac{f_m}{f_0} + \frac{F}{2f_0}$$

pour  $\Phi$  de support borné  $\left[ f_m - \frac{F}{2}; f_m + \frac{F}{2} \right]$

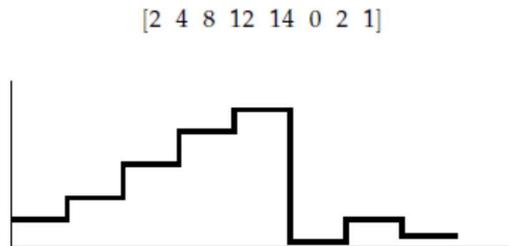
5. Soit le filtre passe bas de décomposition suivant :  $h_0 = \{-0.1294, 0.2241, 0.8365, 0.4830\}$ , déterminer le filtre passe-haut de décomposition et les filtres de reconstruction

**Solution :**  $h_1 = \{-0.4830, 0.8365, -0.2241, -0.1294\}$ ,  
 $f_0 = \{0.4830, 0.8365, 0.2241, -0.1294\}$ ,  
 $f_1 = \{-0.1294, -0.2241, 0.8365, -0.4830\}$



6. Soient les filtres de décomposition et reconstruction suivants, vérifier les propriétés liant les 4 filtres

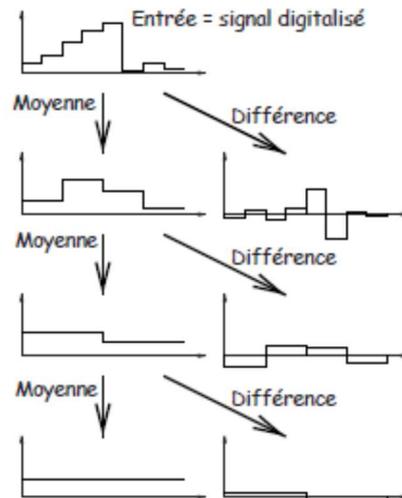
7. Soit le signal suivant, donner et tracer sa décomposition en ondelettes si la fonction d'échelle est  $h_0 = [0.5, 0.5] \Rightarrow$  alors la fonction d'ondelettes est  $h_1 = [-0.5, 0.5]$



Résolution	Moyennes	Détails
8	[ 2 4 8 12 14 0 2 1 ]	
4	[ 3 10 7 1.5 ]	[ -1 -2 7 0.5 ]
2	[ 6.5 4.25 ]	[ -3.5 2.75 ]
1	[ 5.375 ]	[ 1.125 ]

Décomposition

[5.375 1.125 -3.5 2.75 -1 -2 7 0.5]

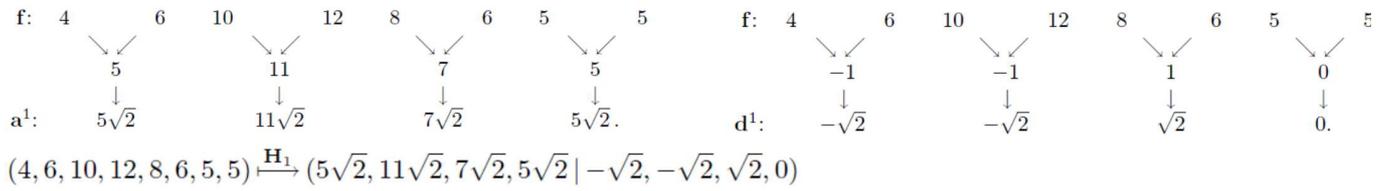


8. Soit le signal suivant :  $f = (4, 6, 10, 12, 8, 6, 5, 5)$ ;
- Donner et tracer sa décomposition au niveau 1 en ondelettes en employant l'ondelette de Haar.
  - Reconstruire le signal à partir du niveau 1
  - Calculer son énergie et puis celle après décomposition.
  - Vérifier qu'elle se conserve et que 98% de cette énergie se trouve dans le signal approximé.
  - Donner la décomposition au niveau 2 puis 3.

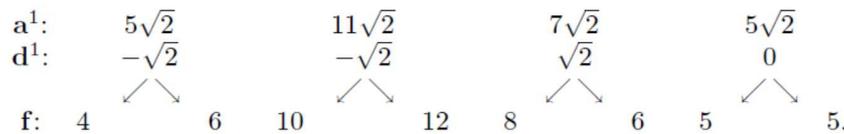
**Solution**

Filtres de Décomposition  $h_0 = \left[ \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \right]$ ,  $h_1 = \left[ -\frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \right]$ , Filtres de Reconstruction  $f_0 = \left[ \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \right]$ ,  $f_1 = \left[ \frac{1}{\sqrt{2}} \quad -\frac{1}{\sqrt{2}} \right]$

Décomposition



Reconstruction

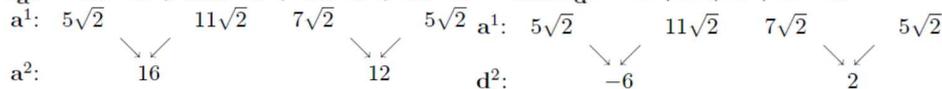


Energie

$\mathcal{E}_f = 4^2 + 6^2 + \dots + 5^2 = 446.$

$\mathcal{E}_{(a^1 | d^1)} = 25 \cdot 2 + 121 \cdot 2 + \dots + 2 + 0 = 446$

$\mathcal{E}_{a^1} = 25 \cdot 2 + 121 \cdot 2 + 49 \cdot 2 + 25 \cdot 2 = 440$   $\mathcal{E}_{d^1} = 2 + 2 + 2 + 0 = 6$



$(a^2 | d^2 | d^1) = (16, 12 \mid -6, 2 \mid -\sqrt{2}, -\sqrt{2}, \sqrt{2}, 0)$

$(a^3 | d^3 | d^2 | d^1) = (14\sqrt{2} \mid 2\sqrt{2} \mid -6, 2 \mid -\sqrt{2}, -\sqrt{2}, \sqrt{2}, 0)$

88% de l'énergie est contenue dans  $a^3$  représentant pourtant le 8<sup>ème</sup> du signal

**Exercices Supplémentaires**

- Soit le signal suivant :  $x(n) = \{0, 1, -2, 3, -4, 5, -6, 7\}$ 
  - Donner sa décomposition en ondelettes au niveau 1 en employant l'ondelette de Haar tel que le filtre passe-bas de décomposition soit  $h_0 = \left\{ \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right\}$
  - Donner le schéma de reconstruction permettant de passer d'un niveau à un autre.
  - Reconstruire le signal à partir du niveau 1.
  - Donner le signal décomposé aux niveaux 2 puis 3.
  - Vérifier que l'énergie se conserve à tous les niveaux (1, 2 et 3)
  - Proposer un moyen de compresser ce signal.

**Solution** Voir examen 17/18

- Soit le signal suivant :  $x(n) = \{11, 9, 5, 7, 5, 11, 7, 9\}$ 
  - Citer un inconvénient de l'ondelette continue.
  - Donner le schéma de décomposition en ondelettes discrètes pour 2 niveaux.
  - Donner le schéma correspondant pour la répartition en fréquences.
  - Donner sa décomposition en ondelettes au niveau 2 en employant les filtres suivants :  $h_0 = \{0.5, 0.5\}$ ,  $h_1 = \{-0.5, 0.5\}$ ,  $f_0 = 2 * \{0.5, 0.5\}$ ,  $f_1 = 2 * \{0.5, -0.5\}$
  - L'énergie se conserve-t-elle ?
  - Reconstruire le signal à partir du niveau 2.

**Solution** Voir Ratt 17/18

TP n°6 (Sous Matlab)

1. De la TFCT aux ondelettes

```
clc; clear all; close all;
Te=0.005; Fe=1/Te; Tfen=50; Trec=2; NF=1024 ; t=0:Te:2; f0=1; f1=50 ; y1=chirp(t,f0,2,f1);
figure;subplot(2,2,1); plot(t,y1);title('Chirp de f0=1Hz à f1=50Hz');
y2=chirp(t,f1,2,f0);subplot(2,2,2); plot(t,y2);title('Chirp de f1=50Hz à f0=1Hz');
yy1=fft(y1,NF);yy1=fftshift(yy1);
axef=Fe*(-1/2:1/NF:1/2-1/NF);subplot(2,2,3);plot(axef,abs(yy1));title ('FFT de y1');
yy2=fft(y2,NF);yy2=fftshift(yy2);
axef=Fe*(-1/2:1/NF:1/2-1/NF);subplot(2,2,4);plot(axef,abs(yy2));title ('FFT de y2');
```

```
figure;subplot(2,2,1); spectrogram(y1,100,20,NF,Fe);
title('TFCT avec Tfen=100 Recouv=20%, NF=1024');
subplot(2,2,2); spectrogram(y2,100,20,NF,Fe);
title('TFCT avec Tfen=100 Recouv=20%, NF=1024');
subplot(2,2,3); spectrogram(y1,10,2,NF,Fe);
title('TFCT avec Tfen=10 Recouv=20%, NF=1024');
subplot(2,2,4); spectrogram(y2,10,2,NF,Fe);
title('TFCT avec Tfen=10 Recouv=20%, NF=1024');
```

```
figure;subplot(2,1,1);COEFS = cwt(y1,1:32,'haar','plot');
subplot(2,1,2);COEFS = cwt(y1,1:129,'haar','plot');
figure; COEFS = cwt(y1,1:32,'haar','3Dlvl');
```

1. La TF permet-elle de localiser temporellement les fréquences
2. L'emploi de la TFCT permet-il de régler le problème ?
3. Quelle est la fenêtre employée par la TFCT (aider vous du help sur spectrogram) et pourquoi ?
4. Faire varier Tfen, quel est l'inconvénient alors ?
5. L'emploi de l'ondelette continue (CWT) permet-il de pallier ce problème ?
6. Quel est l'inconvénient principal ? comment y remédie-t-on ?

2. Ondelettes discrètes et applications

2.1. Filtrés d'analyse et de reconstruction

```
clc;clear all; close all;
wname = 'db3' ; [Lo_D,Hi_D,Lo_R,Hi_R] = wfilters(wname);
subplot(221); stem(Lo_D); title('Filtre passe-bas de décomposition ');
subplot(222); stem(Hi_D); title('Filtre passe-haut de décomposition');
subplot(223); stem(Lo_R); title('Filtre passe-bas de Reconstruction');
subplot(224); stem(Hi_R); title('Filtre passe-haut de Reconstruction');
figure;
subplot(221); zplane(Lo_D,1); title('Filtre passe-bas de décomposition ');
subplot(222); zplane(Hi_D,1); title('Filtre passe-haut de décomposition');
subplot(223); zplane(Lo_R,1); title('Filtre passe-bas de Reconstruction');
subplot(224); zplane(Hi_R,1); title('Filtre passe-haut de Reconstruction');
figure;
subplot(121); [Hb,f]=freqz(Lo_D,1,512,1); [Hh,f]=freqz(Hi_D,1,512,1);
plot(f,abs(Hb),f,abs(Hh)); title('Filtre passe-bas et passe-haut de décomposition ');
subplot(122); [Hb,f]=freqz(Lo_R,1,512,1); [Hh,f]=freqz(Hi_R,1,512,1);
plot(f,abs(Hb),f,abs(Hh)); title('Filtre passe-bas et passe-haut de Reconstruction ');
```

1. Identifier les fonctions d'échelle et les fonctions d'ondelettes
2. Vérifier les relations liant les 4 filtres
3. Tester une ondelette bi-orthogonale wname = 'bior1.5' et commenter

3.2 Décomposition et reconstruction

```
clc;clear all; close all;
load ecg.dat; x=ecg; wname = 'db5';
[Lo_D,Hi_D,Lo_R,Hi_R] = wfilters(wname);
[c,l] = wavedec(x,3,Lo_D,Hi_D);
approx = appcoef(c,l,'db5'); [cd1,cd2,cd3] = detcoef(c,l,[1 2 3]);
subplot(3,2,1); plot(x); title('ecg original');
subplot(3,2,2); plot(approx); title('Approximation Coefficients');
subplot(3,2,3); plot(cd3); title('Level 3 Detail Coefficients');
subplot(3,2,4); plot(cd2); title('Level 2 Detail Coefficients');
subplot(3,2,5); plot(cd1); title('Level 1 Detail Coefficients');
```

```
y = waverec(c,l,Lo_R,Hi_R); subplot(3,2,6); plot(y); title('ecg reconstruit');
```

### 3.3 Compression et dé-bruitage de signaux

Rajouter les instructions suivantes avant la reconstruction  $N=\text{length}(x)$ ;  $c(N/8+1:\text{end})=0$ ;

1. Expliquer le but de ces instructions. Modifier les niveaux de détails supprimés  $c(N/4+1:\text{end})=0$ ;  
 $c(N/2+1:\text{end})=0$ ;
2. Tester d'autres ondelettes en vous aidant du help

TP n°6 (Sous Python)

1. De la TFCT aux ondelettes

```
import numpy as np; import matplotlib.pyplot as plt; import scipy.signal as sp;
NF = 1024; Te=0.005; fe=1/Te; N=round(2/Te);
t = np.linspace(0, 2, N);
x1= sp.chirp(t, f0=50, t1=2, f1=1, method='linear');
x2= sp.chirp(t, f0=1, t1=2, f1=50, method='linear');

TFx1 = np.fft.fft(x1,NF); TFx1 = np.fft.fftshift(TFx1);
TFx2 = np.fft.fft(x2,NF); TFx2 = np.fft.fftshift(TFx2); freq = np.arange(-NF/2,NF/2)*fe/NF;
plt.figure(1)
plt.subplot(221); plt.plot(t, x1); plt.title('Chirp de 50 à 1 Hz'); plt.subplot(222); plt.plot(freq,
np.abs(TFx1));
plt.subplot(223); plt.plot(t, x2); plt.title('Chirp de 1 à 50 Hz'); plt.subplot(224); plt.plot(freq,
np.abs(TFx2));
f, tt, Sxx1 = sp.spectrogram(x1, fe, nperseg=100, noverlap=20);
f, tt, Sxx2 = sp.spectrogram(x2, fe, nperseg=100, noverlap=20);
plt.figure(2)
plt.subplot(221); plt.plot(t, x1); plt.subplot(222); plt.pcolormesh(tt, f, Sxx1)
plt.ylabel('Frequence [Hz]'); plt.xlabel('Temps [sec]'); plt.title('Spectrogramme Tfen=100, TRec=20');
plt.subplot(223); plt.plot(t, x2); plt.subplot(224); plt.pcolormesh(tt, f, Sxx2)
plt.ylabel('Frequence [Hz]'); plt.xlabel('Temps [sec]'); plt.title('Spectrogramme Tfen=100, TRec=20');
f, tt, Sxx1 = sp.spectrogram(x1, fe, nperseg=20, noverlap=4);
f, tt, Sxx2 = sp.spectrogram(x2, fe, nperseg=20, noverlap=4);
plt.figure(3)
plt.subplot(221); plt.plot(t, x1); plt.subplot(222); plt.pcolormesh(tt, f, Sxx1)
plt.ylabel('Frequence [Hz]'); plt.xlabel('Temps [sec]'); plt.title('Spectrogramme Tfen=20, TRec=4');
plt.subplot(223); plt.plot(t, x2); plt.subplot(224); plt.pcolormesh(tt, f, Sxx2)
plt.ylabel('Frequence [Hz]'); plt.xlabel('Temps [sec]'); plt.title('Spectrogramme Tfen=20, TRec=4');
import pywt
Larg = np.arange(1, 31);
cwtmatr,fs = pywt.cwt(x2, Larg, 'mexh')
plt.figure(4)
plt.imshow(cwtmatr, aspect='auto', vmax=abs(cwtmatr).max(), vmin=-abs(cwtmatr).max())
```

1. La TF permet-elle de localiser temporellement les fréquences
2. L’emploi de la TFCT permet-il de régler le problème ?
3. Quelle est la fenêtre employée par la TFCT (aider vous du help sur spectrogram) et pourquoi ?
4. Faire varier Tfen, quel est l’inconvénient alors ?
5. L’emploi de l’ondelette continue (CWT) permet-il de pallier ce problème ?
6. Quel est l’inconvénient principal ? comment y remédie-t-on ?

2. Ondelettes discrètes et applications

2.1. Filtres d’analyse et de reconstruction

```
import pywt; import matplotlib.pyplot as plt; import scipy.signal as sp;
ondelettes = pywt.Wavelet('db3')
plt.figure(1)
plt.subplot(221); plt.stem(ondelettes.dec_lo); plt.title('Passe-bas de Décomposition');
plt.subplot(222); plt.stem(ondelettes.dec_hi); plt.title('Passe-Haut de Décomposition');
plt.subplot(223); plt.stem(ondelettes.rec_lo); plt.title('Passe-bas de Reconstruction');
plt.subplot(224); plt.stem(ondelettes.rec_hi); plt.title('Passe-bas de Reconstruction');
L=512;
f,H_dec_lo=sp.freqz(ondelettes.dec_lo,[1,0],L);
f,H_dec_hi=sp.freqz(ondelettes.dec_hi,[1,0],L);
f,H_rec_lo=sp.freqz(ondelettes.rec_lo,[1,0],L);
f,H_rec_hi=sp.freqz(ondelettes.rec_hi,[1,0],L);
plt.figure(2)
plt.subplot(121); plt.plot(f,abs(H_dec_lo),f,abs(H_dec_hi)); plt.title('Passe-bas et Passe-haut de
Décomposition');
plt.subplot(122); plt.plot(f,abs(H_rec_lo),f,abs(H_rec_hi)); plt.title('Passe-bas Passe-haut de
Reconstruction');
```

1. Identifier les fonctions d'échelle et les fonctions d'ondelettes
2. Vérifier les relations d'orthogonalité liant les 4 filtres
3. Tester une ondelette bi-orthogonale wname = 'bior1.3' et commenter

## 2.2. Décomposition et reconstruction

```
import pywt; import matplotlib.pyplot as plt; import numpy as np;
x = np.genfromtxt('ecg.dat');
coeffs = pywt.wavedec(x, 'db5', level=3)
cA3, cD3, cD2, cD1 = coeffs
plt.figure(1)
plt.subplot(321); plt.plot(x); plt.title('Signal original');
plt.subplot(322); plt.plot(cA3); plt.title('Approximation niveau 3');
plt.subplot(323); plt.plot(cD3); plt.title('Détail niveau 3');
plt.subplot(324); plt.plot(cD2); plt.title('Détail niveau 2');
plt.subplot(325); plt.plot(cD1); plt.title('Détail niveau 1');
x_reconst=pywt.waverec([cA3, cD3, cD2, cD1] , 'db5');
plt.subplot(326); plt.plot(x_reconst); plt.title('Signal Reconstruit');
```

## 2.3. Compression et dé-bruitage de signaux

1. Mettre les détails de certains niveaux à 0 et reconstruire (supprimer lors de la reconstruction)
2. Choisir un et mettre les valeurs de détails en dessous du seuil à zéro.
3. Tester d'autres ondelettes et observer la qualité de la reconstruction

## 2.4. Application sur une image

A l'aide de `pywt.dwt2`, décomposer et reconstruire une image sur plusieurs niveaux, mettre des détails à zéros et commenter.

## Références

- [1] F. Cottet. Aide-mémoire Traitement du Signal. Dunod, Paris, 2005.
- [2] J.L. Lecornu D. Le Roux. Mts 201 A Signaux Numeriques Telecom Bretagne
- [3] O. Français. Théorie de l'échantillonnage et de la quantification. ESIEE.
- [4]. Maurice Charbit, Gérard Blanchet. Eléments de base pour le Traitement Numérique du Signal et de l'Image. Telecom Paris.
- [5] [https://moodle.insa-rouen.fr/pluginfile.php/6997/mod\\_resource/content/0/cours7.pdf](https://moodle.insa-rouen.fr/pluginfile.php/6997/mod_resource/content/0/cours7.pdf)
- [6] <http://chamilo2.grenet.fr/inp/courses/ENSE32A4EMAAAK0/document/html/toc1.htm>
- [7] <http://www.groupe.polymtl.ca/ele2700/TP/seance3-TP2-HIVER-2015.pdf>
- [8] Alexandre Renaux. Traitement Numérique Des Signaux. IFIPS / Université Paris Sud Orsay
- [9] Olivier Sentieys. Signaux et Systèmes Discrets. ENSSAT - Université de Rennes 1.
- [10] J.L. Zarader. Cours de Traitement du Signal. Ecole Polytechnique Universitaire de Paris
- [11] M. Frikel. Cours Traitement Numérique du Signal. ENSICAEN.
- [12] C. Doignon. Traitement Numérique du Signal Déterministe. Université Louis Pasteur de Strasbourg
- [13] <http://fr.audiofanzine.com/techniques-du-son/forums/t.392548,plug-ins-d-eq-sonnent-ils-differemment-pourquoi,p.18.html>
- [14] [http://perso.iut-nimes.fr/flouchet/syst%20ech/T\\_en\\_z.pdf](http://perso.iut-nimes.fr/flouchet/syst%20ech/T_en_z.pdf)
- [15] <http://pageperso.univ-lr.fr/pierre.courtellemont/TD4.pdf>
- [16] [https://perso.univ-rennes1.fr/laurent.albera/alberasiteweb/cours/tp\\_filtrage.pdf](https://perso.univ-rennes1.fr/laurent.albera/alberasiteweb/cours/tp_filtrage.pdf)
- [17] Joel Leroux. Techniques numériques pour le traitement du signal. Polytech, Nice-Sofia.
- [18] [http://www.eaeie.org/sites/all/themes/eaeie/tnprojects/theiere\\_signal/theiere\\_signal\\_fr/frame.htm](http://www.eaeie.org/sites/all/themes/eaeie/tnprojects/theiere_signal/theiere_signal_fr/frame.htm)
- [19] J. Prado, G. Blanchet. Méthodes Numérique pour le traitement du signal.
- [20] <https://perso.esiee.fr/~venardo/doc/algoI4.pdf>
- [21] [http://www.laurent-duval.eu/Documents-Supelec-FI/Duval\\_L\\_2009\\_Lecture-Supelec-TNS\\_Bancs-Filtres.pdf](http://www.laurent-duval.eu/Documents-Supelec-FI/Duval_L_2009_Lecture-Supelec-TNS_Bancs-Filtres.pdf)
- [22] <https://tel.archives-ouvertes.fr/tel-01128245/file/2008PEST0211.pdf>
- [23] <http://people.irisa.fr/Nancy.Bertin/wp-content/uploads/2014/09/cours6.pdf>
- [24] <https://docplayer.fr/37834686-Gel-4101-traitement-numerique-du-signal.html>
- [25] <http://people.irisa.fr/Nancy.Bertin/wp-content/uploads/2014/09/cours41.pdf>
- [26] P.Nayman, "Certains Aspects Du Traitement Du Signal ", LPNHE paris ,2012-2013.
- [27] <http://www.le2i.cnrs.fr/IMG/publications/ondelettes.pdf>
- [28] <https://www.i2m.univ-amu.fr/~caroline.chaux/GEOMDATA/TI-te5215.pdf>
- [29] <https://homepages.laas.fr/adoncsc/Wavelet/RapportWavelet.pdf>
- [30] [http://dobigeon.perso.enseeiht.fr/teaching/TNS2/Cours\\_TNS\\_part3.pdf](http://dobigeon.perso.enseeiht.fr/teaching/TNS2/Cours_TNS_part3.pdf)
- [31] <https://www.fsg.ulaval.ca/departements/professeurs/paul-fortier-111/>