Tutoriel sur un filtre passe-partout : le filtre RII du premier ordre*

J. Arzi
Courriel : contact AT tsdconseil.fr

14 mars 2016

Avant-propos

Ce filtre numérique très simple (un unique coefficient !), aussi appelé **filtre exponentiel** du fait de sa réponse impulsionnelle, est, de la même manière que son équivalent dans le domaine analogique (le filtre électrique RC), très facile à régler et à implémenter.

Du fait de son extrême simplicité, il est souvent dédaigné au profit de filtres plus sophistiqués, et pourtant il a beaucoup d'applications, et même il s'agit du filtre optimal pour certains types de problèmes. Nous présenterons dans ce petit tutoriel :

1. Comment on peut interpréter intuitivement ce filtre (point de vue lissage et point de vue contrôle),
2. Comment faire pour concevoir facilement ce filtre (choix du coefficient) en fonction de paramètres physiques concrets et intuitifs (fréquence de coupure, constante de temps),
3. Quelques contraintes et pièges à éviter lors de l'implémentation (en particulier en virgule fixe),
4. Comment comprendre ce filtre comme un filtre de Kalman (dans certaines conditions, c'est donc un estimateur optimal).

*La version la plus récente de ce document est disponible à cette adresse : http://www.tsdconseil.fr/tutos/index.html

Table des matières

1 Définition et interprétations

1.1 Première formulation : lissage

Une première manière d'écrire ce filtre sous forme d'équation est la suivante :

$$\boxed{y_n = (1-\gamma)y_{n-1} + \gamma x_n} \tag{1}$$

où x_n est le signal d'entrée, y_n est le signal en sortie de filtre, et γ est le coefficient (fixe) du filtre. Ce coefficient γ est parfois aussi appelé ≪ **coefficient d'oubli** ≫ : plus γ est proche de 1, plus le filtre ≪ oublie ≫ rapidement les ancienne données.

Dans cette interprétation, on considère le filtre comme un **lisseur** : chaque échantillon de sortie est une **moyenne pondérée** entre le dernier échantillon d'entrée et la dernière sortie calculée.

Le rapport avec un filtre RC électrique apparait clairement si on applique ce filtre sur un échelon inversé [1] ($x_{n\leq 0} = 1, x_{n>0} = 0$) :

$$y_n = (1-\gamma)y_{n-1} = (1-\gamma)^n y_0 = (1-\gamma)^n \text{ en supposant } y_0 = 1$$

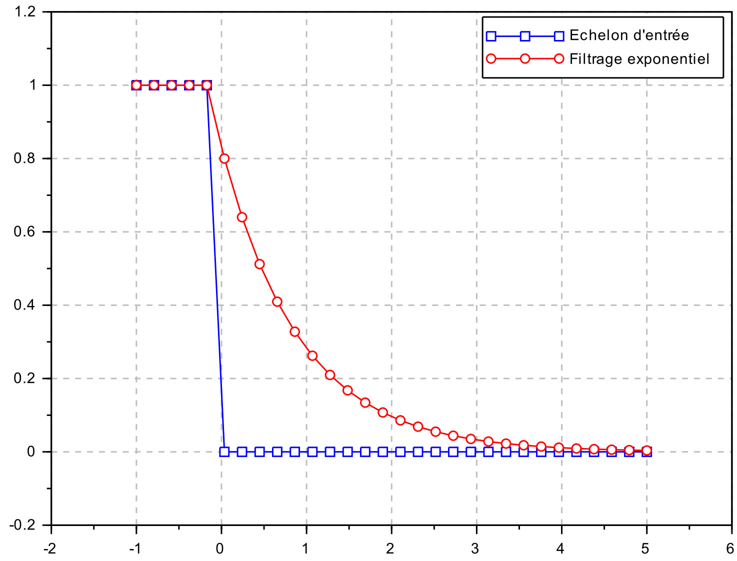

Figure 1 – *Filtrage d'un échelon (inversé)*

Autrement dit la réponse décroit suivant une loi exponentielle. Si on veut le mettre en forme comme pour un filtre analogique RC, il suffit d'écrire :

1. On a bien-sûr la même chose avec un échelon normal $x_{n\leq 0} = 0, x_{n>0} = 1$, mais le calcul est juste plus complexe.

$$(1-\gamma)^n = e^{n\log(1-\gamma)} = e^{-t\cdot\frac{-\log(1-\gamma)}{T_e}}$$

où T_e est la période d'échantillonnage. Le facteur $\frac{-T_e}{\log 1-\gamma}$ correspond donc à la constante de temps τ du filtre RC, et plus généralement (voir annexe A page 7), on peut calculer le coefficient d'oubli γ en fonction de la constante de temps τ souhaitée :

$$\boxed{\gamma = 1 - e^{-T_e/\tau}} \tag{2}$$

Alternativement, on peut aussi exprimer γ en fonction de la fréquence de coupure f_c souhaitée :

$$\boxed{\gamma = 1 - e^{-2\pi f_c/f_e}} \tag{3}$$

où f_e est la fréquence d'échantillonnage.

Ainsi, ce filtre est très pratique a régler à partir de paramètres intuitifs, et peut éventuellement être modifié dynamiquement.

1.2 Deuxième formulation : suivi

Une deuxième manière (complètement équivalente) d'écrire l'équation (1) est la suivante :

$$\boxed{y_n = y_{n-1} + \gamma(x_n - y_{n-1})} \tag{4}$$

Cette interprétation est celle dite de **contrôle** : x_n est un signal (bruité) à suivre, y_n est l'estimation courante, et $x_n - y_{n-1}$ est l'erreur de suivi. Autrement dit, γ peut être vu comme un gain sur l'erreur (ou rétro-action) : plus γ est élevé, plus le système réagit rapidement aux changement sur l'entrée.

Nous verrons à la fin de ce tutoriel que sous cette forme, le filtre RII est tout simplement un filtre de Kalman (pour un modèle très simple et en régime établi) !

2 Implémentation (en C)

Si on dispose d'un processeur supportant la **virgule flottante**, l'implémentation est triviale et correspond directement à l'équation (1) :

```
y = (1 - gamma) * y + gamma * x;
```

où `y` sert à la fois à d'accumulateur et de variable de sortie.

En revanche, pour une implémentation en **virgule fixe** (c'est-à-dire en utilisant uniquement des calculs en nombres entiers), le coefficient d'oubli γ va être représenté par un entier entre 0 et $N = 2^k$, et on considérera de manière conventionnelle que `gamma=`2^k correspond à $\gamma = 1$.

L'équation (1) devient alors (première implémentation naïve) :

```
y = ((N - gamma) * y + gamma * x) >> k;
```

Le décalage de k bits vers la droite (division par 2^k) permet de simuler une multiplication fractionnaire par γ à partir de sa représentation entière `gamma` (en effet, $\gamma = \frac{\texttt{gamma}}{2^k}$).

Le problème avec cette implémentation est que l'accumulateur (et variable de sortie) y est représenté avec la même précision que l'entrée x, si bien que les petites variations de y lors du suivi de l'entrée x sont tronquées à chaque itération, et comme, en fonction du coefficient d'oubli, la sortie y peut évoluer plus lentement que l'entrée x, le filtre peut très bien ne jamais converger vers x !

Pour résoudre ce problème, il suffit de représenter l'accumulateur en virgule fixe, comme la variable `gamma`. Par exemple, on peut ajouter k bits dans la représentation de y, et l'équation de mise à jour devient alors :

```
y = (((N - gamma) * y) >> k) + gamma * x;
```

3 Interprétation comme un filtre de Kalman

Le filtre de Kalman est une méthode très générique (et extensible) pour déterminer de manière optimale l'état (caché) d'un système à évolution linéaire et pour lequel on dispose d'observations bruitées (par un bruit que l'on suppose gaussien et sans mémoire). Le modèle général du problème est ce que l'on appelle la représentation d'état :

$$\begin{aligned} \mathbf{x}_{n+1} &= A\mathbf{x}_n + B\mathbf{u}_n + \mathbf{v}_n \\ \mathbf{y}_n &= C\mathbf{x}_n + D\mathbf{u}_n + \mathbf{w}_n \end{aligned}$$

où :

$\mathbf{x}_n$ est le vecteur d'état (variables cachées du système, que l'on cherche à déterminer, par exemple la position d'un robot),

$\mathbf{u}_n$ est le vecteur d'entrée / commande (entrée que l'on contrôle / connait, par exemple le courant que l'on injecte dans les moteurs),

$\mathbf{y}_n$ est le vecteur de sortie / observations (par exemple, les valeurs fournies par des accéléromètres / gyroscopes),

$\mathbf{v}_n, \mathbf{w}_n$ sont les bruits de process et d'observation (hypothèse : **bruit blanc gaussien**, de matrices covariance connues Q et R)

A, B, C, D sont les matrices caractérisant le système (e.g. système **linéaire**)

Sans entrer dans les détails de la théorie de Kalman, en régime asymptotique (après suffisamment d'itérations), pour un système stable et observable, l'équation pour résoudre ce système se ramène à :

$$\hat{x}_{n+1} = A\hat{x}_n + K \cdot (y_n - C\hat{x}_n) \tag{5}$$

Où K est un gain fixe (appelé gain asymptotique), qui peut être déterminé en fonction des matrices du systèmes (équation de Riccati). Sous cette forme, on reconnait quasiment notre filtre RII du premier ordre, sous la forme de l'équation (4) :

- $y_n - C\hat{x}_n$ est l'erreur de suivi (erreur entre les observations attendues et réelles),
- K correspond à notre facteur d'oubli γ

Bien-sûr, la formulation de Kalman est plus générale car elle permet de traiter de systèmes à plusieurs dimensions (A, K et C sont des matrices). Ceci-dit, dans le cas particulier d'un système scalaire :

$$x_n = x_{n-1} + v_n \tag{6}$$

$$y_n = y_{n-1} + w_n \tag{7}$$

C'est-à-dire, pour un système représentant une **marche aléatoire** (déplacement aléatoire de v_n à chaque itération), observée de manière imparfaite (bruit d'observation w_n), alors la solution asymptotique de Kalman coïncide exactement avec un filtre RII d'ordre 1 :

$$\hat{x}_{n+1} = \hat{x}_n + K \cdot (y_n - \hat{x}_n) \tag{8}$$

Autrement dit, *le simple filtre RII d'ordre 1 est le filtre optimal pour suivre une valeur scalaire soumise à une évolution aléatoire*, et de plus la théorie de Kalman (équations de Riccati) fournit les outils théoriques pour calculer le coefficient d'oubli optimal.

On trouve (voir annexe B page 8) :

$$K_{opt} = \frac{1}{1 + 1/X} \tag{9}$$

avec :

$$X = \frac{\sigma_v^2}{2\sigma_w^2}\left(1 + \sqrt{1 + 4\frac{\sigma_w^2}{\sigma_v^2}}\right) \tag{10}$$

où σ_v et σ_w sont respectivement les écarts type sur les bruits de processus (marche aléatoire) et d'observation. Notez que le rapport $\frac{\sigma_v^2}{\sigma_w^2}$ peut être interprété comme le rapport signal à bruit (SNR), σ_v déterminant l'évolution réelle du signal, et σ_w les erreurs de mesures.

Dans la figure ci-après (figure 3 page 7), on a tracé la valeur optimale du gain en fonction du SNR $\frac{\sigma_v^2}{\sigma_w^2}$. On peut noter sur cette figure quelques cas particuliers importants :

1. Si on observe une valeur qui varie lentement, mais avec un bruit d'observation important ($\sigma_v \ll \sigma_w$), on obtient un gain K_{opt} faible, ce qui se comprends dans la mesure où asymptotiquement (au bout d'un nombre suffisant d'itérations), on connait bien la valeur et on ne prends pas trop en compte les nouvelles observations.
2. Dans le cas inverse, si on observe une valeur qui varie rapidement, et avec bruit d'observation assez faible ($\sigma_v \gg \sigma_w$), on obtient K_{opt} proche de 1, c'est-à-dire que l'on tient compte surtout de la dernière observation, et moins des observations précédentes, ce qui est aussi logique.

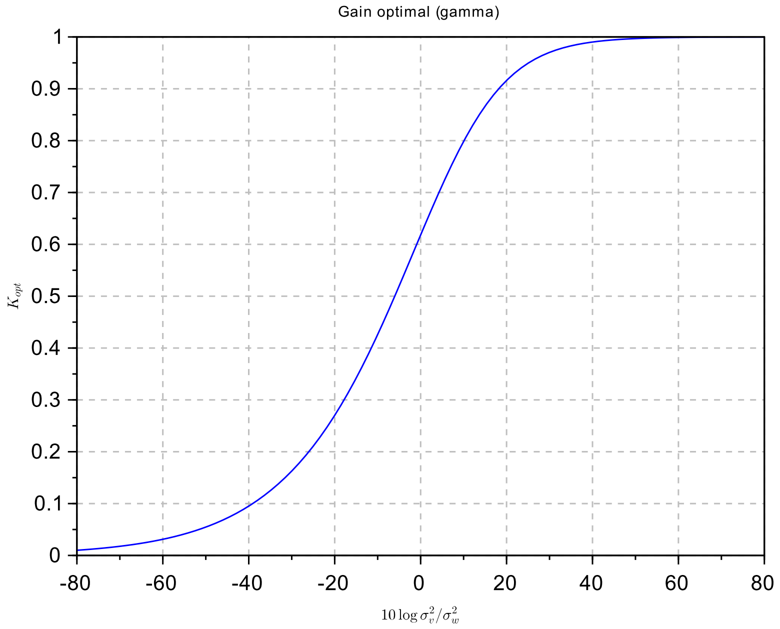

FIGURE 2 – *Gain optimal en fonction du rapport variance de processus sur bruit d'observation (≪ SNR ≫)*

Annexes

A Détermination du coefficient γ

Le coefficient γ (facteur d'oubli) peut être mis en rapport avec la constante du temps, en considérant la réponse à un échelon du filtre ($x_n = 1, y_0 = 0$) :

$$y_{n+1} = y_n + \gamma(1 - y_n)$$

Soit :

$$y_{n+1} - 1 = y_n(1 - \gamma) + \gamma - 1 = (1 - \gamma)(y_n - 1)$$

Et donc :

$$y_n = 1 - (1 - \gamma)^n \tag{11}$$

Pour un filtre RC analogique, la réponse est de type : $y(t) = 1 - e^{-t/\tau}$, où τ **est la constante de temps du filtre**, et en identifiant $t = nT_e$ (avec $T_e = 1/f_e$: période d'échantillonnage), on obtient :

$$(1 - \gamma)^n = e^{-nT_e/\tau}$$

Soit :

$$n \log 1 - \gamma = -nT_e/\tau$$

Et donc :

$$\boxed{\gamma = 1 - e^{-T_e/\tau}} \tag{12}$$

Alternativement, on peut aussi calculer le coefficient en fonction de la **fréquence de coupure souhaitée** $f_c = \frac{1}{2\pi\tau}$:

$$\boxed{\gamma = 1 - e^{-2\pi T_e f_c} = 1 - e^{-2\pi f_c/f_e}} \tag{13}$$

Inversement, on peut aussi déterminer la fréquence de coupure en fonction du coefficient :

$$f_c = \frac{-f_e \log(1-\gamma)}{2\pi} \tag{14}$$

B Calcul du gain optimal pour une marche aléatoire

Dans le cas général, l'équation de Riccati s'écrit (voir [1]) :

$$X = Q + AXA^T - A\left(I + \frac{1}{XC^T R^{-1} C}\right)^{-1} XA^T \tag{15}$$

et alors :

$$K_{opt} = \frac{XC^T}{CXC^T + R} \tag{16}$$

Dans notre cas très simple (scalaire), $A = C = 1$, $Q = \sigma_v^2$ et $R = \sigma_w^2$, et alors l'équation (15) devient :

$$\sigma_v^2 - X\frac{1}{1 + \frac{\sigma_w^2}{X}} = 0 \Leftrightarrow \sigma_v^2 = \frac{X^2}{\sigma_w^2 + X} \Leftrightarrow X^2 - \sigma_v^2 X - \sigma_v^2 \sigma_w^2 = 0$$

Soit :

$$X = \frac{\sigma_v^2}{2}\left(1 + \sqrt{1 + 4\frac{\sigma_w^2}{\sigma_v^2}}\right) \tag{17}$$

et :

$$K_{opt} = \frac{1}{1 + \sigma_w^2/X} \tag{18}$$

Références

[1] I. RIBEIRO, *Kalman and Extended Kalman Filters : Concept, Derivation and Properties*, 2004