Eléments de Traitement du Signal

Lionel Lacassagne

LIP6
Université Pierre et Marie Curie (Paris 6)

UPMC
SORBONNE UNIVERSITÉS

Pourquoi ?

- Ce n'est pas un cours de TS !
- Vous en faites déjà mais on ne vous l'a pas dit (pour ne pas vous faire peur ;-)
- Parce que c'est nécessaire pour manipuler des grandeurs physiques à travers des capteurs
- Eléments de TS = les *bouts* de TS dont vous avez besoin pour faire vous même votre système (DIY)
  - comprendre ce qu'on appelle le bruit
  - les différents types de filtres pour filtrer le bruit (*débruiter*)
  - les différents types de filtres pour détecter des informations particulières dans des données (contours, transition)
  - être capable d'implémenter un opérateur de TS à partir d'une équation (très simple quand expliqué dans le bon sens)
- ⇒ vous rendre autonome pour vos "manip / bidouilles"
- parce que la programmation d'un système embarqué consiste à implémenter / parallélisme / optimiser des opérateurs de TS
- ⇒ améliorer votre CV d'électronicien/informaticien pour vos candidatures (stages, emplois, ...)

Exemples

- audio : codec + amélioration, suppression du bruit de fond (vent sur micro)
- image : stabilisation pour prendre des photos, gestion de l'accéléromètre pour faire des images panoramiques
- vidéo : stabilisation d'une caméra embarquée pour avoir un film *regardable* ex : descente en VTT
- télé +100 Hz : création d'images intermédiaires pour avoir un rendu plus fluide

Origine et utilisation des filtres #1

- Les grandeurs physiques sont continues
  - sens mathématiques : dérivables (de dérivés continues)
  - sens physique : varient *lentement* par rapport à une perturbation
  - $\Rightarrow$ objectif : filtrer pour débruiter = diminuer l'impact de la perturbation, la supprimer.
  - filtrage passe-bas : on cherche la partie continue du signal (et non ses variations)

- Les grandeurs physiques varient au cours du temps
  - $\Rightarrow$ objectif : filtrer pour détecter et mesurer les variations = faire la part entre les variations dues au bruit et les variations de la grandeur physique.
  - filtrage passe-haut : on cherche la partie haute-fréquence (= qui varie) du signal

Monde réel continu de la Physique et monde discret numérique du calcul

- monde continu vs monde discret
  - Dans le monde réel les signaux (associés aux grandeurs physiques) sont continus
  - Dans le monde numérique, les signaux sont échantillonnés et discrétisés.

- Mathématiques vs Physique
  - En mathématiques, les fonctions sont connues, on peut les intégrer, calculer leur primitive ou les dériver : $\int_a^b f(x)dx = [F(x)]_a^b$, $\frac{\partial f(x)}{dx}$
  - En physique, on a des valeurs mesurées, pas de fonctions, et on construit un modèle de comportement, et un modèle de bruit.

- Et on souhaite les traiter pour :
  - retrouver la fonction qui a produit ces valeurs $\rightarrow$ *problèmes inverses*
  - les analyser : l'intégration devient numérique c'est une somme (des additions), la dérivation et la différenciation deviennent aussi numérique, c'est un ensemble d'additions et de soustractions : $\int \rightarrow ++$ et $\partial \rightarrow +-$

Origine et utilisation des filtres #2

- Intégration numérique
  - $\int_a^b f(x)dx = \sum_{k=0}^{n-1} f(a+k\times h)\times h$ avec $h=(b-a)/n$
  - intégration rectangles : somme des produits hauteur $\times$ largeur
- Dérivation numérique et taux de variation
  - dérivée à droite : $f'_d(x) = \lim\limits_{\substack{h\to 0\\ h>0}} \frac{f(x+h)-f(x)}{h}$
  - dérivée centrale : $f'_c(x) = \lim\limits_{\substack{h\to 0\\ h>0}} \frac{f(x+h)-f(x-h)}{2h}$
- Changement de notations
  - l'inconnue $x \to n$ le numéro de l'échantillon
  - grandeur $f \to x$ la variable
  - souvent, $h=1$
- Appliquées à l'intégrale et à la dérivée
  - la somme devient $\sum_{k=0}^{n-1} f(x_k) = f(x_0)+f(x_1)+\ldots+f(x_n-1)$,
  - les dérivées deviennent $f'_d(x) \simeq (x_{n+1}-x_n)$ et $f'_c(x) \simeq (x_{n+1}-x_{n-1})/2$
  - additions = intégration et soustractions = dérivation

La convolution

- C'est le fait d'appliquer un filtre à un signal.
- Soit x un signal en entrée, h un filtre de m coefficients et y le signal de sortie, alors $\boxed{\text{y = x * h}}$ (le symbole de la convolution est l'étoile).
- Pour tout point n du signal

$$\boxed{y(n) = \sum_{k=0}^{k=m-1} h(h)x(n-k)}$$

- Ce calcul est répété pour les n points de la sortie y.

- Filtrer un signal, convoluer un signal par un filtre, ce n'est que réaliser une somme pondérée.
- c'est une opération bien plus simple et bien plus courante qu'on ne le croit ...

Convolution et gestion des bords

- Soit le signal x et le filtre $h = [3, 2, 1]/6$. En convoluant x et h on obtient :
  - $(3 \times 6 + 2 \times 5 + 1 \times 4)/6 = 32/6 \simeq 5.3$
  - $(3 \times 7 + 2 \times 6 + 1 \times 5)/6 = 38/6 \simeq 6.3$

n	0	1	2	3
$x(n)$	4	5	6	7
$y(n)$	?	?	5.3	6.3

  - il manque $y(0)$ et $y(1)$

- Plusieurs stratégies possibles :
  - recopie des premières valeurs de l'entrée

n	0	1	2	3
$x(n)$	4	5	6	7
$y(n)$	4	5	5.3	6.3

  - duplication de **la** première valeur de sortie

n	0	1	2	3
$x(n)$	4	5	6	7
$y(n)$	5.3	5.3	5.3	6.3

Convolution et gestion des bords

Une autre stratégie :

- Soit le signal x et le filtre $h = [3, 2, 1]/6$. En convoluant x et h on obtient :
  - extension du signal x en duplicant la première entrée :
  - $(3 \times 5 + 2 \times 4 + 1 \times 4)/6 = 27/6 \simeq 4.7$
  - $(3 \times 4 + 2 \times 4 + 1 \times 4)/6 = 24/6 = 4.0$

n	-2	-1	0	1	2	3
$x(n)$	4	4	4	5	6	7
$y(n)$			4	4.7	5.3	6.3

  - comme le filtre était normalisé (somme des coefficients égale à 1) il était inutile de calculer $y(0)$, car moyenner un signal constant ne change pas la valeur : c'est $x(0)$.
- avantage : les valeurs de sortie sont progressives et non constantes, contrairement aux cas précédents.

Construction de quelques filtres

- filtres moyenneurs M_k (k la taille du filtre) : $M_k = [1, 1, \ldots, 1]/k$
  - $M_2 = [1, 1]/2$, $M_2 = [1, 1, 1]/3$, $M_5 = [1, 1, 1, 1, 1]/5$
- Dans un certains nombres de cas, le filtre causal de taille k (filtrage de la valeur courant et des valeurs dans l'intervalle $[n : n - k + 1]$ est remplacé par un filtre à "cheval" sur le passé et le futur (centré en n et filtrant les valeurs dans l'intervalle $[n - k/2 : n + k/2]$).
- Dans ce cas, les filtres de taille impaire sont préférés aux filtres de taille paire : il faut donc être capable de calculer $1/k$.

- filtres binomiaux B_k (k la taille du filtre) : coefficients du triangle de Pascal.
  - autre manière de retrouver les coefficients : en convoluant plusieurs fois M_1 avec lui-même
  - $B_2 = \frac{1}{2}[1, 1] * \frac{1}{2}[1, 1] = \frac{1}{4}[1, 2, 1]$
  - $B_3 = \frac{1}{2}[1, 1] * \frac{1}{4}[1, 2, 1] = \frac{1}{8}[1, 3, 3, 1]$
  - $B_4 = \frac{1}{2}[1, 1] * \frac{1}{8}[1, 3, 3, 1] = \frac{1}{16}[1, 4, 6, 4, 1]$

Construction du filtre gaussien

- filtre gaussien de paramètre σ

$$G_\sigma(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}}$$

- taille de la fenêtre du filtre ? de rayon 2σ ou 3σ.
- table de valeurs (calcul avec arrondi) pour $\sigma = \sqrt{2}/2 = 0.7$ (normalement somme $= 2^p$ et $\delta = 2^p -$ somme)

x	0	1	2	3	somme	δ
$\mathbb{F}$	0.564	0.208	0.010	0.000		
Q_6	36	13	1	0	64	0
Q_8	144	53	3	0	256	0
Q_9	289	106	5	0	511	-1
Q_{10}	578	213	11	0	1024	+2
Q_{13}	4622	1700	85	1	16386	+2

Construction du filtre gaussien

- table de valeurs (calcul avec arrondi) pour $\sigma = 1$

x	0	1	2	3	4	somme	δ
$\mathbb{F}$	0.399	0.242	0.054	0.004	0.000		
Q_6	26	15	3	0	0	62	-2
Q_8	102	62	14	1	0	256	0
Q_{10}	409	248	55	5	0	1025	+1
Q_{12}	1634	991	221	18	1	4096	0

- table de valeurs (calcul avec arrondi) pour $\sigma = 2$

x	0	1	2	3	4	5	6	7	δ
$\mathbb{F}$	0.199	0.176	0.121	0.065	0.027	0.009	0.002	0.000	
Q_6	13	11	8	4	2	1	0		+1
Q_8	51	45	31	17	7	2	1		+1
Q_{10}	204	180	124	66	28	9	2		-2
Q_{12}	817	721	496	265	111	36	9	2	+2

- $\Rightarrow$ Ne pas oublier de compenser les coefficients

- mais comment faire si on veut $\sigma = 8$ et 3σ ? Prendre un filtre de rayon 24 ?
  $\Rightarrow$ prendre un filtre récursif.

Fonction de transfert

Transformée en Z :

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{k=N} b_k \cdot z^{-k}}{\sum_{k=1}^{k=M} a_k \cdot z^{-k}} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + + b_N z^{-N}}{1 + a_1 z^{-1} + a_1 z^{-2} + \cdots + + a_M z^{-M}}$$

Equation aux différences (z^{-1} code un retard) :

$$y(n) = \sum_{k=0}^{k=N} b_k \cdot x(n-k) - \sum_{k=1}^{k=M} a_k \cdot y(n-k)$$

Soit :

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) + \cdots + b_N x(n-N) - a_1 y(n-1) - a_2 y(n-2) - \cdots - a_m y(n-M)$$

Deux types de fonctions de transfert :

- FIR = Finite Impulse Response = filtre à réponse impulsionnelle finie
  - filtre non récursif $M = 0$, y ne dépend que de x
- IIR = Infinite Impulse Response = filtre à réponse impulsionnelle infinie
  - filtre récursif $M \neq 0$, y ne dépend que de x et de y

Filtres récursifs

- Un filtre très simple : $y(n) = \frac{1}{2}\left[x(n) + y(n-1)\right]$
- Quelques itérations en injectant $y(n-1)$ en fonction de $y(n-2)$ dans $y(n)$:
  - $y(n) = \frac{1}{2}\left[x(n) + y(n-1)\right]$ et $y(n-1) = \frac{1}{2}\left[x(n-1) + y(n-2)\right]$
  - $y(n) = \frac{1}{4}\left[2x(n) + x(n-1) + y(n-2)\right]$
  - $y(n) = \frac{1}{8}\left[4x(n) + 2x(n-1) + x(n-2) + y(n-3)\right]$
  - $y(n) = \frac{1}{16}\left[8x(n) + 4x(n-1) + 2x(n-2) + x(n-3) + y(n-4)\right]$
- C'est un filtre moyenneur à décroissance exponentielle
  $\Rightarrow$ on oublie le passé à une certaine vitesse
  - $y(n) = \frac{1}{2}x(n) + \frac{1}{4}x(n-1) + \frac{1}{8}x(n-2) + \frac{1}{16}x(n-3) + \frac{1}{16}y(n-4)$
    si l'entrée et la sortie sont sur 4 bits, alors dans cette expression du calcul $x(n-3$ et $y(n-4)$ ne participent plus au calcul

Filtre récursif d'ordre 1

Filtre de Shen-Castan

- fonction de transfert

$$H(z) = \frac{(1-\alpha)}{(1-\alpha)z^{-1}}$$

- soit

$$y(n) = (1-\alpha)x(n) + \alpha y(n-1)$$

- Le même qu'avant, avec en plus le paramètre $\alpha \in [0,1]$ pour régler la part de l'entrée courante $x(n)$ et la part du passée $y(n-1)$
  - si beaucoup de bruit, prendre α grand
  - si peu de bruit, prendre α petit

Filtre récursif d'ordre 2

Filtre de Federico Garcia Lorcal (FGL)

- fonction de transfert

$$H(z) = \left[\frac{(1-\gamma)}{(1-\gamma)z^{-1}}\right]^2$$

- soit

$$y(n) = (1-\gamma)^2 x(n) + 2\gamma y(n-1) - \gamma^2 y(n-2)$$

- avec $\gamma = e^{-\alpha}$

- filtre de F. Garcia-Lorca, approximation du filtre de Deriche

Fonctionnement des filtres

- Les filtres passe-bas (moyenneurs, binomiaux et gaussien) ne font pas disparaitre le bruit. D'un point de vue numérique, les filtres étalent le bruit. Et seulement si le bruit est d'une certaine nature, ils le font disparaitre (bruit blanc gaussien).

n	0	1	2	3	4	5	6	7	8	9	10
$x(n)$	1	1	1	4	1	1	1	-2	1	1	1
$y(n)$		1	2	2	2	1	0	0	0	1	1

n	0	1	2	3	4	5	6	7	8	9	10
$x(n)$	1	1	1	1	4	1	-2	1	1	1	1
$y(n)$		1	1	2	2	1	0	0	1	1	

Types de bruit

Deux principaux types de bruit :

- bruit additionnel = bruit blanc gaussien.
  - utilisation d'un générateur aléatoire gaussien d'écart type σ
  - le bruit est additionné (en signé) au signal.
  - comme sa moyenne est nulle, il y a – en moyenne – la même quantité de bruit qui est additionnée que celle qui est soustraite
  - $\Rightarrow$ utilisation de filtres moyenneurs (faible complexité de calcul)

- bruit impulsionnel = le signal en un point est détruit.
  - utilisation d'un générateur aléatoire qui va détruire un certain nombre d'échantillons qui seront remplacés par :
  - une valeur aléatoire,
  - le min ou le max codables
  - $\Rightarrow$ utilisation de filtre d'ordre comme le filtre médian (grande complexité de calcul)

Filtre d'ordre - filtre médian

Filtre médian de taille $k = 2r + 1$

- prendre une fenêtre de k échantillons de x et les copier dans un tableau temporaires de k cases
- trier le tableau
- écrire dans y le point se trouvant en $T[k/2]$ soit $T[r]$
- Le filtre supprime le bruit impulsionnel tant que le nombre de point faux reste inférieur à $k/2$ (plus précisément, r points faux plus petit et r points faux plus grands, rejetés aux deux extrémités du tableau.
- Le filtre médian est aussi très performant pour supprimer du bruit additionnel
  - Dans le second cas, un filtre de taille 3 ne suffit pas, il en faut un de taille 5

n	0	1	2	3	4	5	6	7	8	9	10
$x(n)$	1	1	1	4	1	1	1	-2	1	1	1
$y(n)$		1	1	1	1	1	1	1	1	1	

n	0	1	2	3	4	5	6	7	8	9	10
$x(n)$	1	1	1	4	1	-2	1	-3	1	1	1
$y(n)$		1	1	1	1	1	-2	1	1	1	

Filtre d'ordre *low cost*

- Le filtre médian peut avoir un coût trop élevé (trop de calculs pour produire un point en sortie)
- Dans certaines limites, on peut le remplacer par un filtre plus simple
- filtre moyenneur ($k \geq 5$) qui calcule à la volée, la somme des points ainsi que le min et le max, puis soustraire min et max et divise par $(k-2)$
- S'il n'y a au maximum un point faux plus petit et un point faux plus grand, le moyennage ne sera part perturbé par le bruit impulsionnel

Mesure qualitative

Soient les notations suivantes :

- x le signal initial de N échantillons,
- x_b le signal bruité (aussi noté $\widehat{x}$) : $x_b = x +$ bruit,
- x_r le signal débruité reconstruit (aussi noté $\widetilde{x}$) : $x_r =$ filtrage(x_b),

Il y a deux principales mesures pour évaluer la qualité de la reconstruction :

- eqm l'erreur quadratique moyenne : $eqm(x_r, x) = \frac{1}{N}\sum_{k=0}^{N-1}(x_r(i) - x(i))^2$,
- snr : le rapport signal sur bruit et en particulier
  $psnr$: le rapport pic signal sur bruit (en dB)

$$psnr(x_r, x) = 10log_{10}\left(\frac{d^2}{eqm(x_r, x)}\right)$$

,

  - avec d la valeur max du signal qui est souvent remplacée par la valeur max codable du signal quantifié : pour un codage sur 8 bits, $d = 2^8 - 1$
  - on peut aussi calculer le $pnsr(x_b, x)$ du signal bruité pour comparaison