

Chambre de Commerce et d'Industrie de Paris <hr/> ESIEE	Unité : ST4 TP filtrage adaptatif	SIG1
---	--------------------------------------	------

Remis par M. J.-F. BERCHER

ÉNONCÉ

Le but de ce TP est d'illustrer et consolider quelques concepts sur les problèmes d'estimation et filtrage adaptatif. On s'intéresse essentiellement à un problème de soustraction de bruit. Dans un premier temps, on étudie le problème de manière analytique, puis dans un second temps, on expérimente et étudie la problème par simulation. En particulier, et il faut bien le faire une fois, on programmera et on mettra en œuvre un algorithme LMS.

Pour récupérer les fichiers matlab, vous pouvez vous connecter sur

<http://www.esiee.fr/~bercherj/New/TP/TPAda> et téléchargez les fichiers dans un répertoire de travail local.

1 Exercice de mise en train

On considère un problème de soustraction de bruit où $u(n)$ est la référence bruit, entrée du filtre de Wiener. Le signal observé est $x(n) = s(n) + b(n)$. On suppose par ailleurs que l'on dispose d'un lien entre $u(n)$ et $b(n)$: $b(n) = \sum_{i=0}^p \alpha_i u(n-i)$.

1. Exprimer le vecteur d'intercorrélations \mathbf{R}_{ux} en fonction de la matrice de corrélation \mathbf{R}_{uu} et du vecteur de coefficients α ,
2. Quel est le filtre de Wiener optimal ?
3. Donner l'algorithme du gradient, puis l'algorithme LMS correspondant
4. Quel est le pas optimal théorique μ ? Que vaut-il dans le cas où $u(n)$ est un bruit blanc de variance σ^2 ?
5. Donner la première valeur du filtre $w(1)$, obtenue en choisissant des conditions initiales nulles.
6. Quel est ce premier filtre dans le cas où $u(n)$ est un bruit blanc de variance σ^2 et le pas le pas optimal ? Commenter.

2 Algorithme LMS

- Programmez un algorithme LMS. La syntaxe d'appel sera la suivante :

```
function [w, e] = lms(d, w, u, mu);
% fonction [w, e] = lms(d, w, u, mu);
% la recurrence sur le temps etant menee par appel a ce script.
%
% Variables d'entree :
% d : séquence désirée
% w : reponse impulsionnelle recherchee (a mettre a jour)
% u : entree
% mu : pas d'adaptation
% Variables de sortie :
% w : reponse impulsionnelle
% e : erreur
```

- Vous testerez cet algorithme sur *un problème d'identification*. Vous utiliserez comme signal test la sortie d'un filtre excité par un bruit blanc, selon, par exemple

```
x=randn(1,N);
htest=10*[1 0.7 0.7 0.7 0.3 0];
z=filter(htest, [1 ],x) + 0.01*randn(1,N);
```

1. Mettez en œuvre une procédure d'identification adaptative, en écrivant un script `ident.m` utilisant une boucle sur l'algorithme `lms` appelé avec les paramètres pertinents
2. Pour évaluer le comportement de l'algorithme, vous pourrez tracer une estimée de la variance de l'erreur, ainsi que l'erreur quadratique entre le filtre identifié et le filtre exact.
3. Vous proposerez une estimée de la variance de l'erreur utilisant un facteur d'oubli exponentiel, similaire à ce que l'on a vu dans le cadre des MCR – prendre par exemple un facteur d'oubli de 0.9 à 0.99. Examinez la vitesse de convergence en fonction du pas d'adaptation, et les capacités d'adaptation μ , en introduisant une non-stationnarité lente dans le signal, par exemple selon

```
u=randn(1,N)';
htest=10*[1 0.7 0.7 0.7 0.3 0]';
for t=L:N
y(t)=htest' .* (1+cos(2*pi*t/(N))) * u(t:-1:t-L+1);
end;
```

4. Fort aimablement, on vous fournit une implémentation de l'algorithme des moindres carrés récursifs. Reprenez alors les expérimentations précédentes. Comparez et concluez.

- **Soustraction de bruit.** Le signal que l'on cherche à estimer est s , à partir du mélange $x = s + b$, et de la référence bruit u . Vous cherchez à retrouver le signal s . Vous appliquerez donc une structure de soustraction de bruit, où le filtre sera identifié de manière adaptative à l'aide d'un algorithme LMS. Vous disposez pour vos expérimentations, de signaux tests contenus dans les fichiers `sb1.mat` et `sb2.mat`. Le premier fichier contient deux références bruit, l'une stationnaire, l'autre non stationnaire, que vous considérerez successivement. Vous prendrez des valeurs de pas comprises entre 0.01 et 1. Pour le second signal test, contenu dans le fichier `sb2.mat`, on a une non-stationnarité beaucoup plus importante et un rapport signal-à-bruit très défavorable. Il pourra être utile de traiter le problème en « deux passes », afin d'obtenir une première estimée de la réponse impulsionnelle du filtre, qui sera utilisée lors de la seconde « passe ».
- **Soustraction de bruit et signal de parole** On dispose d'un signal de parole perturbé par un cri d'insecte (il s'agit d'une decticelle bariolée). On dispose aussi du signal de l'insecte seul, pris par le deuxième microphone tout près de celui-ci.

```
%signaux disponibles
load parole_bruitee
load decticelle
```

Testez vos algorithmes sur ces signaux. Vous pourrez vous amuser à les écouter à l'aide de la fonction `soundsc`.

Syntaxe d'appel :

```
function [s,h]=sousb(ref,signal,h_ini,mu);
% fonction [s,h]=sousb(ref,signal,h_ini,mu);
% Algorithme de soustraction de bruit :
% Entrées :
% ref : reference bruit seul
% signal : voie signal (signal composite s +b, ou b est corrélé avec ref
% h_ini : réponse impulsionnelle initiale
% mu : pas d'adaptation
% Sortie :
% s : signal identifié par soustraction de bruit
% h : réponse impulsionnelle identifiée
%
```