# Traitement avancé du signal et des images
# Partie Signal

Sébastien Adam
Cours de Licence 3 EEEA-INFO
2022-2023

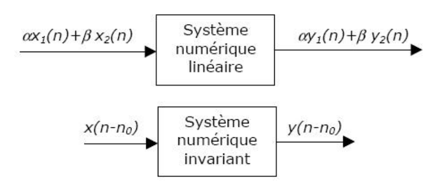
# Plan

# Systèmes et filtres numériques

## Définitions

- Un système numérique est un système qui reçoit en entrée une séquence de nombres $x(n)$ et produit en sortie une séquence de nombres $y(n)$. $x$ est appelé signal d'entrée, ou signal d'excitation. $y$ est appelé signal de sortie ou signal de réponse


- Un filtre numérique est un système numérique particulier, qui est linéaire, invariant et stable



# Systèmes et filtres numériques

## Exemples : analyser les propriétés des systèmes suivants :

- $y(n) = Kx(n) + 1$
- $y(n) = x(n-1)$
- $y(n) = nx(n)$
- $y(n) = |x(n)|^2$

# Systèmes et filtres numériques

## Exemples : analyser les propriétés des systèmes suivants :

- $y(n) = Kx(n) + 1 \rightarrow$ Non linéaire, Invariant et Stable
- $y(n) = x(n-1)$
- $y(n) = nx(n)$
- $y(n) = |x(n)|^2$

# Systèmes et filtres numériques

**Exemples : analyser les propriétés des systèmes suivants :**

- $y(n) = Kx(n) + 1 \rightarrow$ Non linéaire, Invariant et Stable
- $y(n) = x(n-1) \rightarrow$ Linéaire, Invariant et Stable
- $y(n) = nx(n)$
- $y(n) = |x(n)|^2$

# Systèmes et filtres numériques

## Exemples : analyser les propriétés des systèmes suivants :

- $y(n) = Kx(n) + 1 \rightarrow$ Non linéaire, Invariant et Stable
- $y(n) = x(n-1) \rightarrow$ Linéaire, Invariant et Stable
- $y(n) = nx(n) \rightarrow$ Linéaire, Non Invariant, et Instable
- $y(n) = |x(n)|^2$

# Systèmes et filtres numériques

## Exemples : analyser les propriétés des systèmes suivants :

- $y(n) = Kx(n) + 1 \rightarrow$ Non linéaire, Invariant et Stable
- $y(n) = x(n-1) \rightarrow$ Linéaire, Invariant et Stable
- $y(n) = nx(n) \rightarrow$ Linéaire, Non Invariant, et Instable
- $y(n) = |x(n)|^2 \rightarrow$ Non Linéaire, Invariant, et Stable

# Systèmes et filtres numériques

## Représentation des filtres numériques

- On peut représenter un SLI par une équation aux différences finies (equivalent de l'équation différentielle)

  $y(n) + a_1 y(n-1) + ... + a_N y(n-N) = b_0 x(n) + b_1 x(n-1) + ... + b_M x(n-M)$

- La valeur de la sortie $y(n)$ à l'instant courant s'exprime alors comme une combinaison linéaire des $N$ sorties précédentes, de l'entrée courante, et des $M$ entrées précédentes.
- Écrit sous forme un peu plus compacte, cela donne :

  $$y(n) + \sum_{i=1}^{N} a_i y(n-i) = \sum_{i=0}^{M} b_i x(n-i)$$

- On parle de filtres récursifs
- Version non récursive : les $a_i$ sont nuls

  $$y(n) = \sum_{i=0}^{M} b_i x(n-i)$$

# Systèmes et filtres numériques

## Représentation graphique des filtres numériques

- On peut visualiser graphiquement l'équation de récurrence associée à un système numérique en faisant apparaitre trois éléments de base :

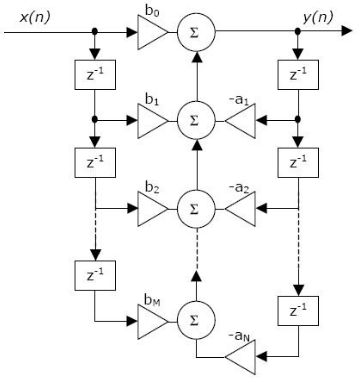
| | |
|---|---|
| un additionneur, qui somme les signaux d'entrée | $\Sigma$ |
| un multiplieur, qui multiplie les signaux d'entrée | a |
| un élément délai, qui retarde d'un échantillon | $z^{-1}$ |

- On construit ainsi le graphe de fluence du système.

# Systèmes et filtres numériques

## Graphe de fluence générique d'un SLI

$$y(n) + \sum_{i=1}^{N} a_i y(n-i) = \sum_{i=0}^{M} b_i x(n-i)$$



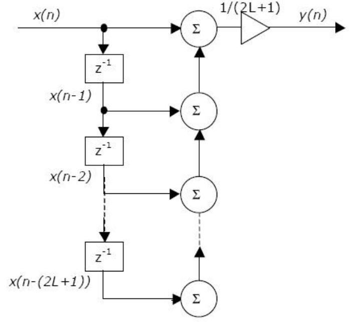
# Systèmes et filtres numériques

## Exemple de graphe de fluence

Un filtre à moyenne mobile est un filtre non causal calculant la moyenne du signal estimée sur 2L+1 valeurs autour de l'échantillon courant

$$y(n) = \frac{1}{2L+1} \sum_{i=-L}^{L} x(n-i)$$

- Les $a_i$ d'un tel filtre sont tous nuls, les $b_i$ valent $1/(2L+1)$
- Il y a en plus un retard sur la sortie
- Un tel filtre est non récursif puisque $y(n)$ ne dépend pas de $y(n-k)$



# Systèmes et filtres numériques

## Exercice

Considérons un signal donnant l'état de votre compte bancaire à chaque fin de mois.

- $y(n)$ est l'état du compte à l'instant $n$,
- $x(n)$ le versement à l'instant $n$,
- $p$ est le taux d'intérêt

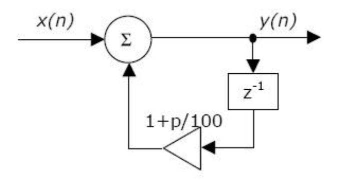
Donner :

- La forme analytique du signal $y(n)$
- Le graphe de fluence d'un tel système
- Les 5 premières valeurs de y

# Systèmes et filtres numériques

## Exercice

$$y(n) = y(n-1) + \frac{p}{100}y(n-1) + x(n)$$




- On a $a_1 = -\frac{1+p}{100}$ et $b_0 = 1$
- Un tel filtre est récursif puisque $y(n)$ dépend de $y(n-1)$

# Systèmes et filtres numériques

## Exercice

- A partir de l'équation aux différences, on peut calculer les $y(n)$
- Remarque : c'est fait par la fonction `filter` sous octave/matlab. Le $a_0$ correspond alors au $y(n)$ : $a_0 = 1$
- Exemple avec p=5 ;

$$y(n) = y(n-1) + \frac{p}{100} y(n-1) + x(n)$$

```
% script de test de la fonction filter sur
% l'exemple compte bancaire

a=[1 -1.05];
b=[1];
x=[100 0 0 0 0 0];
y=filter(b,a,x)

>> exfilter
y =
  100.0000   105.0000   110.2500   115.7625   121.5506   127.6282
```

# Systèmes et filtres numériques

## Autres représentations des SLI

Pour caractériser les filtres, on peut utiliser l'équation aux différences mais aussi

- sa réponse impulsionnelle
- sa fonction de transfert en z

Réponse impulsionnelle d'un filtre

- C'est la réponse $y(n)$ obtenue lorsque l'on met une impulsion de Dirac à l'entrée du filtre
- On la note généralement $h(n)$

# Systèmes et filtres numériques

## Exemple de réponse impulsionnelle

Dans le cas de l'exemple "filtre à Moyenne Mobile". La réponse impulsionnelle est la réponse à une impulsion, on peut l'obtenir par filter

$$y(n) = \frac{1}{2L+1}\sum_{i=-L}^{L} x(n-i) \rightarrow h(n) = \left\{\frac{1}{2L+1}, \frac{1}{2L+1}, \ldots, 0, 0\right\}$$

```
>> input=[1,zeros(1,49)];
>> b=1/7*ones(1,7);
>> repimpfil=filter(b,[1],input)
 Columns 1 through 8:
   0.14286    0.14286    0.14286    0.14286    0.14286    0.14286    0.14286    0
 Columns 9 through 13:
   0.00000    0.00000    0.00000    0.00000    0.00000
```

Lorsque le SLI est non récursif, la RI est alors une séquence limitée de valeurs non nulles qui sont les $b_i$ de l'équation aux différences. On parle alors de filtre à Réponse Impulsionnelle Finie (RIF - FIR)

# Systèmes et filtres numériques

## Exemple de réponse impulsionnelle

Dans le cas de l'exemple "Comptes bancaires"

$y(n) = y(n-1) + \frac{p}{100}y(n-1) + x(n)$

$\rightarrow h(n) = \{1, 1 + p/100, (1 + p/100)^2, (1 + p/100)^3, ...\}$

```
>> a=[1 -1.05];
>> b=[1];
>> x=[1 zeros(1,49)];
>> y=filter(b,a,x)
 Columns 1 through 8:
   1.0000   1.0500   1.1025   1.1576   1.2155   1.2763   1.3401   1.4071
 Columns 9 through 13:
   1.4775   1.5513   1.6289   1.7103   1.7959\end{footnotesize}
```

Lorsque le SLI est récursif, la réponse impulsionnelle est alors une séquence illimitée de valeurs non nulles. On parle alors de filtre à Réponse Impulsionnelle Infinie (RII - IIR)

# Systèmes et filtres numériques

## Stabilité des SLI

On peut évaluer la stabilité d'un SLI en fonction de sa réponse impulsionnelle : Soit $h(n)$ la réponse impulsionnelle d'un SLI, on peut montrer que la condition de stabilité du système s'écrit :

$$\sum_{k=-\infty}^{+\infty} |h(k)| < +\infty$$

- Exemple :

$$h(n) = \{1, 1 + p/100, (1 + p/100)^2, (1 + p/100)^3, ...\}$$

- Les coefficients tendent vers l'infini : le système est instable

# Systèmes et filtres numériques

## Réponse à une entrée quelconque

On peut exploiter la réponse impulsionnelle pour calculer la sortie du filtre pour une entrée quelconque. Pour cela, on exploite le fait que tout signal numérique peut s'écrire comme une somme d'impulsions pondérées et décalées.

$$x(n) = \{x(0), x(1), ...\} = x(0)\delta(n) + x(1)\delta(n-1) + x(2)\delta(n-2) + ...$$

On a alors par linéarité et invariance :

$$\begin{aligned} y(n) &= x(0)h(n) + x(1)h(n-1) + x(2)h(n-2) + ... \\ &= x(n) * h(n) = \sum_{i=-\infty}^{+\infty} x(i)h(n-i) \end{aligned}$$

# Systèmes et filtres numériques

## Convolution numérique

Un produit de convolution numérique est défini par :

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

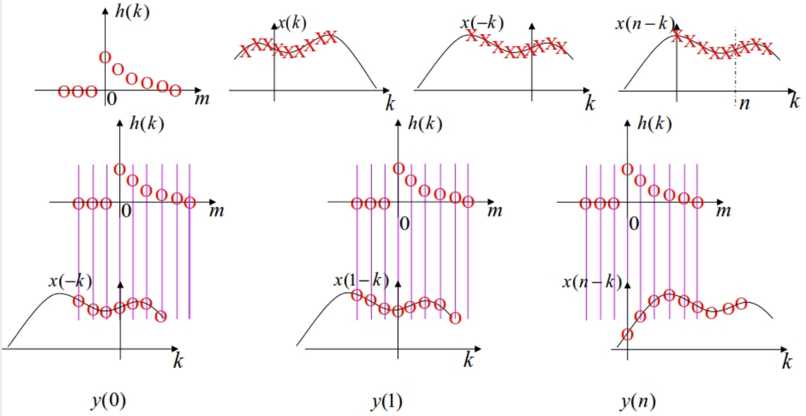
La convolution est commutative. Ainsi, le produit de convolution peut aussi être écrit

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

Pour obtenir une valeur $y(n_0)$, il suffit d'inverser $h(n)$, de positionner $h(0)$ sur $x(n_0)$ et de calculer le produit scalaire entre les séquence $h(n)$ et $x(n)$ ainsi définies : $y(n_0)$ est une combinaison linéaire des valeurs de $x(n)$ autour de $x(n_0)$, avec comme coefficient les valeurs de $h(n)$.

# Systèmes et filtres numériques

## Exemple de convolution numérique



# Systèmes et filtres numériques

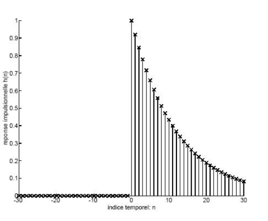
## Exemple de convolution numérique

Soit la réponse impulsionnelle $h(n)$ d'un système linéaire invariant telle que :

$$h(n) = \begin{cases} a^n & \text{si } n \geq 0 \\ 0 & \text{si } n < 0 \end{cases}$$

C'est à dire

$$h(n) = a^n u(n)$$



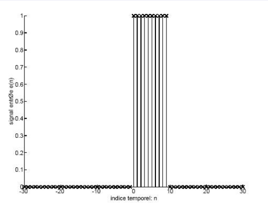
# Systèmes et filtres numériques

## Exemple de convolution numérique

On met en entrée de ce système un échelon définit par :

$$e(n) = u(n) - u(n - N) \text{ pour } N \text{ fixé}$$

Pour $N = 10$ on a donc :



# Systèmes et filtres numériques

## Exemple de convolution numérique

La sortie s'obtient alors en calculant :

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

On peut alors distinguer plusieurs configurations

- si $n < 0$, $h(n-k)$ et $e(n)$ n'ont aucun échantillon en commun : $s(n) = 0$
- si $n \in [0, N[$, on a recouvrement d'échantillons non nuls pour $k \in [0, n]$, d'où :

$$y(n) = \sum_{k=0}^{n} a^{n-k} = a^n \frac{1-a^{-(n+1)}}{1-a^-1}$$

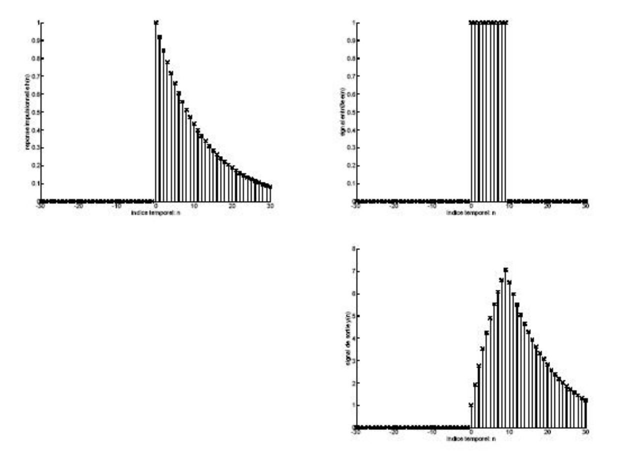
- si $n \geq N$, on a recouvrement d'échantillons non nuls pour $k \in [0, N]$, d'où

$$y(n) = \sum_{k=0}^{N-1} a^{n-k} = a^n \frac{1-a^{-N}}{1-a^-1}$$

# Systèmes et filtres numériques

## Exemple de convolution numérique

Allure de la réponse :



# Systèmes et filtres numériques

## Convolution numérique et produit de polynômes

- L'opération de convolution peut également être vue comme celle du produit de deux polynômes.
- Soient deux séquences $x(n) = \{1, 2, 3, 0, 0, ...\}$ et $h(n) = \{2, 3, 4, 0, 0, ...\}$
- $x(n) * h(n) = \{2, 7, 16, 17, 12, 0, 0, ...\}$ pour $n \geq 0$
- Si on pose : $X(z) = 1 + 2z^{-1} + 3z^{-2} = z^{-2}(z^2 + 2z + 3)$
- et $H(z) = 2 + 3z^{-1} + 4z^{-2} = z^{-2}(2z^2 + 3z + 4)$
- On a alors : $Y(z) = X(z) \times H(z) = z^{-4}(2z^4 + 7z^3 + 16z^2 + 17z + 12)$

$\rightarrow$ C'est l'intérêt de la transformée en Z qui permet de transformer un produit de convolution en simple produit.