

# Signaux et Systèmes Filtrage Numérique

A.Saadane

[Abdelhakim.saadane@univ-nantes.fr](mailto:Abdelhakim.saadane@univ-nantes.fr)

Département Electronique et Technologies Numériques



# Sommaire

- 1 Organisation du module
- 2 SLIT : rappels
- 3 Introduction filtrage numérique
- 4 Synthèse d'un filtre numérique non récursif
- 5 Synthèse d'un filtre numérique récursif
- 6 Structures de filtres RII
- 7 Structures de base des filtres RIF

# Objectifs du cours

- Faire des rappels sur les Systèmes Linéaires Invariants dans le Temps (SLIT).
- Introduire le filtrage numérique.
- Synthétiser un filtre numérique non récursif (FIR : Finite Impulse Response).
- Synthétiser un filtre numérique récursif (IIR : Infinite Impulse Response)
- Utiliser les filtres numériques synthétisés dans des applications de traitement de signaux réels.

# Organisation pratique du module

- **Documents**

- Transparents cours
- Polycopié Travaux Dirigés
- Polycopié Travaux Pratiques

- **Volume horaire du module**

- Cours (9 séances)
- TD et TP (10 séances)
- Contrôle (1.30H)

# Organisation pédagogique du cours

- Objectifs des séances de cours

- 2 séances : Rappels SLIT + Introduction au filtrage numérique.
- 3 séances : Filtres numériques non récursifs (FIR).
- 3 séances : Filtres numériques récursifs (IIR).
- 1 séance : Structures des filtres numériques.

- Organisation des TDs

- 2 séances : SLIT (Convolution et Propriétés).
- 1 séance : SLIT (Filtres numériques)
- 1 séance : Filtres numériques et Transformée en Z.

## • Organisation des TPs (Matlab)

- 2 séances : Filtrage multicadence : décomposition-reconstruction en sous bandes.
- 2 séances : Filtrage numérique d'un signal bruité (synthèse et structures).
- 2 séances : Récepteur DTMF.

## • Évaluation du module

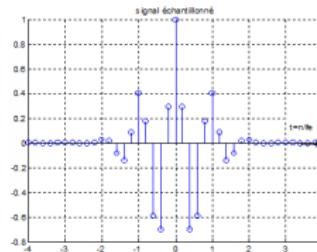
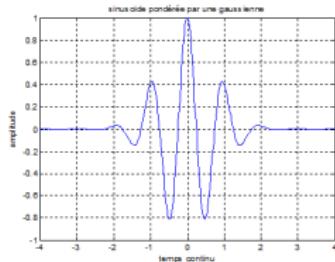
- Examen à mi-parcours (Vendredi 5 novembre 2021) 50%.
- Examen final (décembre ? Janvier ?) 50%.

# Signal Discret

- Suite de nombres

$$x_a(nT) = x(n) = x_n \quad \text{avec } n \in \mathbb{Z}$$

- $\mathbb{Z}$  l'ensemble des nombres entiers,
- $x(n)$  ou  $x_n$  l'amplitude du signal à l'instant  $nT$ ,
- $T$  période d'échantillonnage, intervalle entre 2 échantillons successifs.



- Exemple :

- Impulsion numérique :  $\delta_n = \begin{cases} 1 & \text{si } n = 0 \\ 0 & \text{si } n \neq 0 \end{cases}$
- Matlab : `impulsion = [1, zeros(1,N)];` % N = nombre de zeros.

# Système discret

- Processus induisant une transformation du signal d'entrée  $x_n$



- Caractérisé par la loi (transformation) entrée / sortie

$$y_n = T[x_n]$$

- Exemples :

- Moyenneur  $y_n = \frac{1}{4} * [x_n + x_{n-1} + x_{n-2} + x_{n-3}]$
- Elévateur au carré  $y_n = (x_n)^2$

# Système discret linéaire et invariant dans le temps

- **Linéarité :**

- $T[x_n + z_n] = T[x_n] + T[z_n]$    et    $T[ax_n] = aT[x_n]$
- Exemple : Retard  $y_n = [x_{n-m}]$ 
  - $T[x_n + z_n] = x_{n-m} + z_{n-m} = T[x_n] + T[z_n]$
  - $T[ax_n] = ax_{n-m} = aT[x_n]$

- **Invariance :** Tout décalage temporel de  $x_n$  engendre le même décalage de  $y_n$

- $y_n = T[x_n] \implies y_{n-k} = T[x_{n-k}]$

- Exemple : élévateur au carré

$$y_n = (x_n)^2 \implies y_{n-k} = (x_{n-k})^2 = T[x_{n-k}]$$

# Système causal et stable

- **Causalité :** La sortie à un instant  $n$  ne dépend que des valeurs à l'instant  $n$  et des valeurs précédentes (instants  $n-1, n-2, \dots$ )

- Exemples :

- Système récursif :  $y_n = x_n + x_{n-1} + y_{n-1} + y_{n-2}$
- Système non récursif :  $y_n = x_n + x_{n-1}$

- Contre exemple :

- Système non récursif :  $y_n = x_n + x_{n+1}$

- **Stabilité :** Si l'entrée est bornée, la sortie l'est aussi

$$|x_n| < \infty \implies |y_n| < \infty$$

- Exemple : élévateur au carré :  $y_n = (x_n)^2$

$$\text{si } |x_n| < L_x < \infty \text{ alors } |x_n| < (L_x)^2 < \infty$$

- Contre exemple : Log :  $y_n = \log_{10}(|x_n|)$

même si  $|x_n| < \infty$ ; pour  $x_n = 0$  alors  $y_n = -\infty$ .

# Caractérisation d'un SLIT

- SLIT = Système Linéaire Invariant dans le Temps
- Caractériser un SLIT = Trouver la relation sortie / entrée
- Domaine temporel = 2 types de caractérisation
  - ① Réponse impulsionnelle.
  - ② Équation récurrente.
- Domaine Transformé = 2 types de caractérisation
  - ① Réponse fréquentielle.
  - ② Fonction de transfert en Z.

# Domaine temporel : réponse impulsionnelle

- Dans ce cas, la relation sortie/entrée est définie par

$$y_n = x_n \circledast h_n = \sum_{k=-\infty}^{\infty} x_k h_{n-k}$$

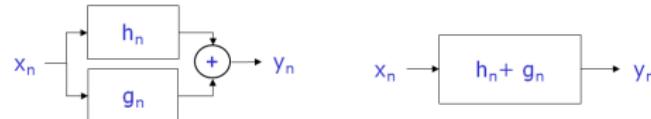
- Propriétés

- Commutativité

$$y_n = x_n \circledast h_n = \sum_{k=-\infty}^{\infty} x_k h_{n-k} = \sum_{k=-\infty}^{\infty} h_k x_{n-k} = h_n \circledast x_n$$

- Distributivité

$$\begin{aligned} y_n &= x_n \circledast h_n + x_n \circledast g_n = \sum_{k=-\infty}^{\infty} x_k h_{n-k} + \sum_{k=-\infty}^{\infty} x_k g_{n-k} \\ &= \sum_{k=-\infty}^{\infty} x_k [h_{n-k} + g_{n-k}] = x_n \circledast [h_n + g_n] \end{aligned}$$



- Mise en cascade



## Réponse impulsionnelle : exemple de calcul

$$x_n = \begin{cases} 1 & \text{pour } 0 \leq n \leq 4 \\ 0 & \text{ailleurs} \end{cases} \quad h_n = \begin{cases} a^n & \text{pour } 0 \leq n \leq 6 \\ 0 & \text{ailleurs} \end{cases}$$

$$y_n = \sum_{k=-\infty}^{\infty} x_k h_{n-k} = \sum_{k=0}^4 x_k h_{n-k}$$

$$n < 0 \quad y_n = 0$$

$$0 \leq n \leq 4 \quad y_n = \sum_{k=0}^n a^{n-k} = \frac{1-a^{n+1}}{1-a}$$

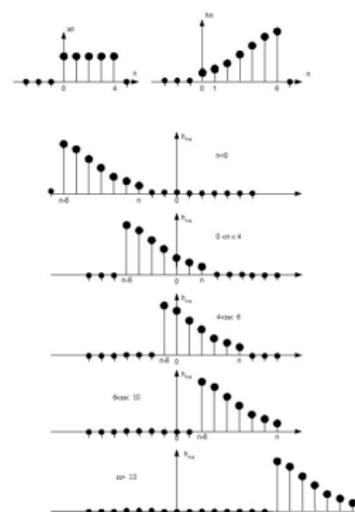
$$4 \leq n \leq 6 \quad y_n = \sum_{k=0}^4 a^{n-k} = a^n \frac{1-a^{-5}}{1-a^{-1}}$$

$$6 \leq n \leq 10 \quad y_n = \sum_{k=n-6}^4 a^{n-k}$$

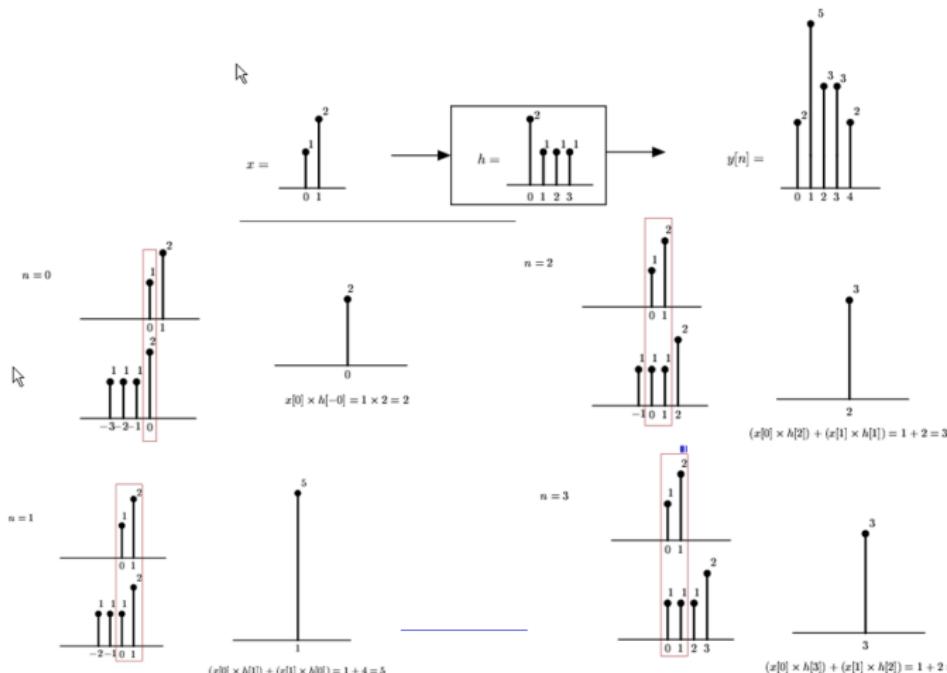
en posant  $r = k - n + 6$

$$6 \leq n \leq 10 \quad y_n = \sum_{r=0}^{10-n} a^{6-r} = a^6 \frac{1-a^{n-11}}{1-a^{-1}}$$

$$10 < n \quad y_n = 0$$



# Réponse impulsionnelle : 2eme exemple de calcul



# Domaine temporel : Equation récurrente

- Dans le cas d'équation récurrente, la relation sortie/entrée est :

$$\sum_{k=0}^N a_k y_{n-k} = \sum_{m=0}^M b_m x_{n-m}$$

- Ce qui également peut s'écrire (en supposant  $a_0=1$ )

$$y_n = \sum_{m=0}^M b_m x_{n-m} - \sum_{k=1}^N a_k y_{n-k}$$

- Exemple de calcul

$$y_n = x_n + \frac{1}{2}y_{n-1} - \frac{1}{4}y_{n-2} \quad \text{avec} \quad x_n = \delta_n$$

- $n = 0 : \quad y_0 = 1 + \frac{1}{2}y_{-1} - \frac{1}{4}y_{-2} = 1 \quad \text{système causal}$
- $n = 1 : \quad y_1 = 0 + \frac{1}{2}y_0 - \frac{1}{4}y_{-1} = \frac{1}{2}$
- $n = 2 : \quad y_2 = 0 + \frac{1}{2}y_1 - \frac{1}{4}y_0 = 0$

# Domaine transformé : Réponse Fréquentielle (RF)

- Dans ce cas, la relation sortie/entrée s'obtient par le calcul de la TF de la RI

$$y_n = x_n \circledast h_n \implies Y(f) = X(f)H(f) \quad \text{ou} \quad H(f) = \frac{Y(f)}{X(f)}$$

$H(f) = RF =$  le rapport entre  $Y(f)$  (TF sortie) et  $X(f)$  (TF entrée).

- Pour  $x_n = \delta_n$  on a  $y_n = h_n$  et donc

$$Y(f) = H(f)$$

La RF  $H(f)$  peut également être définie comme étant la transformée de Fourier de la réponse impulsionnelle  $h_n$ .

# Réponse Fréquentielle : Exemple de calcul

- Le signal d'entrée

$$x_n = \left(\frac{1}{2}\right)^n \Gamma_n$$

- la réponse impulsionnelle

$$h_n = \left(\frac{1}{4}\right)^n \Gamma_n$$

- Utilisation du produit fréquentiel  $Y(f) = X(f)H(f)$  (+simple)

$$\begin{aligned} Y(f) &= \frac{1}{1 - \frac{1}{2} \exp(-2j\pi f)} \frac{1}{1 - \frac{1}{4} \exp(-2j\pi f)} \\ &= \frac{2}{1 - \frac{1}{2} \exp(-2j\pi f)} - \frac{1}{1 - \frac{1}{4} \exp(-2j\pi f)} \end{aligned}$$

- La sortie temporelle

$$y_n = \left(2 \left(\frac{1}{2}\right)^n - \left(\frac{1}{4}\right)^n\right) \Gamma_n$$

# Domaine transformé : Fonction de transfert en Z (FTZ)

- la TZ de l'équation récurrente

$$\sum_{k=0}^N a_k y_{n-k} = \sum_{m=0}^M b_m x_{n-m} \implies \sum_{k=0}^N a_k Y(z) z^{-k} = \sum_{m=0}^M b_m X(z) z^{-m}$$

- La FTZ (qui relie donc la sortie à l'entrée)

$$Y(z) = \frac{\sum_{m=0}^M b_m z^{-m}}{1 + \sum_{k=1}^N a_k z^{-k}} X(z) \implies \frac{Y(z)}{X(z)} = \frac{\sum_{m=0}^M b_m z^{-m}}{1 + \sum_{k=1}^N a_k z^{-k}}$$

- Remarques :
  - L'ordre d'un SLIT = l'ordre de sa fonction de transfert.
  - L'ordre d'une fonction de transfert = le degré le plus élevé du dénominateur.

# FTZ : Exemple de calcul

- La FTZ

$$\frac{Y(z)}{X(z)} = \frac{1}{1 - \frac{1}{2}z^{-1}}$$

- Le signal d'entrée

$$x_n = \Gamma_n \quad \text{soit} \quad X(z) = \frac{1}{1 - z^{-1}}$$

- La sortie dans le domaine transformé

$$Y(z) = \frac{1}{1 - \frac{1}{2}z^{-1}} \frac{1}{1 - z^{-1}} = -\frac{1}{1 - \frac{1}{2}z^{-1}} + \frac{2}{1 - z^{-1}}$$

- La sortie temporelle

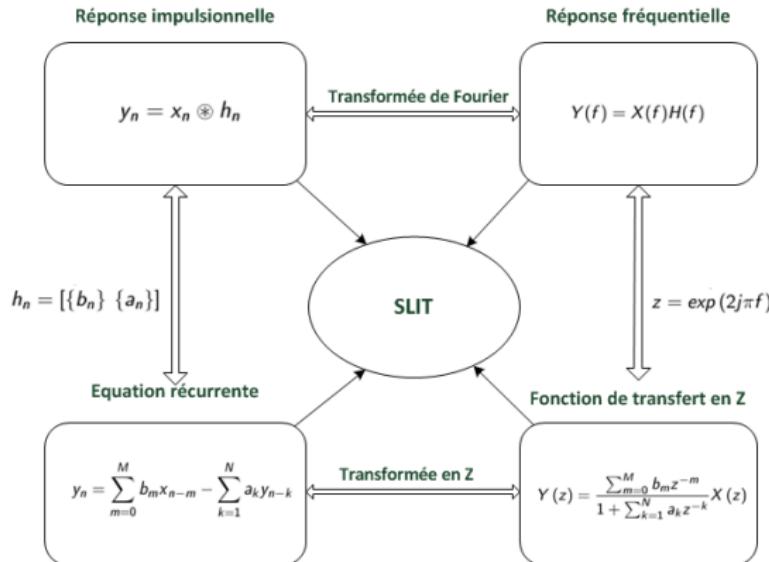
$$y_n = \left[ -\left(\frac{1}{2}\right)^n + 2 \right] \Gamma_n$$

- Remarque : la récurrence peut également être utilisée.

# Récapitulatif

重點

- Un SLIT est caractérisé (peut être représenté) par sa réponse impulsionnelle, son équation récurrente, sa fonction de transfert en Z ou sa réponse fréquentielle.
- Ces différentes représentations sont équivalentes



# Introduction au filtrage numérique

- Introduire le filtrage numérique.
- Synthétiser un filtre numérique non récursif (FIR : Finite Impulse Response).
- Synthétiser un filtre numérique récursif (IIR : Infinite Impulse Response)
- Utiliser les filtres numériques synthétisés dans des applications de traitement de signaux réels.

# Filtre Linéaire

- Filtre linéaire = Système linéaire Invariant dans le Temps (SLIT)



- Peut donc être caractérisé par l'une ou l'autre des représentations d'un SLIT : réponse impulsionnelle, fonction de transfert en Z, réponse fréquentielle ou équation récurrente.

# Types de filtres numériques

2 grandes classes de filtres numériques

- **Les filtres non récursifs** ou filtres FIR (Finite Impulse Response)
  - Leur réponse impulsionnelle s'annule au bout d'un certain temps (nombre fini d'éléments).
  - La sortie de ces filtres ne dépend que de l'entrée (entrée courante et entrées précédentes).

$$y_n = \sum_{i=0}^N b_i x_{n-i} = b_0 x_n + b_1 x_{n-1} + b_2 x_{n-2} \dots b_N x_{n-N}$$

- Leur FTZ (ou transmittance) n'admet pas de pôles

$$Y(z) = \sum_{i=0}^N b_i z^{-i} X(z) \implies \frac{Y(z)}{X(z)} = \sum_{i=0}^N b_i z^{-i}$$

- Ils sont donc stables.
- Ils n'ont pas d'équivalents analogiques.

# Types de filtres numériques

2 grandes classes de filtres numériques

- **Les filtres récursifs** ou filtres IIR (Infinite Impulse Response)
  - Leur réponse impulsionnelle ne s'annule pas (nombre infini d'éléments).
  - La sortie de ces filtres dépend de l'entrée (entrée courante et entrées précédentes) et des sorties précédentes.

$$y_n = \sum_{i=0}^N b_i x_{n-i} + \sum_{i=1}^M a_i y_{n-i}$$

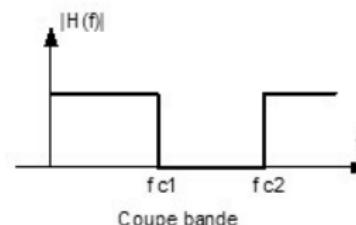
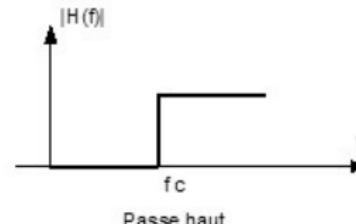
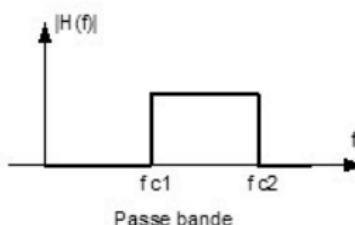
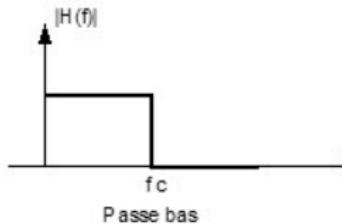
- Leur transmittance admet des pôles

$$Y(z) = \sum_{i=0}^N b_i z^{-i} X(z) + \sum_{i=1}^M a_i z^{-i} Y(z) \implies \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^N b_i z^{-i}}{1 - \sum_{i=1}^M a_i z^{-i}}$$

- Le problème de stabilité se pose.
- Sont synthétisables à partir de gabarits analogiques.

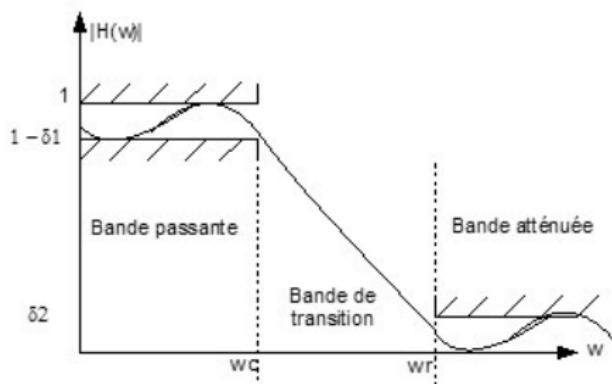
# Spécification d'un filtre

- **Filtrage** = processus permettant de modifier les caractéristiques (amplitude, contenu fréquentiel) du signal d'entrée.
- 4 filtres idéaux de base



# Spécification du gabarit réel d'un filtre passe-bas

Les spécifications portent sur



- la bande passante : fréquences de coupures,
- la bande de transition : (l'ordre du filtre),
- les ondulations en bande passante et en bande atténuee.

# Techniques d'approximation pour le calcul d'un FIR

Plusieurs méthodes. Les plus usuelles :

- Méthode des fenêtres
  - Méthode très simple (présentation et mise en oeuvre)
  - non optimale et donne des filtres de longueur plus importante.
- Approximation "equiripple"
  - Méthode itérative très utilisée.
  - Donne un filtre de longueur minimal satisfaisant au gabarit spécifié
  - En pratique, nécessite un logiciel (matlab ou autre)
- Moindres carrés (et moindres carrés contraints)

# Méthode des fenêtres pour le calcul de filtres RIF

## Approche générale

- ① Se donner le gabarit idéal souhaité  $D(f)$  (périodique)
- ② Par TF inverse calculer les coefficients  $d_i$  de la réponse impulsionnelle (de longueur infinie)
- ③ Tronquer la réponse impulsionnelle en la multipliant par une fenêtre  $w_i$  de longueur N

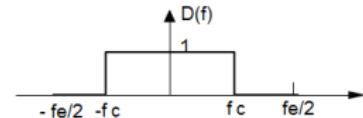
Attention :  $h_i = d_i \cdot w_i \implies H(f) = D(f) * W(f)$

le spectre recherché est donc modifié par le spectre de la fenêtre.

- ④ Rendre causal le filtre par décalage des coefficients.

# Méthode des fenêtres pour le calcul de filtres RIF : Exemple

- Se donner un gabarit : passe-bas idéal



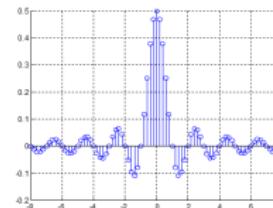
- Calculer la TF inverse :  $D(f)$  périodique  $\rightarrow$  décomposable en série de Fourier

$$D(f) = \sum_{n=-\infty}^{\infty} d_n \exp(-2j\pi fn\Delta) \quad \text{soit} \quad d_n = \frac{1}{f_e} \int_{-\frac{f_e}{2}}^{\frac{f_e}{2}} D(f) \exp\left(+2j\pi \frac{f}{f_e} n\right) df$$

ce qui donne  $d_n = \frac{1}{\pi n} \sin\left(2\pi n \frac{f_c}{f_e}\right)$  avec  $n = -\infty, \dots, 0, \dots \infty$

Illustration pour un demi-bande

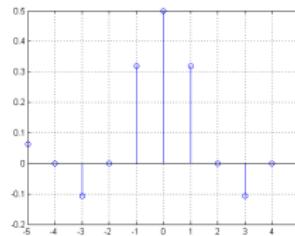
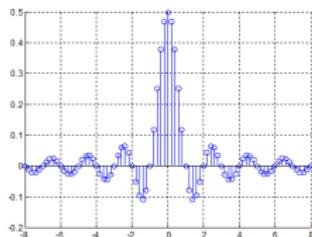
$$f_c = f_e/4$$



# Méthode des fenêtres pour le calcul de filtres RIF : Exemple

- ③ Tronquer la réponse impulsionnelle par fenêtrage (fenêtre rectangulaire avec  $2N+1=11$ )

$$h_n = d_n w_n = \frac{1}{\pi n} \sin\left(\frac{\pi n}{2}\right) \quad \text{avec} \quad n = -5, \dots, 0, \dots, 5$$



La troncature par une fenêtre engendre une modification du spectre

$$h_n = d_n \cdot w_n \implies H(f) = D(f) * W(f)$$

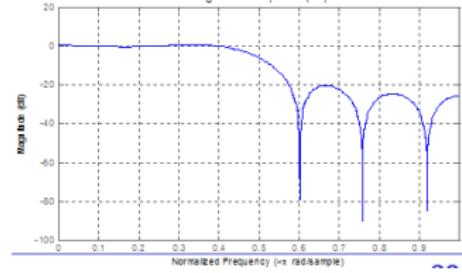
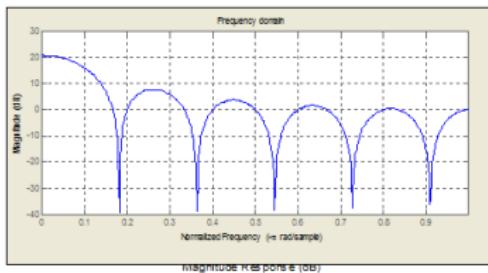
# Méthode des fenêtres pour le calcul de filtres RIF : Exemple

Exemple de Fenêtres :

- Fenêtre rectangulaire

$$w_n = \mathbf{1}_{\{-N, N\}}[n] \implies W(f) = \frac{\sin(\pi f(2N+1))}{\sin(\pi f)} \exp(-2j\pi fN)$$

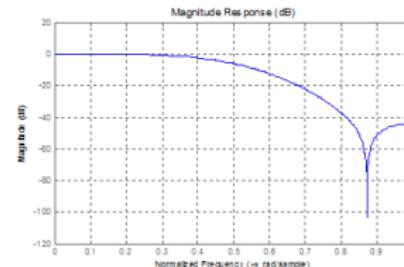
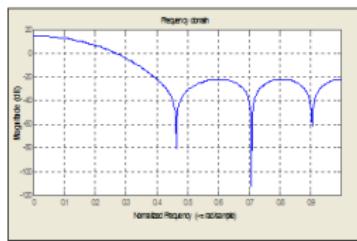
- Illustration dans le cas du demi-bande



- Fenêtre de Hamming

$$w_n = \left[ 0.54 - 0.46 \cos \left( 2\pi \frac{n+N}{2N+1} \right) \right] 1_{\{-N,N\}} [n]$$

- Illustration dans le cas du demi-bande

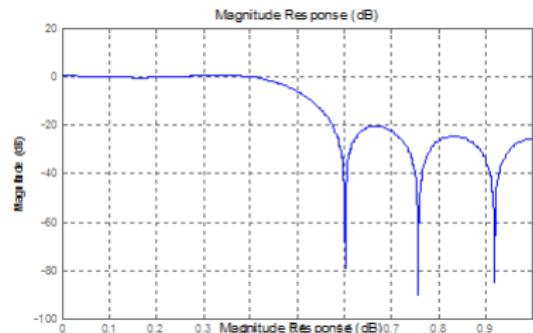


- Autres fenêtres usuelles

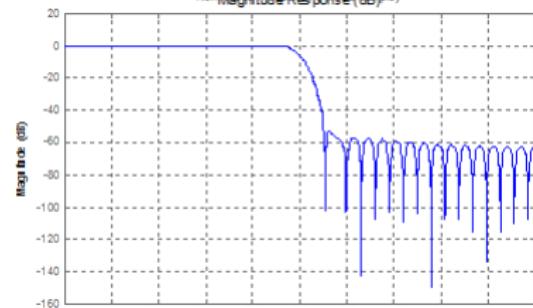
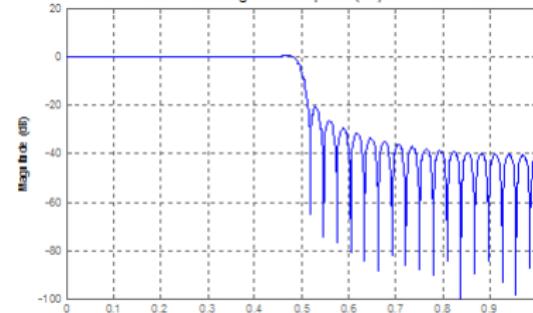
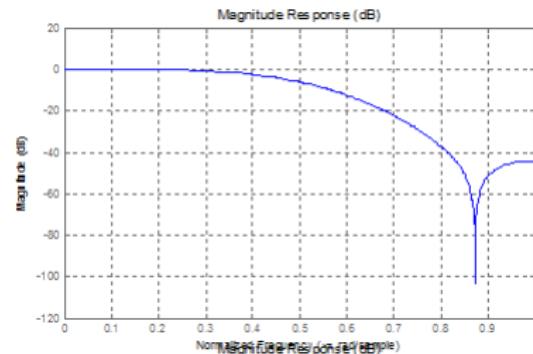
- Fenêtre de Hann      $w_n = \left[ 0.5 - 0.5 \cos \left( 2\pi \frac{n+N}{2N+1} \right) \right] 1_{\{-N,N\}} [n]$
- Fenêtre triangulaire      $w_n = \frac{N+1-|n|}{N+1} 1_{\{-N,N\}} [n]$

# Méthode des fenêtres pour le calcul de filtres RIF : Exemple

Fenêtre rectangulaire longueur 11 et 67



Fenêtre de Hamming longueur 11 et 67



# Méthode des fenêtres pour le calcul de filtres RIF : Exemple

## ④ Rendre le filtre causal (réel)

- Retarder  $h_i$  de  $(N)$
- Conservation de la symétrie des coefficients
- Aucune incidence sur le module de la réponse fréquentielle
- Fonction de transfert en  $Z$  obtenue directement par simple lecture des coefficients obtenus

$$H(z) = \frac{1}{5\pi} - \frac{1}{3\pi}z^{-2} + \frac{1}{\pi}z^{-4} + \frac{1}{2}z^{-5} + \frac{1}{\pi}z^{-6} - \frac{1}{3\pi}z^{-8} + \frac{1}{5\pi}z^{-10}$$

# Approximation "equiripple"

- Principe : chercher à minimiser l'erreur pondérée entre la réponse souhaitée et la réponse réelle

$$e(\omega) = p(\omega) [H_s(\omega) - H_r(\omega)]$$

- "equiripple" = fluctuations d'erreur de même amplitude sont considérées dans le processus de minimisation.
- Méthode itérative.
- Implémentation par algorithme de Parks McClellan (nécessite l'algorithme de Remez).
- donne des filtres de longueur minimale.
- Nécessite un logiciel.

# FIR : Avantages et Inconvénients

- ① Stabilité assurée
- ② Pas de propagation du bruit de quantification (pas de récursivité)
- ③ Phase Linéaire du fait de la symétrie de la fenêtre de troncature

$$\phi(f) = -(N - 1)\pi\Delta f$$

- ④ Temps de propagation de groupe constant (temps mis par le filtre pour traiter un échantillon) = la dérivée de la phase % à la pulsation.

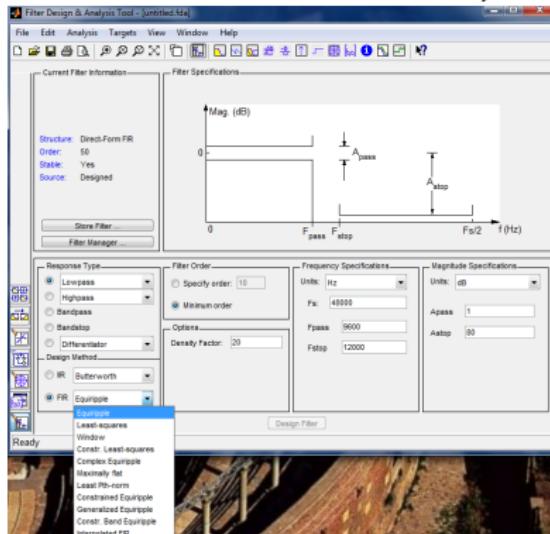
$$tpg = \frac{(N - 1)}{2}\Delta$$

- ⑤ Ordre plus élevé que le IIR pour un gabarit fixé
- ⑥ Plus de mémoire
- ⑦ Plus de calculs

# Synthèse de filtres non récursifs par Matlab

2 approches

- fdatool (déjà vu dans le cas des filtres récursifs)



- En ligne de commande : dépend de la méthode d'approximation choisie

# Synthèse de filtres non récursifs par Matlab : basée fenêtre

3 étapes

- Choisir la longueur L du filtre FIR.
- Choisir la fenêtre de troncature et calculer ses coefficients
  - $W = \text{hamming}(L);$
  - $W = \text{hanning}(L);$
  - ....
- Calculer les coefficients du filtre
  - $b = \text{fir1}(L, W_n);$
  - $b = \text{fir1}(L, W_n, \text{window});$
  - ....

# Synthèse de filtres non récursifs par Matlab : basée equiripple

2 étapes

- Estimer la longueur du filtre à partir des spécifications.
  - $[N, fpts, mag, wt] = \text{remezord}(\text{fedge}, \text{mval}, \text{dev})$  ;
  - $[N, Wn, beta, ftype] = \text{kaiserord}(\text{fedge}, \text{mval}, \text{dev})$  à actualiser
- Calculer les coefficients de la FTZ
  - $b = \text{remez}(N, fpts, mag)$  ;
  - $b = \text{remez}(N, fpts, mag, wt)$  ; ;
  - ....

## Démarche

3 étapes pour le calcul d'un filtre IIR (récursif)

- **Spécification** : Trouver, en fonction de l'application, l'ensemble de contraintes que la réponse fréquentielle (amplitude et phase) du filtre numérique à concevoir doit satisfaire.
- **Approximation de la spécification par un système discret** : Déterminer, à partir du gabarit analogique choisi, les coefficients de la fonction de transfert générale du filtre numérique équivalent.
- **Implantation** : Choix, en fonction de la cible, de la meilleure structure et réalisation.

## 1 - Spécification

Les spécifications portent sur le gabarit

- la bande de transition (l'ordre du filtre)
  - les fréquences de coupures
  - les ondulations en BP et/ou en BA

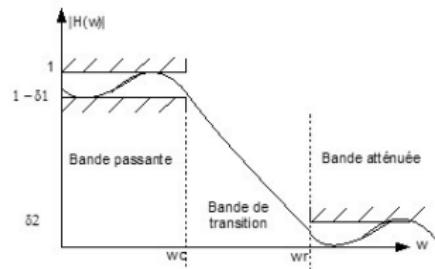
## Exemple : Passe-bas

$$1 - \delta_1 \leq |H(\omega)| \leq 1 \quad |\omega| \leq \omega_p$$

$$|H(\omega)| \leq \delta_2 \quad |\omega| \leq \omega_a$$

## Specs fréquentielles :

- $\delta_1 = 0.01$  = ondulation en BP
  - $\delta_2 = 0.01$  = ondulation en BA
  - $\omega_p = 2\pi(2000)$ ;  $\omega_a = 2\pi(3000)$



## Specs Gain en dB :

- Gain en BP =  $20\log(1) = 0\text{dB}$
  - Gain min en BP =  $20\log(0.99)$
  - Gain en BA=  $20\log(0.01)=-40\text{dB}$

## 2 - Approximation : Démarche

### Principe général

- 1 Définir le gabarit d'un filtre analogique  $H_a(\omega)$ . En général

$$|H_a(\omega)|^2 = \frac{1}{1 + F_a^{(n)}(\omega^2)}$$

$n$  = ordre du filtre et  $F_a(\omega)$  = fonction dépendante du gabarit.

- 2 Déterminer la fonction de transfert en  $s$  correspondante  $H_a(s)$
- 3 Déterminer la fonction de transfert discrète  $H_n(z)$  équivalente au sens d'une transformée  $f(s)$  appropriée.

$$|H_a(\omega)|^2 \quad \Rightarrow \quad H_a(s) H_a(-s) \quad \Rightarrow \quad H_n(z)$$

$$s = j\omega \qquad \qquad \qquad z = f(s)$$

# Approximation : Choix de gabarits

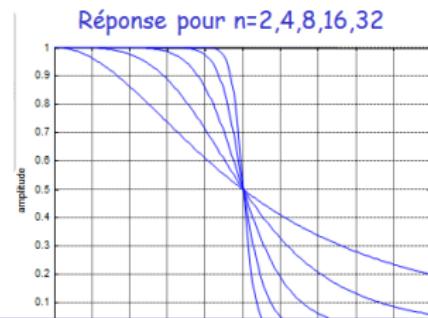
## 4 gabarits usuels

### Butterworth

- réponse fréquentielle plate en BP et BA (+)
- large bande de transition (-)
- décroissance monotone (Asymptote de -6N dB par octave)

$$F_a^{(n)}(\omega^2) = \left(\frac{\omega}{\omega_c}\right)^{2n} \Rightarrow |H_a(\omega)|^2 = \frac{1}{1+\left(\frac{\omega}{\omega_c}\right)^{2n}}$$

$$\lim_{n \rightarrow \infty} |H_a(\omega)|^2 = \begin{cases} 1 & \text{si } |\omega| < \omega_c \\ 0.5 & \text{si } |\omega| = \omega_c \\ 0 & \text{si } |\omega| > \omega_c \end{cases}$$



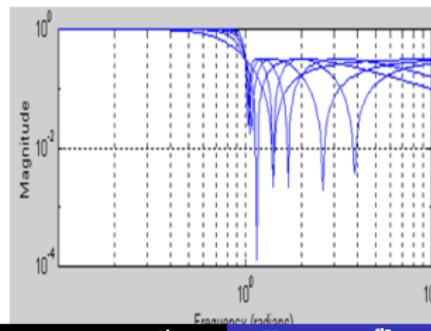
# Approximation : Choix de gabarits (suite)

## Chebyshev II

- Ondulation en BA (inverse de Chebyshev I)
- Réponse fréquentielle plate en BP
- Décroissance monotone pour  $w < w_c$

$$F_a^{(n)}(\omega^2) = \left[ \varepsilon^2 T_N^2\left(\frac{\omega}{\omega_c}\right) \right]^{-1} \Rightarrow |H_a(\omega)|^2 = \frac{1}{1 + [\varepsilon^2 T_N^2\left(\frac{\omega}{\omega_c}\right)]^{-1}}$$

Le facteur d'ondulation  $\varepsilon$  donne également le compromis entre l'ondulation et la largeur de transition



# Approximation : Choix de gabarits (suite et fin)

## Elliptique

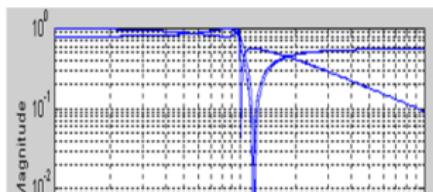
- Appelé aussi filtre de Cauer
- Ondulation en BP et en BA

$$|H(\omega)|^2 = \frac{1}{1 + \epsilon^2 U_N^2(\omega)}$$

avec  $U_N^2(\omega) = \begin{cases} \prod_{i=1}^m \frac{\omega_i^2 - \omega^2}{1 - \omega^2 \omega_i^2} & \text{pour } N = 2m \\ \omega \prod_{i=1}^m \frac{\omega_i^2 - \omega^2}{1 - \omega^2 \omega_i^2} & \text{pour } N = 2m + 1 \end{cases}$

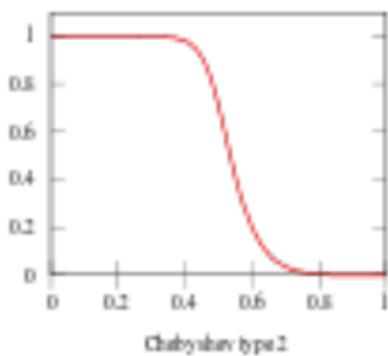
$U_N(\omega)$  = fonction elliptique de Jacobi d'ordre N (tabulée dans tous les softs)

$\epsilon$  = facteur d'ondulation

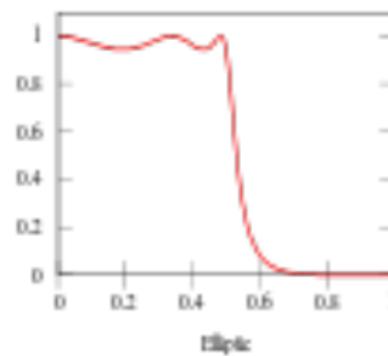


## Récapitulatif

Butterworth



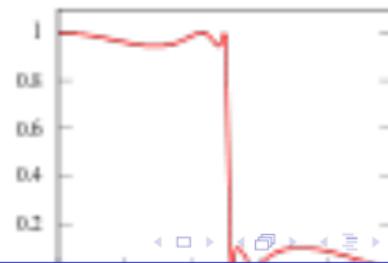
Chabyshev type I



Chabyshev type 2



Elliptic



# Approximation : Fonction de transfert en s

- ➊ Définir le gabarit d'un filtre analogique  $H_a(\omega)$ . En général

$$|H_a(\omega)|^2 = \frac{1}{1 + F_a^{(n)}(\omega^2)}$$

$n$  = ordre du filtre et  $F_a(\omega)$  = fonction dépendante du gabarit.

- ➋ Déterminer la fonction de transfert en s correspondante  $H_a(s)$
- ➌ Déterminer la fonction de transfert discrète  $H_n(z)$  équivalente au sens d'une transformée  $f(s)$  appropriée.

$$\begin{array}{ccc} s = j\omega & & z = f(s) \\ |H_a(\omega)|^2 & \xrightarrow{\quad} & H_a(s) H_a(-s) \\ & & \xrightarrow{\quad} H_n(z) \end{array}$$

Cette deuxième étape nécessite une factorisation spectrale

## • Factorisation spectrale : Principe

- Calculer les pôles et les zéros puis factoriser le numérateur et dénominateur de  $H_a(s) H_a(-s)$ .
- Définir  $H_a(s)$  comme la partie de  $H_a(s) H_a(-s)$  qui contient les pôles situés dans le demi-plan gauche (pour la stabilité) et les zéros du demi-plan gauche (pour la phase minimum). Ces pôles vont être placés symmétriquement.
- Exemple : déterminer la fonction de transfert d'un Butterworth du second ordre de pulsation de coupure égale à 1
  - Réponse Fréquentielle

$$|H_a(\omega)|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_c}\right)^4}$$

- Domaine de Laplace (plan s)

$$H_a(s) H_a(-s) = \frac{1}{1 + \left(\frac{s}{j\omega_c}\right)^4}$$

- Calcul des pôles :

$$(s_k)^4 = (-1) (j^2)^2 (\omega_c)^4 = (-1) (1) (\omega_c)^4 \quad \text{soit} \quad s_k = (-1)^{\frac{1}{4}} \omega_c = 1 \cdot \exp(j(2k+1)\pi)$$

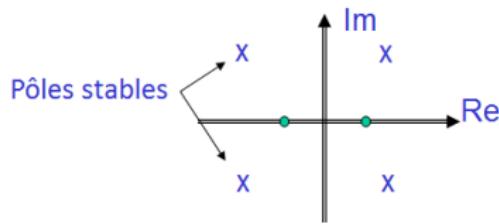
soit 4 pôles d'arguments  $\theta_k = \frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}$  et  $\frac{7\pi}{4}$

- Pôles du demi plan gauche

$$H_a(s) = \frac{1}{(s - 1[\frac{3\pi}{4}]) (s - 1[\frac{5\pi}{4}])} = \frac{1}{\left(s - \left(-\frac{\sqrt{2}}{2} + j\frac{\sqrt{2}}{2}\right)\right) \left(s - \left(-\frac{\sqrt{2}}{2} - j\frac{\sqrt{2}}{2}\right)\right)}$$

finalement

$$H_a(s) = \frac{1}{(s^2 + s\sqrt{2} + 1)}$$



# Approximation : Fonction de transfert en Z

- ➊ Définir le gabarit d'un filtre analogique  $H_a(\omega)$ . En général

$$|H_a(\omega)|^2 = \frac{1}{1 + F_a^{(n)}(\omega^2)}$$

$n$  = ordre du filtre et  $F_a(\omega)$  = fonction dépendante du gabarit.

- ➋ Déterminer la fonction de transfert en  $s$  correspondante  $H_a(s)$
- ➌ **Déterminer la fonction de transfert discrète  $H_n(z)$  équivalente au sens d'une transformée  $f(s)$  appropriée.**

$$|H_a(\omega)|^2 \quad \Rightarrow \quad H_a(s) H_a(-s) \quad \Rightarrow \quad H_n(z)$$

$$s = j\omega \qquad \qquad \qquad z = f(s)$$

# Approximation : Fonction de transfert en Z

- Différentes techniques pour calculer la fonction de transfert d'un IIR à partir de la fonction de transfert du filtre analogique.

$$z = f(s) \\ H_a(s) \implies H_n(z)$$

- Invariance impulsionnelle**
- Dérivation (ou méthode d'Euler)**
- Intégration (ou transformée bilinéaire)**
- Dans la pratique, les filtres sont calculés par des logiciels de synthèse à partir des spécifications souhaitées.

# Equivalence au sens Invariance impulsionnelle

- Le principe consiste à échantillonner la réponse impulsionnelle

$$h_n = h(t)|_{t=n\Delta} \quad \text{où} \quad \Delta = \text{pas d'échantillonnage}$$

- La réponse impulsionnelle continue peut s'écrire

$$H_a(s) = \sum_{k=1}^N \frac{A_k}{(s - s_k)} \Rightarrow h(t) = \sum_{k=1}^N A_k \exp(s_k t) \Gamma(t)$$

- L'échantillonnage de cette RI donne

$$h_n = \sum_{k=1}^N A_k \exp(s_k n \Delta) \Gamma(n \Delta) \Rightarrow H_n(z) = \sum_{n=0}^{\infty} h_n z^{-n} = \sum_{k=1}^N \frac{A_k}{(1 - \exp(s_k \Delta) z^{-1})}$$

- Ce qui se traduit par un changement (subi que par les pôles)

$$\frac{A_k}{(s - s_k)} \longrightarrow \frac{A_k}{(1 - \exp(s_k \Delta) z^{-1})}$$

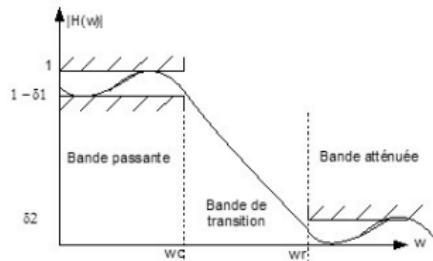
# Exemple

## Spécifications

- $\delta_1 = \delta_2 = 0.2$
- $\omega_c = 0.4\pi ; \omega_r = 0.6\pi$

## Gabarit de butterworth

$$|H_a(\omega)|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}$$



## Système de 2 équations à 2 inconnues

$$|H_a(\omega)|^2 = \frac{1}{1 + \left(\frac{0.4\pi}{\omega_c}\right)^{2n}} = (0.8)^2$$

$$|H_a(\omega)|^2 = \frac{1}{1 + \left(\frac{0.6\pi}{\omega_c}\right)^{2n}} = (0.2)^2$$

## Exemple : suite

- Solution :  $n=5$  et  $\omega_c = 0.43\pi$      $H_a(s) H_a(-s) = \frac{1}{1 + \left(\frac{s}{j0.43\pi}\right)^{10}}$
- les pôles de cette fonction :  $s_k = (1)^{\frac{1}{10}}\omega_c = \omega_c \exp\left(\frac{2jk\pi}{10}\right)$
- Les arguments du  $\frac{1}{2}$  plan gauche :  $\theta_k = \frac{2\pi k}{10} = \frac{3\pi}{5}, \frac{4\pi}{5}, \frac{6\pi}{5}, \frac{7\pi}{5}$
- Transmittance réalisable :

$$H_a(s) = \frac{1}{(s - 0.43\pi \lfloor \frac{3\pi}{5} \rfloor)(s - 0.43\pi \lfloor \frac{4\pi}{5} \rfloor)(s - 0.43\pi \lfloor \frac{\pi}{5} \rfloor)(s - 0.43\pi \lfloor \frac{6\pi}{5} \rfloor)(s - 0.43\pi \lfloor \frac{7\pi}{5} \rfloor)}$$

- Décomposition en éléments simples :

$$H_a(s) = \frac{A}{s-s_1} + \frac{B}{s-s_2} + \frac{C}{s-s_3} + \frac{D}{s-s_4} + \frac{E}{s-s_5}$$

- Transformation des pôles par II :  $\frac{A_k}{(s-s_k)} \longrightarrow \frac{A_k}{(1-\exp(s_k\Delta)z^{-1})}$

- Transmittance en Z :

$$4.44 \left( \frac{0.571}{1 - 0.259z^{-1}} + \frac{-0.486 + 0.277z^{-1}}{1 - 0.471z^{-1} + 0.113z^{-2}} + \frac{-0.486 + 0.277z^{-1}}{1 - 0.378z^{-1} + 0.434z^{-2}} \right)$$

# Equivalence au sens de la dérivation

- Le principe est basé sur le fait qu'une dérivée peut être approchée par la formule d'Euler

$$\left. \frac{dx(t)}{dt} \right|_{t=n\Delta} \approx \frac{x(n\Delta) - x((n-1)\Delta)}{\Delta}$$

- Un SLIT numérique d'entrée  $x_n$  et de sortie  $y_n$  qui réalise cette approximation s'écrit donc

$$y(n\Delta) = \frac{x(n\Delta) - x((n-1)\Delta)}{\Delta} \implies \frac{Y(z)}{X(z)} = \frac{1 - z^{-1}}{\Delta}$$

- Dans le plan  $s$ , la dérivation est  $H(s) = s$ , la fonction de transfert du filtre numérique est alors donnée par

$$H(z) = H(s)|_{s=\frac{1-z^{-1}}{\Delta}}$$

## Equivalence au sens de la dérivation : Exemple

- Synthèse d'un filtre numérique équivalent à un filtre analogique passe bas du premier ordre
  - Premier ordre
  - Gain statique de 1
  - Fréquence de coupure  $5/(2 * \pi)$
  - Fréquence d'échantillonnage 5
- La fonction de transfert analogique s'écrit

$$H(s) = \frac{1}{1 + 0.2s}$$

- La fonction de transfert numérique s'écrit

$$H(z) = H(s)|_{s=\frac{1-z^{-1}}{\triangle}} = \frac{1}{2 - z^{-1}}$$

- L'équation récurrente s'écrit

$$y_t = 0.5u_t + 0.5y_{t-1}$$

# Equivalence au sens de l'intégration

- Le principe est basé sur le fait qu'une intégration peut être approchée par la méthode des trapèzes

$$H(s) = \frac{1}{s} \rightarrow y_t = y_{t-1} + \frac{u_t + u_{t-1}}{2} \Delta$$

$$Y(z) = z^{-1} Y(z) + \frac{\Delta}{2} [U(z) + z^{-1} U(z)] \quad \text{soit} \quad \frac{Y(z)}{U(z)} = \frac{\Delta}{2} \frac{1+z^{-1}}{1-z^{-1}}$$

- La fonction de transfert du filtre numérique est alors donnée par

$$H(z) = H(s)|_{s=\frac{2}{\Delta} \frac{1-z^{-1}}{1+z^{-1}}}$$

- Remarques :

- Relation non linéaire entre les fréquences analogiques et les fréquences numériques.

$$s = \frac{2}{\Delta} \frac{1-z^{-1}}{1+z^{-1}} \implies z = \frac{1+s\Delta/2}{1-s\Delta/2} \implies f_a = \frac{1}{\pi\Delta} \tan(\pi f_n \Delta)$$

# Transformée bilinéaire : Exemple

- Synthèse d'un filtre numérique équivalent à un filtre analogique passe bas du premier ordre
  - premier ordre
  - gain statique de 1
  - fréquence de coupure  $5/(2 * \pi)$
  - fréquence d'échantillonnage 5
- La fonction de transfert analogique s'écrit  $H_a(s) = \frac{1}{1+0.2s}$
- La fonction de transfert numérique s'écrit

$$H_N(z) = H_a(s)|_{s=\frac{2}{\Delta} \frac{1-z^{-1}}{1+z^{-1}}} = \frac{1+z^{-1}}{3-z^{-1}}$$

- L'équation récurrente s'écrit  $y_t = \frac{1}{3}(u_t + u_{t-1} + y_{t-1})$

# Synthèse de filtres récursifs par Matlab

- A partir d'un filtre analogique

- Invariance impulsionale

$$[bz, az] = \text{impinvar}(bs, as, F_e)$$

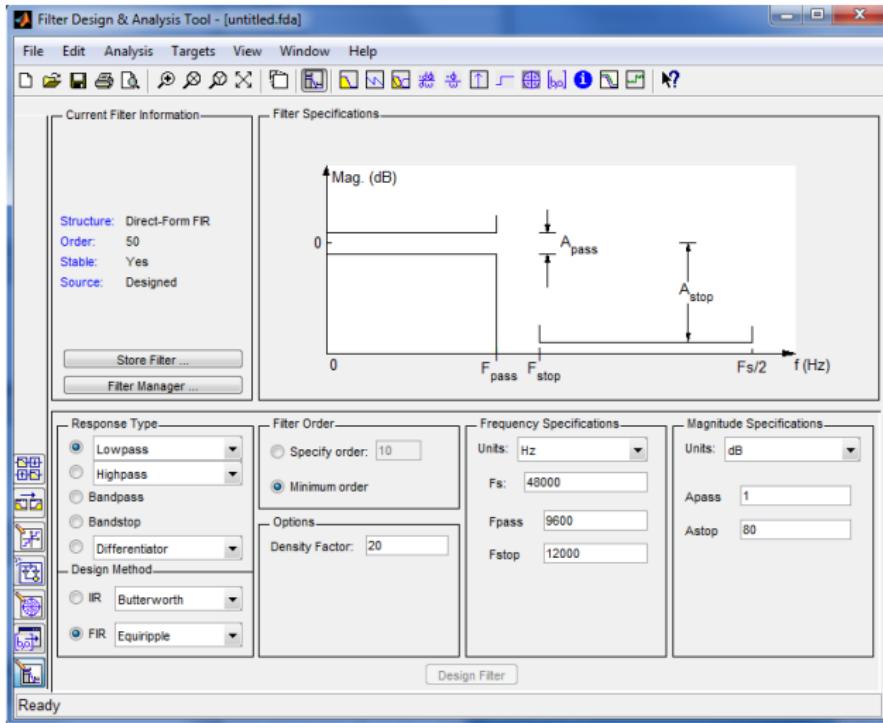
- Transformée bilinéaire

$$[bz, az] = \text{bilinear}(bs, as, F_e)$$

- A partir d'un gabarit

- fdatool (outil de synthèse)
  - En ligne de commande

## fdatool



# Synthèse de filtres récursifs par Matlab : (suite et fin)

## En ligne de commande : 2 étapes

- ➊ Déterminer d'abord l'ordre du filtre et sa pulsation de coupure normalisée
  - $[N, W_n] = \text{buttord}(W_p, W_s, R_p, R_s)$
  - $[N, W_p] = \text{cheb1ord}(W_p, W_s, R_p, R_s)$
  - $[N, W_s] = \text{cheb2ord}(W_p, W_s, R_p, R_s)$
  - $[N, W_p] = \text{ellipord}(W_p, W_s, R_p, R_s)$

- ➋ Calculer les coefficients de la réponse impulsionnelle (ou FTZ)
  - $[b, a] = \text{butter}(N, W_n)$
  - $[b, a] = \text{cheby1}(N, W_n)$
  - $[b, a] = \text{cheby2}(N, R_s, W_s)$
  - $[b, a] = \text{ellip}(N, R_p, R_s, W_p)$

Pour filtrer un signal d'entrée  $x$

$y = \text{filter}(b, a, x)$

# Structures de filtres

- Une structure de filtre est une représentation graphique du flux des signaux.
- Une structure de filtre a deux objectifs : aider à
  - implémenter efficacement le filtre sur un processeur spécifique
  - concevoir une architecture matérielle.
- Une structure est dite canonique si le nombre de retard est égal à l'ordre de la fonction de transfert.
- Un même filtre peut avoir plusieurs structures.

# Structures de filtres

- Plusieurs types de structures

- forme directe,
- forme transposée,
- cascade,
- parallèle,
- ...

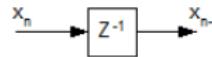
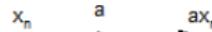
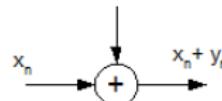
- Ces structures diffèrent

- complexité calculatoire,
- taille de mémoire exigée,
- modularité,
- ...

# Blocs de base

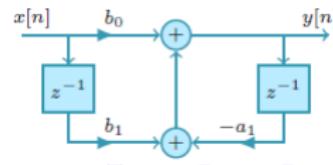
3 blocs de base sont nécessaires pour le filtrage

- Addition
- Multiplication
- Retard



$$\text{Exemple : } H(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}}$$

$$\text{soit } y_n = b_0 x_n + b_1 x_{n-1} - a_1 y_{n-1}$$



# Formes directes

- Equation récurrente d'un IIR = cascade de 2 sous systèmes

- $s_n = \sum_{i=0}^N b_i e_{n-i} + \sum_{i=1}^M a_i s_{n-i} = x_n + \sum_{i=1}^M a_i s_{n-i}$
- $x_n = \sum_{i=0}^N b_i e_{n-i} \Leftrightarrow B(z) = \frac{X(z)}{E(z)} = \sum_{i=0}^N b_i z^{-i}$
- $s_n = x_n + \sum_{i=1}^M a_i s_{n-i} \Leftrightarrow \frac{S(z)}{X(z)} = \frac{1}{A(z)} = \frac{1}{1 - \sum_{i=1}^M a_i z^{-i}}$

- 4 formes directes (utilisation directe des coefficients)

- Direct I :  $B(z)$  suivie de  $\frac{1}{A(z)}$
- Transposed Direct I :  $B(z)$  suivie de  $\frac{1}{A(z)}$  + transposition
- Direct II :  $\frac{1}{A(z)}$  suivie de  $B(z)$
- Transposed Direct II :  $\frac{1}{A(z)}$  suivie de  $B(z)$  + transposition

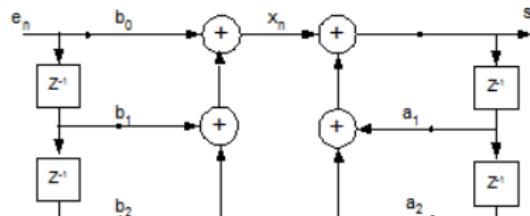
# Structures RII : Forme directe I

## Cellule du second ordre

- Implémentation directe de l'équation récurrente

$$s_n = \sum_{i=0}^2 b_i e_{n-i} + \sum_{i=1}^2 a_i s_{n-i} = x_n + \sum_{i=1}^2 a_i s_{n-i}$$

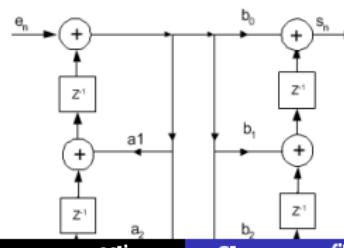
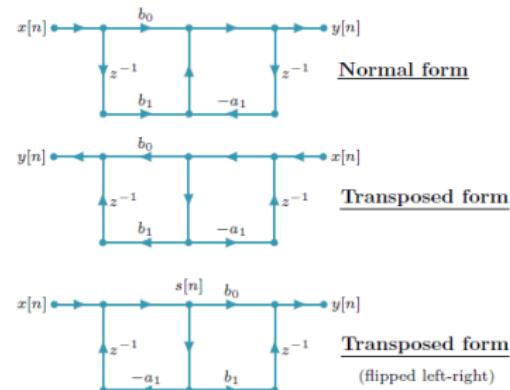
- Implémentation de  $B(z)$  suivie de  $\frac{1}{A(z)}$



- $x_n = b_0 e_n + b_1 e_{n-1} + b_2 e_{n-2}$  et  $s_n = x_n + a_1 s_{n-1} + a_2 s_{n-2}$
- Coût : 4 retards (2N en général), 5 multip. (2N+1) et 4 add. (2N).

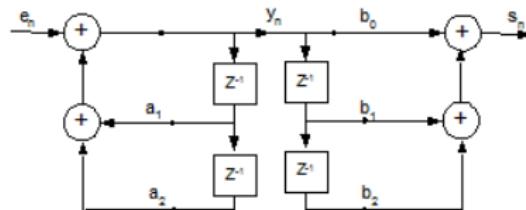
# Structures RII : Forme directe I Transposée

- Inverser la direction de chaque interconnexion
- Inverser la direction de chaque multiplicateur
- Interchanger Jonctions et additionneurs
- Interchanger l'entrée et la sortie

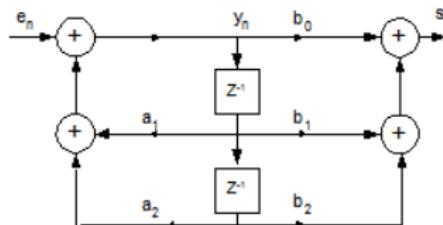


## Structures RII : Forme directe II

- Implémentation de  $\frac{1}{A(z)}$  suivie de  $B(z)$

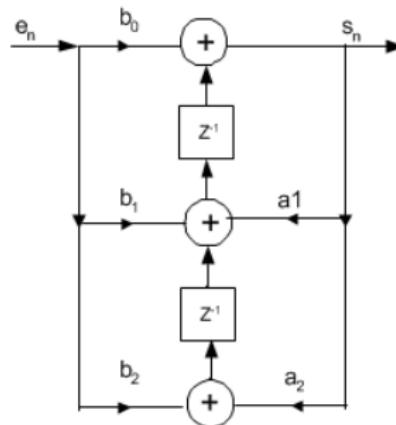


- $y_n = e_n + a_1y_{n-1} + a_2y_{n-2}$  et  $s_n = b_0y_n + b_1y_{n-1} + b_2y_{n-2}$
- Forme canonique (pour éviter de stocker 2 fois la même grandeur)



- Coût : 2 retards (N en général), 5 multipl. ( $2N+1$ ) et 4 add. ( $2N$ )

## Structures RII : Forme directe II transposée



- $s_n = (b_0 e_n + b_1 e_{n-1} + b_2 e_{n-2}) + (a_1 s_{n-1} + a_2 s_{n-2})$
- Coût : 2 retards (N en général), 5 multip. (2N+1) et 4 add. (2N).
- Matlab : structure implémentée par "filter(b,a,x,v)"

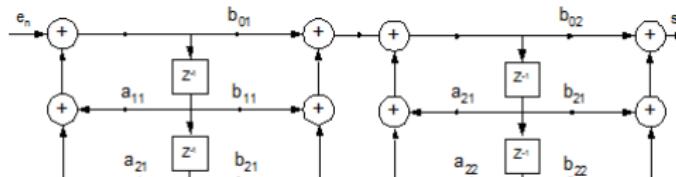
## Structures RII : Forme cascade

- Intéressante pour les fonctions de transfert d'ordre élevé

$$H(z) = \frac{\sum_{i=0}^N b_i z^{-i}}{1 - \sum_{i=0}^N a_i z^{-i}} = \frac{\prod_{i=1}^{N1} (1 - g_i z^{-i}) \prod_{i=1}^{N2} (1 - h_i z^{-i}) (1 - h_i^* z^{-i})}{\prod_{i=1}^{N1} (1 - c_i z^{-i}) \prod_{i=1}^{N2} (1 - d_i z^{-i}) (1 - d_i^* z^{-i})}$$

Avec  $g_i, c_i$  et  $h_i, d_i$  respectivement les zéros et pôles réels et complexes  $N=N1+2N2$ .

- Plusieurs façons de cascader les fonctions de transfert. La plus avantageuse (en termes de stockage et complexité algorithmique) est le produit de cellules du second ordre.
- Exemple du 4<sup>eme</sup> ordre



- Matlab : conversion en cellules 2<sup>eme</sup> ordre "[sos,G] = tf2sos(b,a)"

## Structures RII : Forme Parallèle

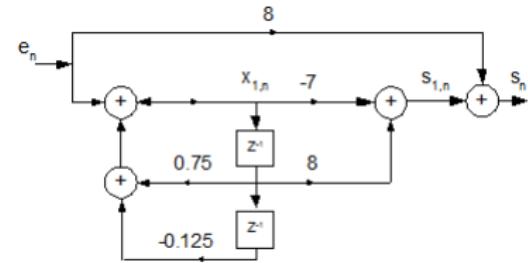
- La fonction de transfert d'un filtre RII peut toujours se décomposer

$$H(z) = \sum_{i=0}^{N-M} C_i z^{-i} + \sum_{i=1}^K \frac{B_{i0} + B_{i1} z^{-1}}{1 + A_{i1} z^{-1} + A_{i2} z^{-2}}$$

- Somme d'un FIR et cellules SOS à implémenter en parallèle.
- Exemple simple

$$H(z) = \frac{1 + 2z^{-1} + z^{-2}}{1 - 0.75z^{-1} + 0.125z^{-2}} = 8 + \frac{-7 + 8z^{-1}}{1 - 0.75z^{-1} + 0.125z^{-2}}$$

- $x_{1,n} = e_n + 0.75x_{n-1} - 0.125x_{1,n-2}$
- $s_{1,n} = -7x_{1,n} + 8x_{1,n-1}$
- $s_n = 8e_n + s_{1,n}$

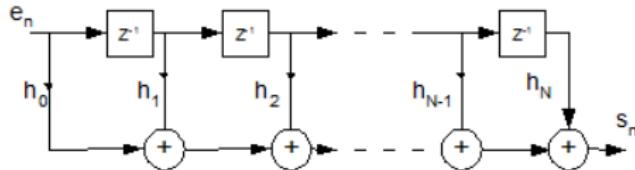


Méthode : voir slide IIresidue-II

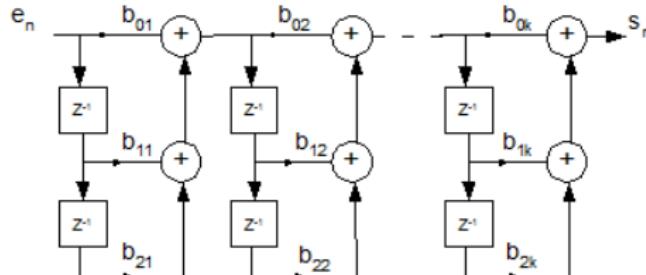
# Structures RIF

- Forme directe : équivalente à celle RII en supprimant les "pôles"

$$H(z) = \sum_{i=0}^N b_i z^{-i} \quad s_n = \sum_{i=0}^N b_i e_{n-i}$$



- Forme directe transposée : idem RII en supprimant les pôles
- Forme cascade : cascader des cellules second ordre



# Structure RIF : Forme spécifique

- exploite la propriété de symétrie des coefficients

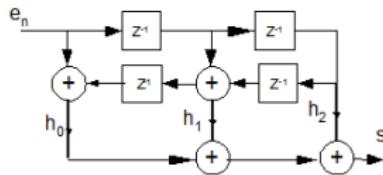
$$h_n = h_{N-n} \quad \text{pour } 0 \leq n \leq N$$

- Dépendant de N (pair ou impair)  $\implies$  2 structures
- N impair

$$s_n = \sum_{i=0}^N h_i e_{n-i} = \sum_{i=0}^{(N/2)-1} h_i e_{n-i} + h_{\frac{N}{2}} e_{n-\frac{N}{2}} + \sum_{i=\frac{N}{2}+1}^N h_i e_{n-i}$$

- en posant  $i = N - k$

$$s_n = \sum_{i=0}^{(N/2)-1} h_i e_{n-i} + h_{\frac{N}{2}} e_{n-\frac{N}{2}} + \sum_{k=0}^{\frac{N}{2}-1} h_{N-k} e_{n-N+k} = \sum_{i=0}^{(N/2)-1} h_i (e_{n-i} + e_{n-N+i}) + h_{\frac{N}{2}} e_{n-\frac{N}{2}}$$



- Coût  $\cdot \frac{N}{2} + 1$  multiplications au lieu de N pour la structure directe