US332S Conception sonore

Audio numérique

Cours 2 Dynamique et filtrage

Matthias Puech
`matthias.puech@lecnam.net`

Master 1 JMIN — Cnam ENJMIN, Angoulême

19 mars 2019

Audio Numérique

Traitements audio

- Graphe de flot de signal

Traitement de la dynamique

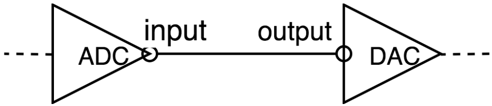
- Principe
- Compresseur, limiteur, gate
- Enveloppe d'amplitude

Filtrage

- Une première approximation
- Typologie des filtres
- Caractérisations
- Systèmes linéaires
- Etude de cas

Rappels

- un signal numérique est un flot de nombres
- … qui approximent un signal analogique
  (temps discret, valeurs discrètes)
- nous n'avons vu qu'un traitement trivial "*bypass*"
  (conversion numérique → conversion analogique)



Que peut-on faire entre input et output ?

Tous les 1/44100$^{\text{ème}}$ de seconde (F_s), le processeur :

- reçoit un sample de input (une valeur numérique)
- effectue des opérations sur cette valeur
- renvoit un sample vers output (une valeur numérique)

1. attention : vision extrêmement simplifiée

Que peut-on faire entre input et output ?

"From signal to symphony"

Tous traitement audio-numérique est expressible en terme :

- opérations arithmétiques (+, -, ×, /, $\sqrt{}$)
- comparaison de valeurs (=, <, >, ≤)
- lecture/écriture en mémoire

Chaque instruction prend un cycle de l'horloge du processeur [1]

1. attention : vision extrêmement simplifiée

Que peut-on faire entre input et output ?

"From signal to symphony"

Tous traitement audio-numérique est expressible en terme :

- opérations arithmétiques (+, -, ×, /, $\sqrt{\ }$)
- comparaison de valeurs (=, $<$, $>$, $\leq$)
- lecture/écriture en mémoire

Chaque instruction prend un cycle de l'horloge du processeur[1]

Exemple (traitement en temps réel)

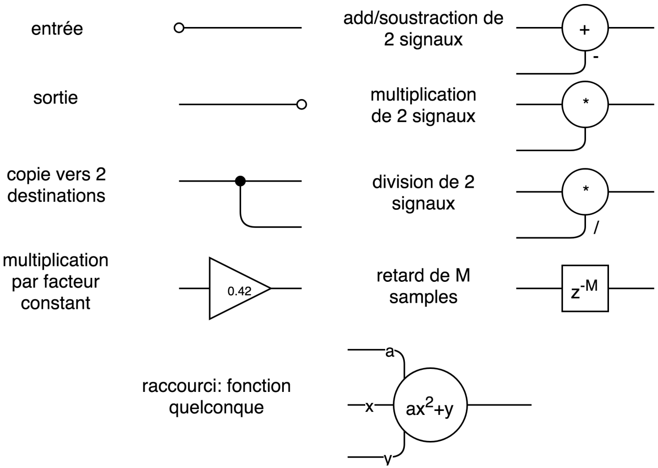
À 1GHz et avec $F_s = 44100$Hz, le processeur peut faire ≈ 10000 opérations avant de devoir produire un sample de sortie.

1. attention : vision extrêmement simplifiée

Graphe de flot de signal

Plutôt que d'écrire chaque instruction, représentons un traitement par un réseau d'opérateurs basiques, reliées par des fils

Les briques de base



Échauffement : le *crossfade*

Exercice

Ecrire le graphe de flot pour un *crossfader*, qui mixe deux signaux audio x et y selon la valeur d'un signal c :

- si $c = 0$, on entendra que x
- si $c = 1$, on entendra que y
- si $c = 0.5$ on entendra x à 50% et y à 50%

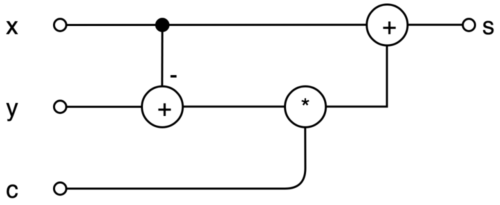
Échauffement : le *crossfade*

Exercice

Ecrire le graphe de flot pour un *crossfader*, qui mixe deux signaux audio x et y selon la valeur d'un signal c :

- si $c = 0$, on entendra que x
- si $c = 1$, on entendra que y
- si $c = 0.5$ on entendra x à 50% et y à 50%

Solution



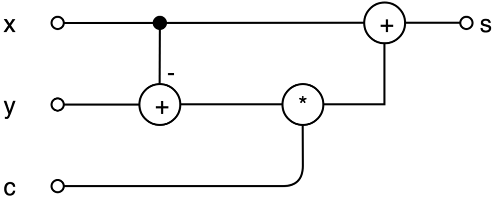
Échauffement : le *crossfade*

Exercice

Ecrire le graphe de flot pour un *crossfader*, qui mixe deux signaux audio x et y selon la valeur d'un signal c :

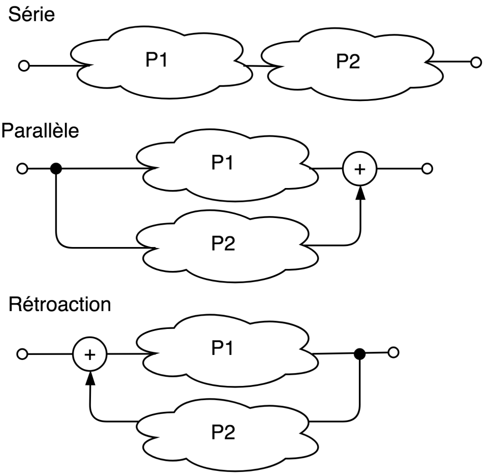
- si $c = 0$, on entendra que x
- si $c = 1$, on entendra que y
- si $c = 0.5$ on entendra x à 50% et y à 50%

Solution

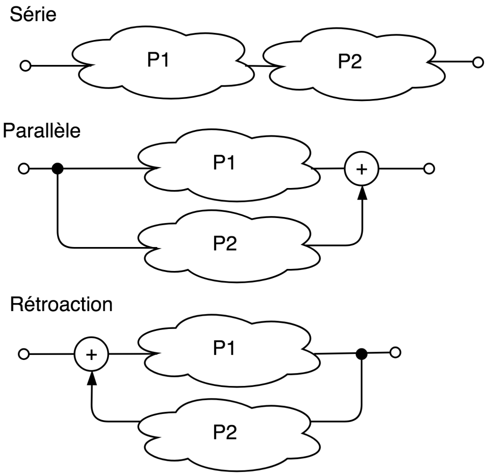


$$s[n] = x[n] + c[n](y[n] - x[n])$$

Graphes récurrents



Graphes récurrents



Causalité

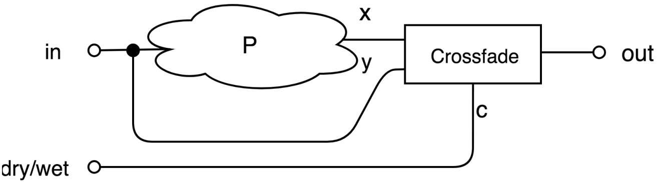
Un graphe n'est réalisable que s'il est *causal* : si le calcul d'un sample $s[n]$ ne dépend pas de sa valeur. Attention à la rétroaction !

Graphes récurrents

Exemple : Contrôle Dry/Wet

Graphes récurrents

Exemple : Contrôle Dry/Wet



Audio Numérique

Traitements audio
- Graphe de flot de signal

Traitement de la dynamique
- Principe
- Compresseur, limiteur, gate
- Enveloppe d'amplitude

Filtrage
- Une première approximation
- Typologie des filtres
- Caractérisations
- Systèmes linéaires
- Etude de cas

Traitement de la dynamique *(Dynamic Range Processing)*

Famille de traitements qui modifient la plage dynamique d'un signal d'entrée $s[n]$ en modifiant dynamiquement son amplitude en fonction de son enveloppe d'amplitude $a_s[n]$.

Traitement de la dynamique *(Dynamic Range Processing)*

Famille de traitements qui modifient la plage dynamique d'un signal d'entrée $s[n]$ en modifiant dynamiquement son amplitude en fonction de son enveloppe d'amplitude $a_s[n]$.

plage dynamique
: rapport (en dB) entre les moments les plus silencieux et les plus forts
(cf. rapport signal/bruit)

modifier dynamiquement l'amplitude
: = multiplier $s[n]$ par un signal $a_s[n]$ qui représente son *enveloppe d'amplitude*
(tourner le bouton de volume)

enveloppe d'amplitude
: $a_s[n]$ donne pour chaque instant n le volume perçu de $s[n]$
(de 0 = silence à 1 = volume maximum)

Traitement de la dynamique *(Dynamic Range Processing)*

Applications

compresseur uniformise le volume perçu d'un signal
(monte le son dans les passage silencieux et/ou le baisse dans les passage forts)

expanseur accentue la plage dynamique
(monte le son dans les passage forts)

limiteur baisse le son quand le volume passe au-dessus du maximum (-1… 1)
(pour éviter l'ecrêtage)

noise gate réduire au silence des passages de bruit de fond
(coupe le son quand le volume est en dessous d'un seuil)

compander paire compresseur/expanseur inverses l'un de l'autre
(utilisé pour réduire le bruit de fond à l'enregistrement/transmission, *e.g.* Dolby B/C)

Traitement de la dynamique *(Dynamic Range Processing)*

Paramètres

courbe de transition statique associe à un volume d'entrée (0..1) un volume de sortie ($0..\infty$)

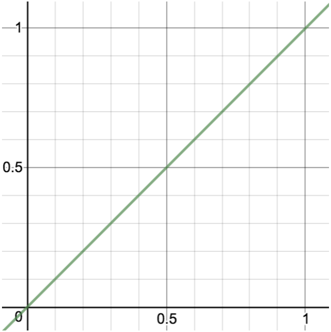
algorithme de mesure du volume plusieurs façon de le calculer (le plus souvent : *Peak* ou *Root-Mean-Square*)

attack/release temps de transition mis par la mesure de volume pour monter/descendre de sa valeur précédente à la nouvelle

lookahead petit temps d'avance de la mesure de niveau (permet d'anticiper les transitoires brutales)

Traitement de la dynamique *(Dynamic Range Processing)*

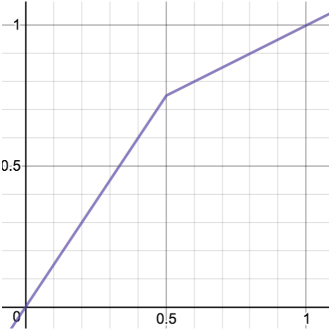
Courbes de transition



Bypass
(pas de changement)

Traitement de la dynamique *(Dynamic Range Processing)*

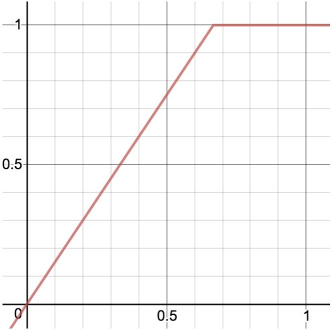
Courbes de transition



Compresseur
(ici seuil=0.5, ratio=1.6)

Traitement de la dynamique *(Dynamic Range Processing)*

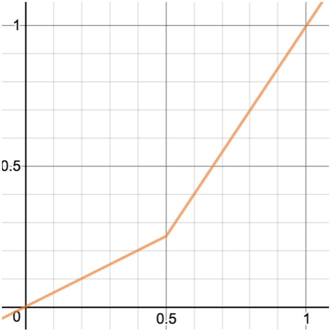
Courbes de transition



Limiteur
(ici limite=0.65)

Traitement de la dynamique *(Dynamic Range Processing)*

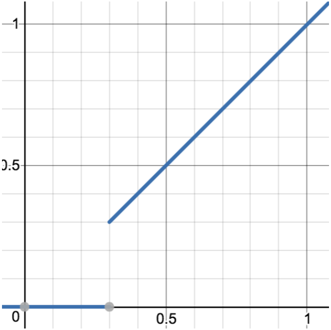
Courbes de transition



Expanseur
(ici seuil=0.5, ratio=0.7)

Traitement de la dynamique *(Dynamic Range Processing)*

Courbes de transition

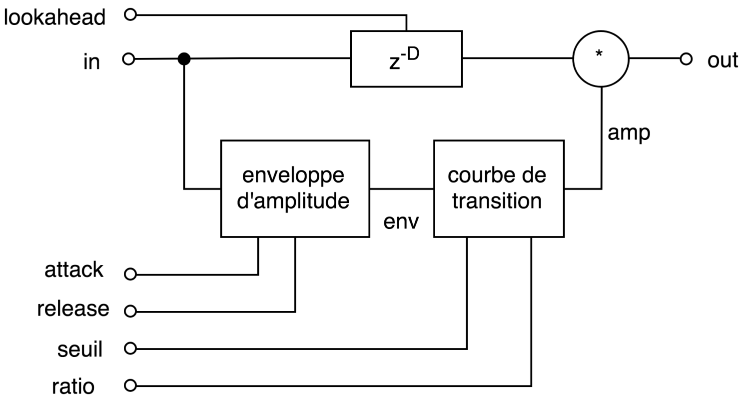


Noise gate
(ici seuil=0.3)

Traitement de la dynamique *(Dynamic Range Processing)*

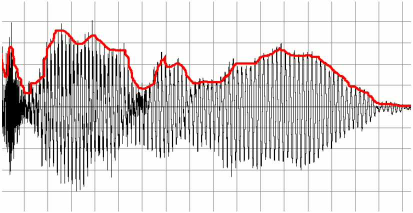
DEMO [`08.dynamics`]

Graphe de flot



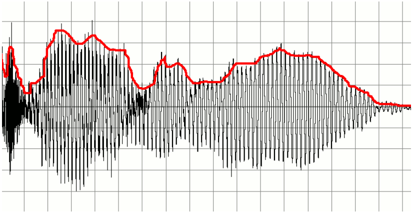
Enveloppe d'amplitude

C'est le signal qui donne le niveau sonore perçu par l'oreille à chaque instant d'un signal audio



Enveloppe d'amplitude

C'est le signal qui donne le niveau sonore perçu par l'oreille à chaque instant d'un signal audio



Remarques

- compris entre 0 (silence) et 1 (volume maximum)
  (pas de valeurs négatives)
- suit les pics locaux du signal, mais pas ses oscillations du domaine audible (>20Hz)
- paramètre : comment et combien l'enveloppe "suit" le signal
  (attack/release, RMS/Peak)

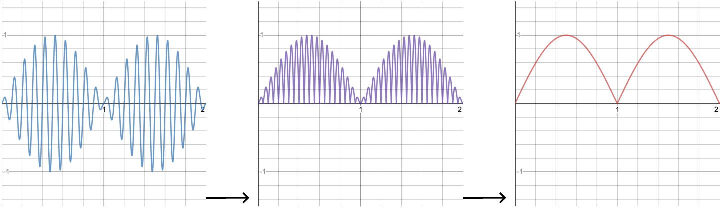
Enveloppe d’amplitude

Implementation

On calcule l’enveloppe d’amplitude en deux étapes :

rectification élimine la partie inférieure du signal (de -1…1 vers 0…1)

lissage élimine les variations rapides (>20Hz) du signal rectifié



Rectification

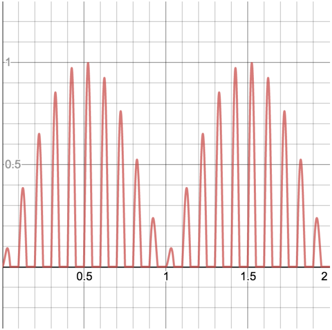
Transforme un signal $s[n]$ centré sur zéro entre -1 et 1
en un signal $r[n]$ entre 0 et 1, centré autour de l'amplitude de $s[n]$.
Plusieurs implémentations possibles :

Rectification

Transforme un signal $s[n]$ centré sur zéro entre -1 et 1
en un signal $r[n]$ entre 0 et 1, centré autour de l'amplitude de $s[n]$.
Plusieurs implémentations possibles :

Demi-rectification

$r[n] = \{s[n] \text{ si } s[n] > 0,\ 0 \text{ sinon}\}$



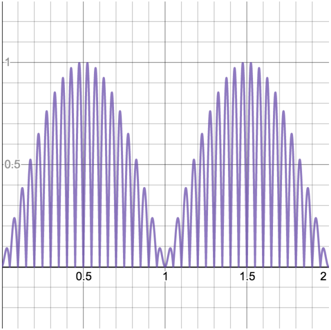
(simple à mettre en oeuvre en électronique)

Rectification

Transforme un signal $s[n]$ centré sur zéro entre -1 et 1
en un signal $r[n]$ entre 0 et 1, centré autour de l'amplitude de $s[n]$.
Plusieurs implémentations possibles :

Rectification totale

$r[n] = |s[n]|$ (valeur absolue)



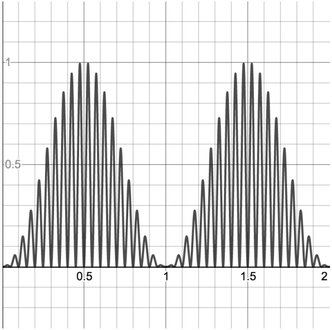
(largement utilisé, base de la détection "*peak*")

Rectification

Transforme un signal $s[n]$ centré sur zéro entre -1 et 1
en un signal $r[n]$ entre 0 et 1, centré autour de l'amplitude de $s[n]$.
Plusieurs implémentations possibles :

Fonction carré

$r[n] = s[n]^2$



(base de la détection "*RMS*" ; compensé plus tard par une racine carrée)

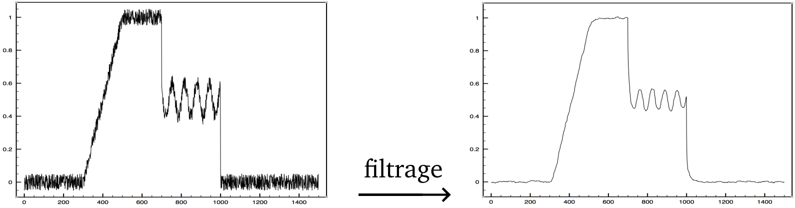
Lissage

Élimine mouvements brusques & oscillations rapides du signal rectifié $r[n]$ pour ne garder qu'un “mouvement général” $l[n]$

Lissage

Élimine mouvements brusques & oscillations rapides du signal rectifié $r[n]$ pour ne garder qu'un "mouvement général" $l[n]$

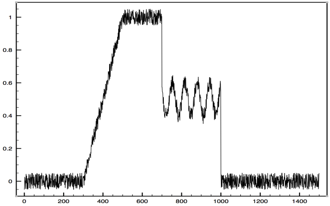
$=$ *filtrage passe-bas* (voir plus loin)



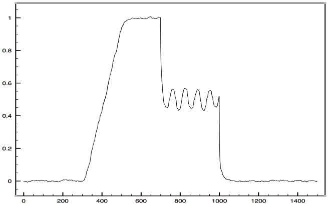
Lissage

Élimine mouvements brusques & oscillations rapides du signal rectifié $r[n]$ pour ne garder qu'un "mouvement général" $l[n]$

$=$ *filtrage passe-bas* (voir plus loin)



filtrage $\longrightarrow$



Plusieurs implémentations possibles :

Moyenne glissante

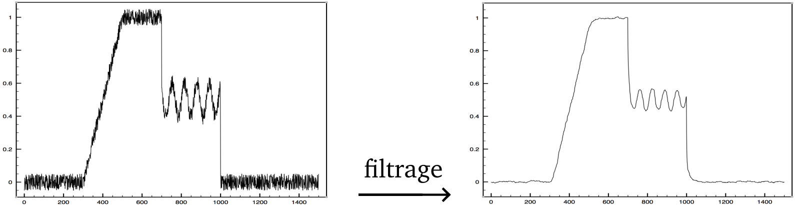
Le signal lissé est la moyenne des N derniers samples du rectifié :

$$l[n] = \frac{1}{N}\sum_{0}^{N} r[n-N]$$

Lissage

Élimine mouvements brusques & oscillations rapides du signal rectifié $r[n]$ pour ne garder qu'un “mouvement général” $l[n]$

$=$ *filtrage passe-bas* (voir plus loin)



Plusieurs implémentations possibles :

Facteur d'oubli

Autrement appelé *filtre passe-bas à un pôle*

$$l[n] = l[n-1] + C(r[n] - l[n-1])$$

Caractérisation d'un filtre dans le temps

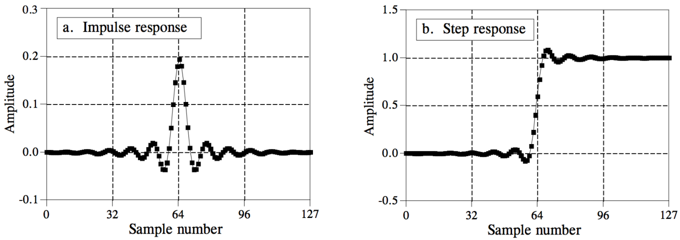
Réponse impulsionnelle réponse à la fonction de dirac :

$$\delta[n] = \begin{cases} 1 \text{ si } n = 0 \\ 0 \text{ sinon} \end{cases}$$

Réponse indicielle réponse à la fonction échelon :

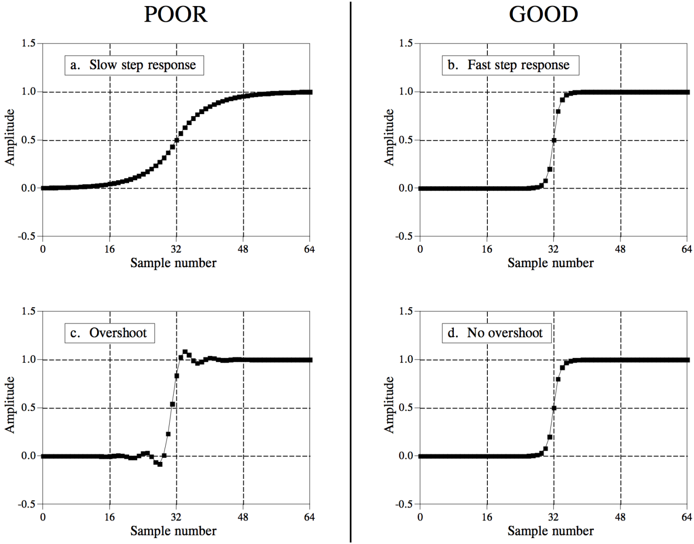
$$H[n] = \begin{cases} 1 \text{ si } n \geq 0 \\ 0 \text{ sinon} \end{cases}$$

Example



Caractérisation d'un filtre dans le temps

La réponse indicielle nous indique comment le filtre réagit à un mouvement brusque

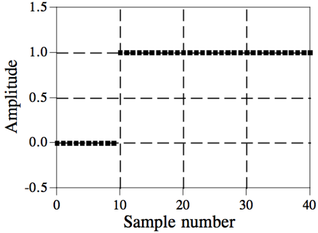


Caractérisation d'un filtre dans le temps

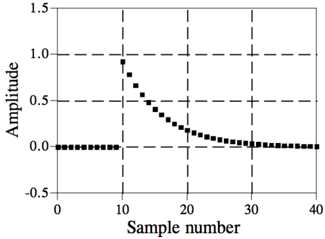
Effets des filtres dans le temps

passe-bas "lisse" le signal entrant / en fait la moyenne (cf. exemples ci-dessus)

passe-haut "élastique" qui rammène le signal vers 0 (plus ou moins vite)



passe-haut $\longrightarrow$



Moyenne glissante

Quel est le graphe de flot qui calcule la moyenne glissante ?

$$l[n] = \frac{1}{N} \sum_{0}^{N} r[n-N]$$

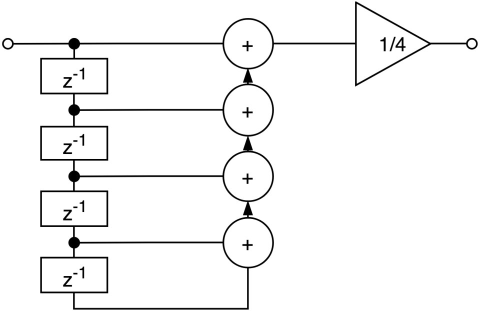
Moyenne glissante

Quel est le graphe de flot qui calcule la moyenne glissante ?

$$l[n] = \frac{1}{N} \sum_{0}^{N} r[n-N]$$

Solution 1

Avec N cases mémoire, N additions et 1 multiplication par sample :



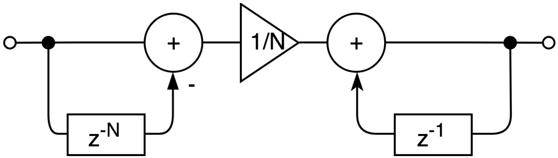
Moyenne glissante

Quel est le graphe de flot qui calcule la moyenne glissante ?

$$l[n] = \frac{1}{N}\sum_{0}^{N} r[n-N]$$

Solution 2

Strictement équivalente, en N cases mémoire, 2 additions et 1 multiplication par sample :



Facteur d'oubli

Un filtre simple et efficace pour lisser un signal dans le temps :

$$\boxed{l[n] = l[n-1] + C\,(r[n] - l[n-1])}$$

- C est le *coefficient* du filtre $(0\ldots1)$
- $r[n] - l[n-1]$ est l'*erreur* (différence entre l'entrée et la sortie)
- le mouvement de $l[n]$ est proportionnel à l'erreur

Facteur d'oubli

Un filtre simple et efficace pour lisser un signal dans le temps :

$$\boxed{l[n] = l[n-1] + C\,(r[n] - l[n-1])}$$

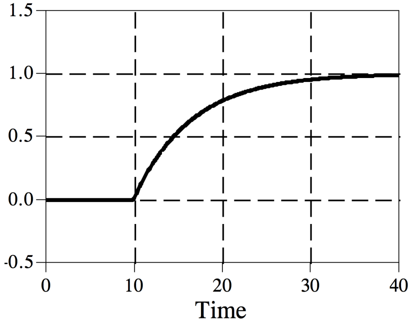
- C est le *coefficient* du filtre $(0\ldots1)$
- $r[n]-l[n-1]$ est l'*erreur* (différence entre l'entrée et la sortie)
- le mouvement de $l[n]$ est proportionnel à l'erreur

Remarques

- si $C=0$, alors $y[n]=y[n-1]$
  (la sortie est immobile)
- si $C=1$, alors $y[n]=x[n]$ (la sortie suit exactement l'entrée)
- si $0<C<1$, $y[n]$ "traîne" derrière $x[n]$

Facteur d'oubli

Réponse indicielle

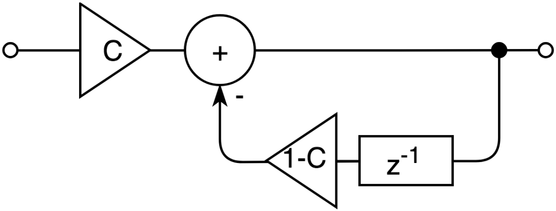


réponse exponentielle, infinie
⇝ *cf.* le paradoxe d'Achille et de la tortue

Facteur d’oubli

Graphe de flot de signal

Une addition, une case mémoire, deux multiplications



Un filtre très peu cher en ressources !

Facteur d'oubli asymmétrique

Pour suivre fidèlement les pics d'un signal, on préfère que le filtre monte rapidement mais redescende plus lentement
$\rightsquigarrow$ deux coefficients :

C_a temps d'attaque

C_r temps de relâchement

Facteur d’oubli asymmétrique

Pour suivre fidèlement les pics d’un signal, on préfère que le filtre monte rapidement mais redescende plus lentement
$\rightsquigarrow$ deux coefficients :

C_a temps d’attaque

C_r temps de relâchement

Solution

$$l[n] = l[n-1] + F(r[n] - l[n-1])$$

$$F(x) = \begin{cases} C_a x & \text{si } x > 0 \\ C_r x & \text{sinon} \end{cases}$$

RMS ou Peak

Root-Mean-Square

Bonne estimation de l'énergie (volume moyen) d'un signal audio ; utilisé en compression pour égaliser les niveaux

rectification fonction carré (*square*)

lissage moyenne glissante (*mean*)
(paramètre : nombre N de samples de la moyenne)

post-traitement on prend la racine carrée (*root*) du signal lissé
(contrebalance l'effet du carré)

RMS ou Peak

Root-Mean-Square

Bonne estimation de l'énergie (volume moyen) d'un signal audio ;
utilisé en compression pour égaliser les niveaux

- **rectification** fonction carré (*square*)
- **lissage** moyenne glissante (*mean*)
  (paramètre : nombre N de samples de la moyenne)
- **post-traitement** on prend la racine carrée (*root*) du signal lissé
  (contrebalance l'effet du carré)

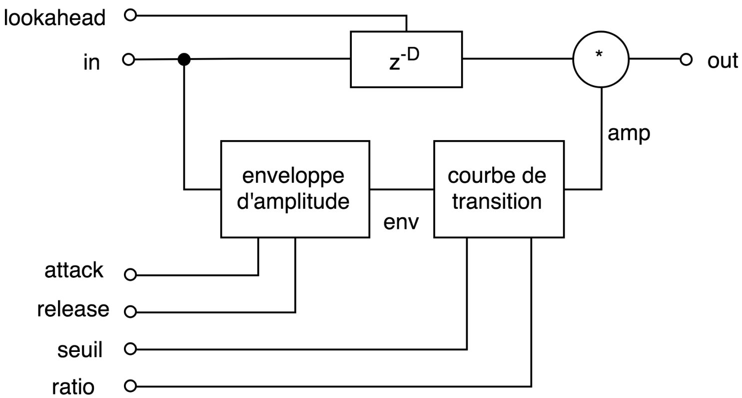
Peak

Suit précisément les contours de la forme d'onde ;
utilisé dans les limiteurs, pour ne jamais dépasser le maximum

- **rectification** totale (valeur absolue)
- **lissage** facteur d'oubli asymétrique avec attaque rapide
  (paramètres : facteurs d'attaque/de relâchement)

Courbe de transition

Rappel



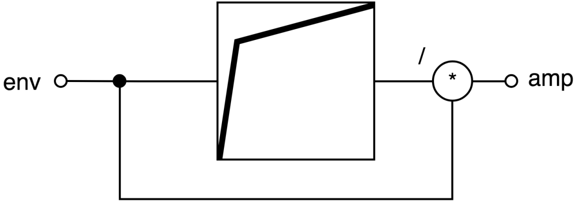
On a un signal env qui représente l'amplitude de in
Comment calculer l'amplification/attenuation amp ?

Courbe de transition

Exemples

- si env=0,5 et la courbe de transition associe 0,7 à cette amplitude, alors il faut amplifier le signal entrant par 1,4 (0,7/0,5=1,4)
- si env=0,8 et la courbe de transition associe 0,6 à cette amplitude, alors il faut atténuer le signal entrant par 0,75 (0,6/0,8=0,75)

Graphe de flot de signal



Audio Numérique

Traitements audio

- Graphe de flot de signal

Traitement de la dynamique

- Principe
- Compresseur, limiteur, gate
- Enveloppe d'amplitude

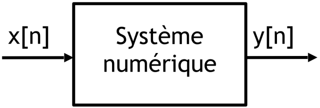
Filtrage

- Une première approximation
- Typologie des filtres
- Caractérisations
- Systèmes linéaires
- Etude de cas

Filtrage

Définition (absurde)

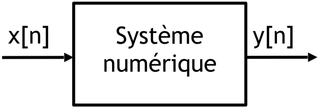
Un filtre est un système numérique



Filtrage

Définition

Un filtre est un système numérique linéaire invariant dans le temps



linéaire

1. changer l'amplitude de l'entrée ne change que l'amplitude de la sortie :

   $A \cdot x[n] \xrightarrow{\text{filtre}} A \cdot y[n]$

2. filtrer la somme de deux signaux revient à sommer chaque signal filtré :

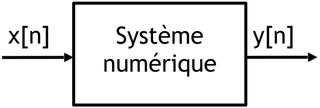
   Si $x_1[n] \xrightarrow{\text{filtre}} y_1[n]$ et $x_2[n] \xrightarrow{\text{filtre}} y_2[n]$,

   alors $x_1[n] + x_2[n] \xrightarrow{\text{filtre}} y_1[n] + y_2[n]$

Filtrage

Définition

Un filtre est un système numérique linéaire invariant dans le temps



invariant dans le temps décaler le signal d'entrée dans le temps ne fait que décaler le signal en sortie :

$$x[n-T] \xrightarrow{\text{filtre}} y[n-T]$$

Filtrage

Mais oublions tout cela un moment…

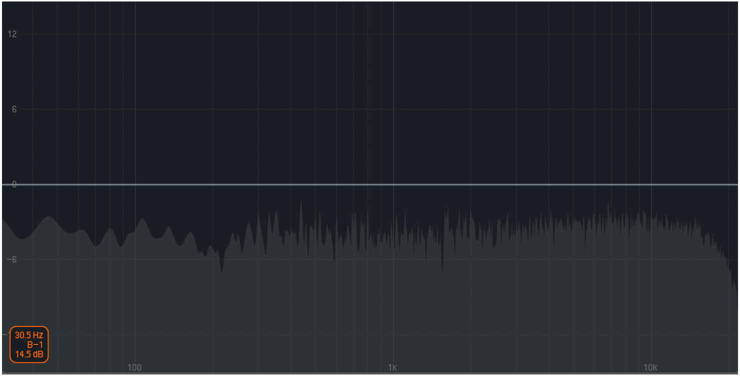
En première approximation

Un filtre modifie le contenu spectral d'un signal en attenuant/amplifiant des fréquences données.

Filtrage

Exemple

Filtrage de bruit blanc

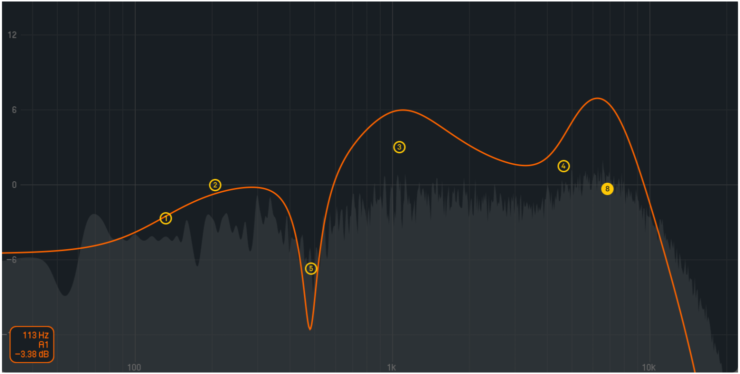


On multiplie son spectrogramme par une *réponse en fréquence*

Filtrage

Exemple

Filtrage de bruit blanc

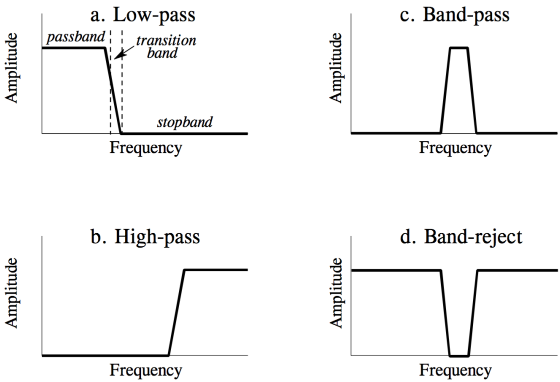


On multiplie son spectrogramme par une *réponse en fréquence*

Un filtre ≈ une réponse en fréquence

(fonction fréquence → amplitude)

Typologie des filtres usuels [09.filters.pd]



- le *passe-bas* coupe les fréquences *au dessus* d'un seuil
- le *passe-haut* coupe les fréquences *en dessous* d'un seuil
- le *coupe-bande* coupe autour d'une fréquence
- le *passe-bande* coupe toutes les fréquences *sauf* autour d'une

Typologie des filtres usuels `[09.filters.pd]`

Applications

band-pass isoler une partie du spectre dans un enregistrement
(ex : un oiseau dans la circulation)

high-pass supprimer les bruits de manipulation
(ex : prise son avec perche ⇝ *rumble* basses fréquences)

band-reject supprimer le *hum* 50Hz parasite

low-pass supprimer les interférences haute fréquence

low-pass couper après la fréquence de Nyquist avant d'échantillonner
(pour éviter l'*aliasing*)

Typologie des filtres usuels [09.filters.pd]

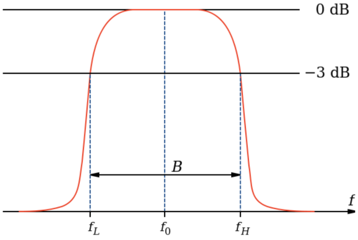
Caractéristiques d'un filtre

fréquence de coupure f_0 (ou *cutoff*) fréquence à partir de laquelle la transition est amorcée
(à -3dB après l'amorce de la descente)

pente "vitesse" de transition (mesurée en dB/Oct)

facteur de qualité pour les filtres passe/coupe-bande :

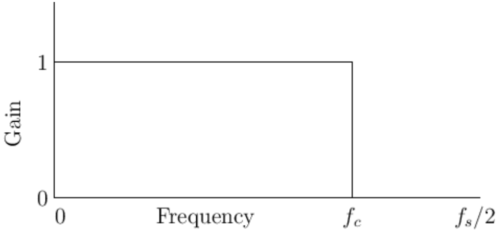
$$Q = \frac{f_0}{f_H - f_L}$$



Typologie des filtres usuels `[09.filters.pd]`

Le filtre idéal

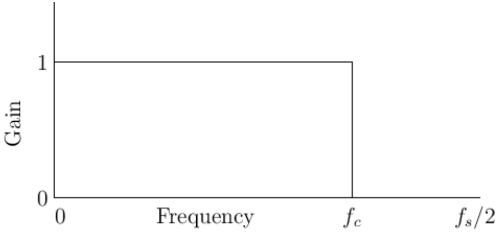
Idéalement on voudrait un filtre de pente infinie,
qui couperait drastiquement à sa fréquence de coupure



Typologie des filtres usuels `[09.filters.pd]`

Le filtre idéal

Idéalement on voudrait un filtre de pente infinie,
qui couperait drastiquement à sa fréquence de coupure

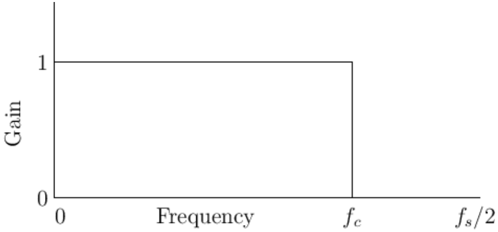


Ce filtre n'existe pas en pratique !
(il demanderait un temps de calcul infini)

Typologie des filtres usuels `[09.filters.pd]`

Le filtre idéal

Idéalement on voudrait un filtre de pente infinie,
qui couperait drastiquement à sa fréquence de coupure



Ce filtre n'existe pas en pratique !
(il demanderait un temps de calcul infini)

⇝ la conception d'un filtre est un ensemble de compromis

Exemple historique : le filtre résonant [10.vcf.pd]

Invention de Bob Moog

- circuit analogique simple et omniprésent
- passe-bas + passe-bande
- deux contrôles :
  - fréquence de coupure f_0
  - facteur de qualite Q (ou *emphasis*)
- la *résonance* est une bosse à f_0 de hauteur proportionnelle à Q

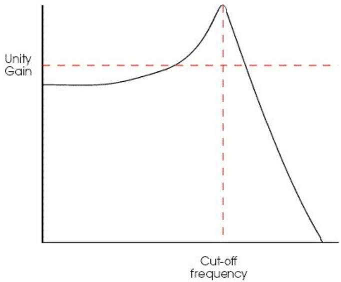


Figure 13: A typical resonant low-pass filter response.

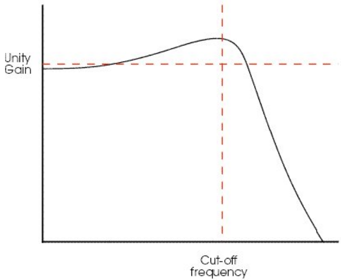
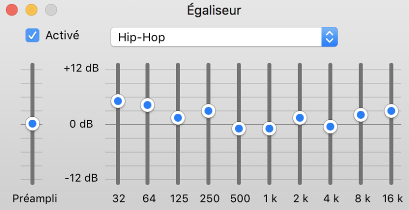


Figure 14: Resonant low-pass filter with low Q.

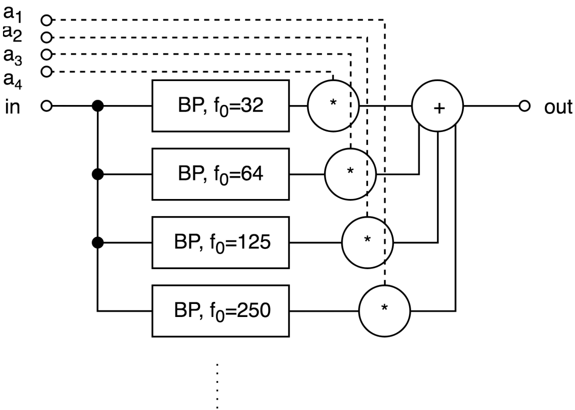
Utilisé dans les lecteurs hi-fi, par goût ou pour corriger les défauts potentiels de l'acoustique d'une pièce



Application Egaliseur

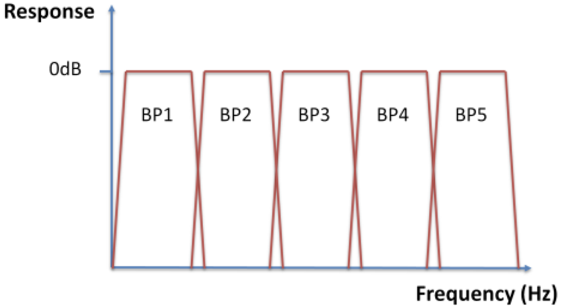
Implémentation

Une banque de filtres passe-bande en parallèle, avec contrôle de leur amplification



Réponse en fréquence

C'est la somme des réponses en fréquence de chaque filtre :



Au delà de cette première approximation

…il y a (beaucoup) plus.

Au delà de cette première approximation

…il y a (beaucoup) plus.

Définition

Un filtre est un système numérique linéaire invariant dans le temps (LTI)

linéaire

1. changer l'amplitude de l'entrée ne change que l'amplitude de la sortie :
   $A \cdot x[n] \xrightarrow{\text{filtre}} A \cdot y[n]$
2. filtrer la somme de deux signaux revient à sommer chaque signal filtré :
   Si $x_1[n] \xrightarrow{\text{filtre}} y_1[n]$ et $x_2[n] \xrightarrow{\text{filtre}} y_2[n]$,
   alors $x_1[n] + x_2[n] \xrightarrow{\text{filtre}} y_1[n] + y_2[n]$

invariant dans le temps décaler le signal d'entrée dans le temps ne fait que décaler le signal en sortie :
$x[n-T] \xrightarrow{\text{filtre}} y[n-T]$

Caractérisations d’un filtre

Il existe plusieurs représentations d’un même filtre.
Toutes contiennent les mêmes informations
et sont interchangeables :

- réponse en fréquence et en phase
  (comment le filtre réagit à des sinusoïdes)
- équation de différence / graphe de flot de signal (comment le filtre est implémenté)
- réponse impulsionnelle (comment le filtre réagit à une impulsion brève)

Réponse en fréquence et en phase

Que se passe-t-il si on excite un filtre avec une sinusoïde ?

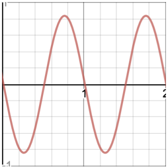
Réponse en fréquence et en phase

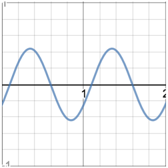
Que se passe-t-il si on excite un filtre avec une sinusoïde ?

Théorème

Un filtre soumis à sinusoïde produit sinusoïde de même fréquence (avec une amplitude et une phase potentiellement différente)

$$A\cos(2\pi fn + \phi) \xrightarrow{\text{filtre}} B\cos(2\pi fn + \theta)$$





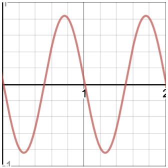
Réponse en fréquence et en phase

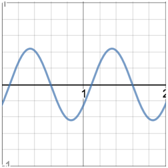
Que se passe-t-il si on excite un filtre avec une sinusoïde ?

Théorème

Un filtre soumis à sinusoïde produit sinusoïde de même fréquence (avec une amplitude et une phase potentiellement différente)

$$A\cos(2\pi fn+\phi)\xrightarrow{\text{filtre}}B\cos(2\pi fn+\theta)$$



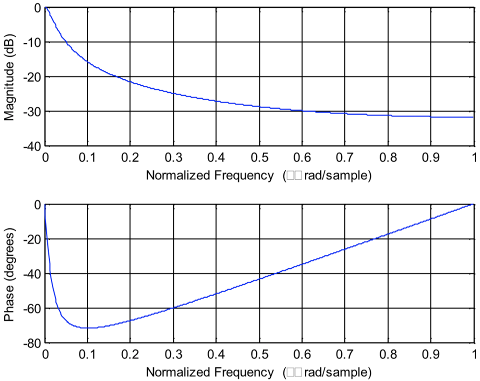


- le rapport des amplitudes B/A en fonction de la fréquence définit la *réponse en fréquence* $X(f)$
- la différence des phases $\theta-\phi$ en fonction de la fréquence est la *réponse en phase* $\Theta(f)$

Réponse en fréquence et en phase

Exemple : le filtre “facteur d’oubli”

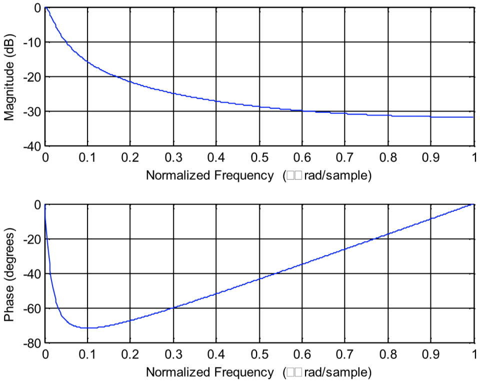
avec $C = 0.05$: réponse en fréquence, puis réponse en phase



Réponse en fréquence et en phase

Exemple : le filtre "facteur d'oubli"

avec $C = 0.05$: réponse en fréquence, puis réponse en phase



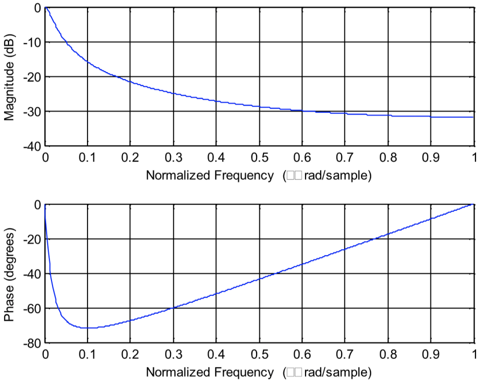
Remarque

Par la propriété de superposition et la théorie de Fourier, on peut déduire la sortie du filtre sur toute entrée

Réponse en fréquence et en phase

Exemple : le filtre “facteur d’oubli”

avec $C = 0.05$: réponse en fréquence, puis réponse en phase



Remarque

Par la propriété de superposition et la théorie de Fourier,
on peut déduire la sortie du filtre sur toute entrée ⇝ *caractérisation*

Equation de différence

Exprime un sample de sortie en fonction des samples précédents :

Exemple (facteur d'oubli : $x[n] \rightarrow y[n]$)

$y[n] = x[n-1] + C(x[n] - y[n-1])$

Equation de différence

Exprime un sample de sortie en fonction des samples précédents :

Exemple (facteur d'oubli : $x[n] \rightarrow y[n]$)

$y[n] = x[n-1] + C(x[n] - y[n-1])$

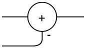
Théorème

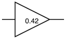
Tout filtre est la somme pondérée de son historique d'entrées/sorties :

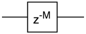
$$\begin{aligned} y[n] = a_0 x[n] + a_1 x[n-1] + \cdots + a_N x[n-N] \\ + b_1 y[n-1] + \cdots + b_N y[n-N] \end{aligned}$$

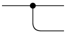
Equation de différence

Remarques

- le graphe d'un filtre est causal et ne s'écrit qu'avec :
  - l'addition 

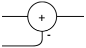
  - la multiplication constante 

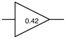
  - le retard 

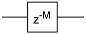
  - la copie 

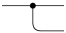
Equation de différence

Remarques

- le graphe d'un filtre est causal et ne s'écrit qu'avec :
  - l'addition 

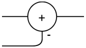
  - la multiplication constante 

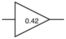
  - le retard 

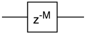
  - la copie 
- un filtre qui ne dépend que de ses valeurs d'entrée ($b_i = 0$) est dit à *réponse impulsionnelle finie* (*FIR*)

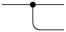
Equation de différence

Remarques

- le graphe d'un filtre est causal et ne s'écrit qu'avec :
  - l'addition 

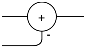
  - la multiplication constante 

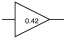
  - le retard 

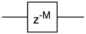
  - la copie 
- un filtre qui ne dépend que de ses valeurs d'entrée ($b_i = 0$) est dit à *réponse impulsionnelle finie* (*FIR*)
- un filtre qui dépend de sa propre sortie ($b_i \neq 0$) est dit *récursif*, ou à *réponse impulsionnelle infinie* (*IIR*)

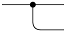
Equation de différence

Remarques

- le graphe d'un filtre est causal et ne s'écrit qu'avec :
  - l'addition 

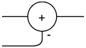
  - la multiplication constante 

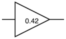
  - le retard 

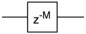
  - la copie 
- un filtre qui ne dépend que de ses valeurs d'entrée ($b_i = 0$) est dit à *réponse impulsionnelle finie* (*FIR*)
- un filtre qui dépend de sa propre sortie ($b_i \neq 0$) est dit *récursif*, ou à *réponse impulsionnelle infinie* (*IIR*)
- la taille N de l'historique est l'*ordre* du filtre
  (facteur d'oubli = filtre du premier ordre)

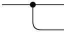
Equation de différence

Remarques

- le graphe d'un filtre est causal et ne s'écrit qu'avec :
  - l'addition 

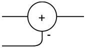
  - la multiplication constante 

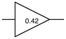
  - le retard 

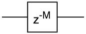
  - la copie 
- un filtre qui ne dépend que de ses valeurs d'entrée ($b_i = 0$) est dit à *réponse impulsionnelle finie* (*FIR*)
- un filtre qui dépend de sa propre sortie ($b_i \neq 0$) est dit *récursif*, ou à *réponse impulsionnelle infinie* (*IIR*)
- la taille N de l'historique est l'*ordre* du filtre
  (facteur d'oubli = filtre du premier ordre)
- les coefficients a_i et b_i déterminent complètement sa réponse

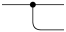
Equation de différence

Remarques

- le graphe d'un filtre est causal et ne s'écrit qu'avec :
  - l'addition 

  - la multiplication constante 

  - le retard 

  - la copie 
- un filtre qui ne dépend que de ses valeurs d'entrée ($b_i = 0$) est dit à *réponse impulsionnelle finie* (*FIR*)
- un filtre qui dépend de sa propre sortie ($b_i \neq 0$) est dit *récursif*, ou à *réponse impulsionnelle infinie* (*IIR*)
- la taille N de l'historique est l'*ordre* du filtre
  (facteur d'oubli = filtre du premier ordre)
- les coefficients a_i et b_i déterminent complètement sa réponse
  ⇝ *caractérisation*

Réponse Impulsionnelle

Que se passe-t-il si on excite un filtre avec un signal d'un sample ?

Réponse Impulsionnelle

Que se passe-t-il si on excite un filtre avec un signal d'un sample ?

Définition (RI)

signal de sortie $h[n]$ d'un filtre soumis à une *impulsion unitaire* $\delta[n]$:

$$\delta[n] = \begin{cases} 1 \text{ si } n = 0 \\ 0 \text{ sinon} \end{cases} \xrightarrow{\textit{filtre}} h[n]$$

Réponse Impulsionnelle

Que se passe-t-il si on excite un filtre avec un signal d'un sample ?

Définition (RI)

signal de sortie $h[n]$ d'un filtre soumis à une *impulsion unitaire* $\delta[n]$:

$$\delta[n] = \begin{cases} 1 \text{ si } n = 0 \\ 0 \text{ sinon} \end{cases} \xrightarrow{\textit{filtre}} h[n]$$

Remarque

Par la propriété de superposition, la RI permet de déduire la sortie du filtre sur toute entrée

Réponse Impulsionnelle

Que se passe-t-il si on excite un filtre avec un signal d'un sample ?

Définition (RI)

signal de sortie $h[n]$ d'un filtre soumis à une *impulsion unitaire* $\delta[n]$:

$$\delta[n] = \begin{cases} 1 \text{ si } n = 0 \\ 0 \text{ sinon} \end{cases} \xrightarrow{\textit{filtre}} h[n]$$

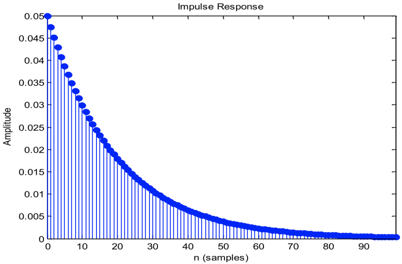
Remarque

Par la propriété de superposition, la RI permet de déduire la sortie du filtre sur toute entrée $\rightsquigarrow$ *caractérisation*

Réponse Impulsionnelle

Exemple : le filtre “facteur d’oubli”

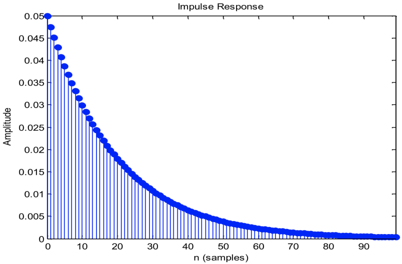
Sa réponse impulsionnelle est infinie (IIR) :



Réponse Impulsionnelle

Exemple : le filtre "facteur d'oubli"

Sa réponse impulsionnelle est infinie (IIR) :



Produit de convolution

C'est l'opération $x * h$ permettant de construire la réponse $y[n]$ du filtre à un signal quelconque $x[n]$ à partir de sa RI $h[n]$

$$\text{Si } x[n] \xrightarrow{\text{filtre de RI } h[n]} y[n], \text{ alors } x * h = y$$

Parenthèse Le produit de convolution

Une opération extrêmement générale, que l'on retrouve dans divers branches des mathématiques
(traitement du signal, probabilité, algèbre, *deep learning*...)

$$(f * g)[n] = \sum_{i=0}^{\infty} f[n-i]g[i]$$

Parenthèse Le produit de convolution

Une opération extrêmement générale, que l'on retrouve dans divers branches des mathématiques
(traitement du signal, probabilité, algèbre, *deep learning*...)

$$(f * g)[n] = \sum_{i=0}^{\infty} f[n-i]g[i]$$

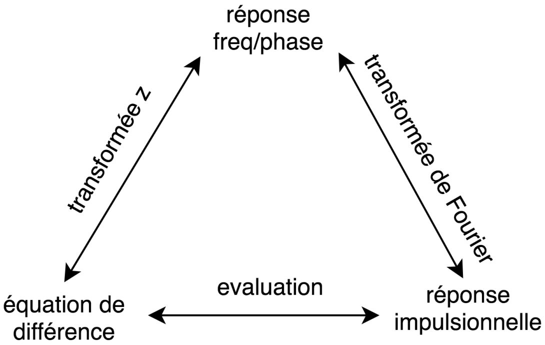
Dualité temps/fréquence

- calculer la convolution de deux signaux revient à multiplier leur réponse en fréquence, point-à-point
- calculer la convolution de deux réponses en fréq. revient à multiplier les deux signaux point-à-point

⇝ correspondance très profonde, et très utile
(ex : pour comprendre et anticiper l'effet d'un traitement)

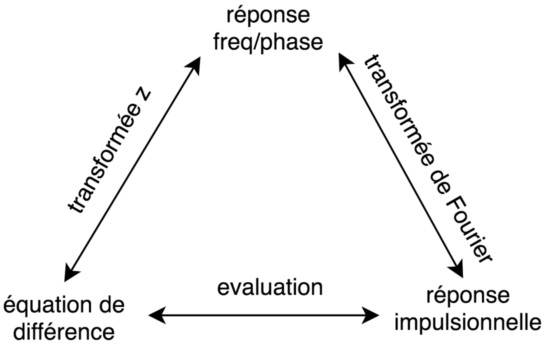
Caractérisation d'un filtre

Trois caractérisations équivalentes,
trois angles de vision sur le même objet



Caractérisation d'un filtre

Trois caractérisations équivalentes,
trois angles de vision sur le même objet



- Les "traductions" se font par des outils mathématiques (transformée z, de Fourier...)
- permettent la conception de filtre aux propriétés voulues

Linéarité et traitements usuels [11.linear]

Un filtre fait la somme d'un signal et de versions retardées de lui
⇝ un grand nombre d'effets audio sont des filtres !
(au delà du simple filtre LP/HP/BP)

Linéarité et traitements usuels `[11.linear]`

Un filtre fait la somme d'un signal et de versions retardées de lui
$\rightsquigarrow$ un grand nombre d'effets audio sont des filtres !
(au delà du simple filtre LP/HP/BP)

- gain

Linéarité et traitements usuels [11.linear]

Un filtre fait la somme d'un signal et de versions retardées de lui
⇝ un grand nombre d'effets audio sont des filtres !
(au delà du simple filtre LP/HP/BP)

- gain ⇝ linéaire

Linéarité et traitements usuels `[11.linear]`

Un filtre fait la somme d'un signal et de versions retardées de lui
$\rightsquigarrow$ un grand nombre d'effets audio sont des filtres !
(au delà du simple filtre LP/HP/BP)

- gain $\rightsquigarrow$ linéaire
- égaliseur

Linéarité et traitements usuels [11.linear]

Un filtre fait la somme d'un signal et de versions retardées de lui
⇝ un grand nombre d'effets audio sont des filtres !
(au delà du simple filtre LP/HP/BP)

- gain ⇝ linéaire
- égaliseur ⇝ linéaire

Linéarité et traitements usuels [11.linear]

Un filtre fait la somme d'un signal et de versions retardées de lui
$\rightsquigarrow$ un grand nombre d'effets audio sont des filtres !
(au delà du simple filtre LP/HP/BP)

- gain $\rightsquigarrow$ linéaire
- égaliseur $\rightsquigarrow$ linéaire
- résonateur

Linéarité et traitements usuels [11.linear]

Un filtre fait la somme d'un signal et de versions retardées de lui
$\rightsquigarrow$ un grand nombre d'effets audio sont des filtres !
(au delà du simple filtre LP/HP/BP)

- gain $\rightsquigarrow$ linéaire
- égaliseur $\rightsquigarrow$ linéaire
- résonateur $\rightsquigarrow$ linéaire

Linéarité et traitements usuels `[11.linear]`

Un filtre fait la somme d'un signal et de versions retardées de lui
$\rightsquigarrow$ un grand nombre d'effets audio sont des filtres !
(au delà du simple filtre LP/HP/BP)

- gain $\rightsquigarrow$ linéaire
- égaliseur $\rightsquigarrow$ linéaire
- résonateur $\rightsquigarrow$ linéaire
- compresseur

Linéarité et traitements usuels [11.linear]

Un filtre fait la somme d'un signal et de versions retardées de lui
⇝ un grand nombre d'effets audio sont des filtres !
(au delà du simple filtre LP/HP/BP)

- gain ⇝ linéaire
- égaliseur ⇝ linéaire
- résonateur ⇝ linéaire
- compresseur ⇝ non-linéaire

Linéarité et traitements usuels [11.linear]

Un filtre fait la somme d'un signal et de versions retardées de lui
⇝ un grand nombre d'effets audio sont des filtres !
(au delà du simple filtre LP/HP/BP)

- gain ⇝ linéaire
- égaliseur ⇝ linéaire
- résonateur ⇝ linéaire
- compresseur ⇝ non-linéaire
- distortion

Linéarité et traitements usuels [11.linear]

Un filtre fait la somme d'un signal et de versions retardées de lui
⇝ un grand nombre d'effets audio sont des filtres !
(au delà du simple filtre LP/HP/BP)

- gain ⇝ linéaire
- égaliseur ⇝ linéaire
- résonateur ⇝ linéaire
- compresseur ⇝ non-linéaire
- distortion ⇝ non-linéaire

Linéarité et traitements usuels [11.linear]

Un filtre fait la somme d'un signal et de versions retardées de lui
$\rightsquigarrow$ un grand nombre d'effets audio sont des filtres !
(au delà du simple filtre LP/HP/BP)

- gain $\rightsquigarrow$ linéaire
- égaliseur $\rightsquigarrow$ linéaire
- résonateur $\rightsquigarrow$ linéaire
- compresseur $\rightsquigarrow$ non-linéaire
- distortion $\rightsquigarrow$ non-linéaire
- retard

Linéarité et traitements usuels [11.linear]

Un filtre fait la somme d'un signal et de versions retardées de lui
$\rightsquigarrow$ un grand nombre d'effets audio sont des filtres !
(au delà du simple filtre LP/HP/BP)

- gain $\rightsquigarrow$ linéaire
- égaliseur $\rightsquigarrow$ linéaire
- résonateur $\rightsquigarrow$ linéaire
- compresseur $\rightsquigarrow$ non-linéaire
- distortion $\rightsquigarrow$ non-linéaire
- retard $\rightsquigarrow$ linéaire

Linéarité et traitements usuels [11.linear]

Un filtre fait la somme d'un signal et de versions retardées de lui
$\rightsquigarrow$ un grand nombre d'effets audio sont des filtres !
(au delà du simple filtre LP/HP/BP)

- gain $\rightsquigarrow$ linéaire
- égaliseur $\rightsquigarrow$ linéaire
- résonateur $\rightsquigarrow$ linéaire
- compresseur $\rightsquigarrow$ non-linéaire
- distortion $\rightsquigarrow$ non-linéaire
- retard $\rightsquigarrow$ linéaire
- réverbération

Linéarité et traitements usuels `[11.linear]`

Un filtre fait la somme d'un signal et de versions retardées de lui
⇝ un grand nombre d'effets audio sont des filtres !
(au delà du simple filtre LP/HP/BP)

- gain ⇝ linéaire
- égaliseur ⇝ linéaire
- résonateur ⇝ linéaire
- compresseur ⇝ non-linéaire
- distortion ⇝ non-linéaire
- retard ⇝ linéaire
- réverbération ⇝ linéaire

Linéarité et traitements usuels [11.linear]

Un filtre fait la somme d'un signal et de versions retardées de lui
⇝ un grand nombre d'effets audio sont des filtres !
(au delà du simple filtre LP/HP/BP)

- gain ⇝ linéaire
- égaliseur ⇝ linéaire
- résonateur ⇝ linéaire
- compresseur ⇝ non-linéaire
- distortion ⇝ non-linéaire
- retard ⇝ linéaire
- réverbération ⇝ linéaire
- chorus, flanger

Linéarité et traitements usuels `[11.linear]`

Un filtre fait la somme d'un signal et de versions retardées de lui
$\rightsquigarrow$ un grand nombre d'effets audio sont des filtres !
(au delà du simple filtre LP/HP/BP)

- gain $\rightsquigarrow$ linéaire
- égaliseur $\rightsquigarrow$ linéaire
- résonateur $\rightsquigarrow$ linéaire
- compresseur $\rightsquigarrow$ non-linéaire
- distortion $\rightsquigarrow$ non-linéaire
- retard $\rightsquigarrow$ linéaire
- réverbération $\rightsquigarrow$ linéaire
- chorus, flanger $\rightsquigarrow$ ... semi-linéaire

Propriétés des traitements linéaires

Les systèmes formés uniquement de filtres
ont des propriétés intéressantes :

Propriétés des traitements linéaires

Les systèmes formés uniquement de filtres
ont des propriétés intéressantes :

distributivité deux filtres f_1 et f_2 en parallèle forment un filtre, dont la réponse en fréquence est l'addition point-à-point des réponses

Propriétés des traitements linéaires

Les systèmes formés uniquement de filtres
ont des propriétés intéressantes :

distributivité
: deux filtres f_1 et f_2 en parallèle forment un filtre, dont la réponse en fréquence est l'addition point-à-point des réponses
(ex : l'egaliseur vu précédemment)

associativité
: deux filtres f_1 et f_2 en série forment un filtre, dont la réponse en fréquence est la multiplication point-à-point des réponses
(ex : un LP suivi d'un HP = un BP)

Propriétés des traitements linéaires

Les systèmes formés uniquement de filtres
ont des propriétés intéressantes :

distributivité
: deux filtres f_1 et f_2 en parallèle forment un filtre, dont la réponse en fréquence est l'addition point-à-point des réponses
(ex : l'egaliseur vu précédemment)

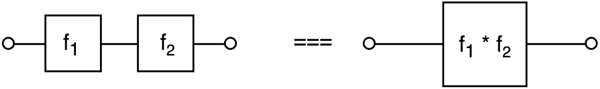
associativité
: deux filtres f_1 et f_2 en série forment un filtre, dont la réponse en fréquence est la multiplication point-à-point des réponses
(ex : un LP suivi d'un HP = un BP)

commutativité
: l'ordre d'application de filtres en série n'a pas d'influence sur la sortie
(ex : $f_1 \to f_2 \to f_3 \equiv f_2 \to f_3 \to f_1$)

Propriétés des traitements linéaires

Applications

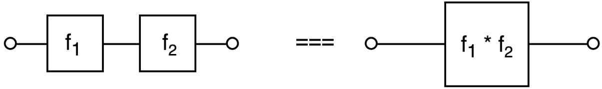
- l'application de deux égaliseurs successifs fait la même chose que l'appication d'un égaliseur combinant les deux traitements



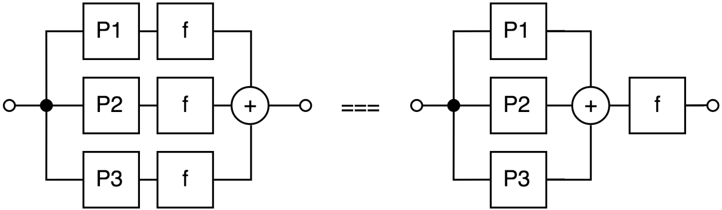
Propriétés des traitements linéaires

Applications

- l'application de deux égaliseurs successifs fait la même chose que l'appication d'un égaliseur combinant les deux traitements



- pas nécessaire d'appliquer le même filtre sur chaque voix d'un mixage, on peut l'appliquer directement sur le *master* (moins de processeur utilisé, plus flexible)



Etude de cas Filtre “première différence”

Calcule la différence des deux dernières valeurs d’entrée :

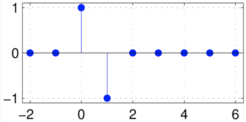
$$y[n] = x[n] - x[n-1]$$

Etude de cas Filtre “première différence”

Calcule la différence des deux dernières valeurs d’entrée :

$$y[n] = x[n] - x[n-1]$$

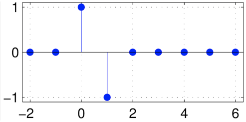
- réponse impulsionnelle



Etude de cas Filtre “première différence”

Calcule la différence des deux dernières valeurs d'entrée :

$$y[n] = x[n] - x[n-1]$$

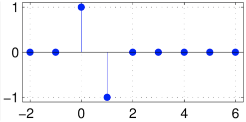
- réponse impulsionnelle 

$\rightsquigarrow$ FIR

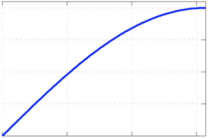
Etude de cas Filtre "première différence"

Calcule la différence des deux dernières valeurs d'entrée :

$$y[n] = x[n] - x[n-1]$$

- réponse impulsionnelle 

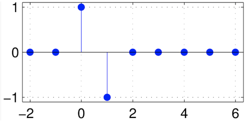
$\rightsquigarrow$ FIR

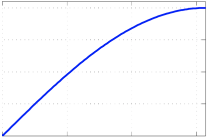
- réponse en fréquence 

Etude de cas   Filtre “première différence”

Calcule la différence des deux dernières valeurs d’entrée :

$$y[n] = x[n] - x[n-1]$$

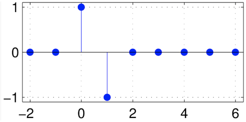
- réponse impulsionnelle 

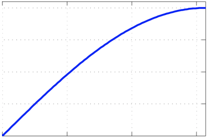
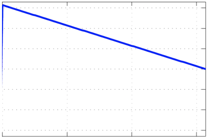
$\rightsquigarrow$ FIR
- réponse en fréquence  $\rightsquigarrow$ *passe-haut*

Etude de cas Filtre "première différence"

Calcule la différence des deux dernières valeurs d'entrée :

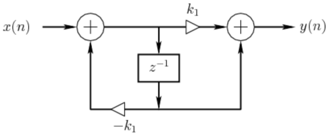
$$y[n] = x[n] - x[n-1]$$

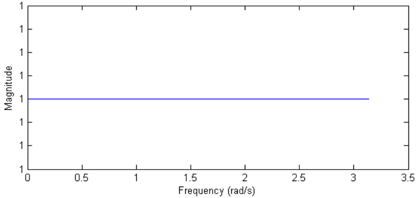
- réponse impulsionnelle 

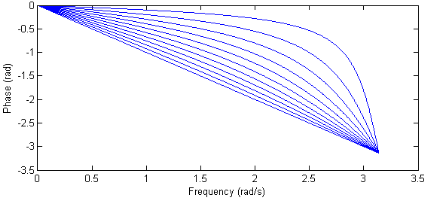
$\rightsquigarrow$ FIR
- réponse en fréquence  $\rightsquigarrow$ *passe-haut*
- réponse en phase 

Etude de cas Filtre passe-tout

Laisse passer toutes les fréquences mais modifie la phase





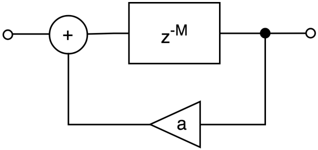


Applications

- compenser le déphasage d'un autre filtre
  (implémentation des filtres à phase linéaire)
- mixer $y[n]$ avec $x[n]$ pour créer des trous dans le spectre
  (principe du *phaser*)
- effet de *dispersion*/d'étalement dans le temps du signal
  (base des réverbérations algorithmiques)

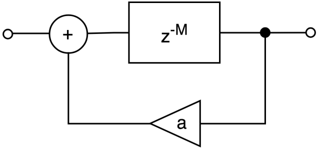
Etude de cas Filtre en peigne

Un retard dans une boucle de rétroaction :

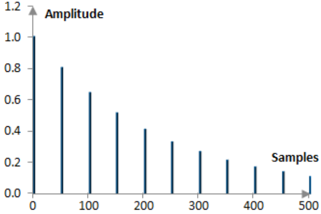


Etude de cas Filtre en peigne

Un retard dans une boucle de rétroaction :

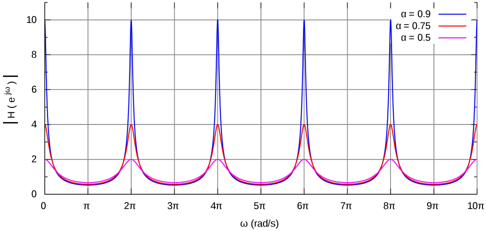


- réponse impulsionnelle :



Etude de cas   Filtre en peigne

- réponse en fréquence :

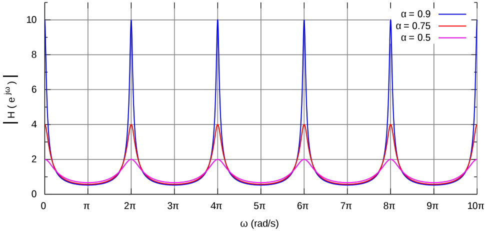


⇝ ressemble à un peigne

- la hauteur des pics dépend de a (coefficient de rétroaction)
- leur écartement dépend de M (temps de retard)

Etude de cas Filtre en peigne

- réponse en fréquence :



⇝ ressemble à un peigne

- la hauteur des pics dépend de a (coefficient de rétroaction)
- leur écartement dépend de M (temps de retard)

Quiz

Qu'entend-t-on quand le temps de retard M est très long ?