

TP : Filtrage adaptatif (P.Jardin, décembre 2005)

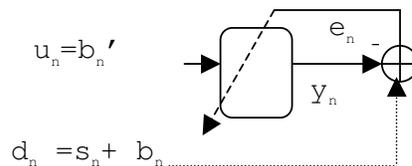
Objectifs du TP (sous Matlab): Implanter un filtre adaptatif (algorithme LMS) pour faire de la soustraction de bruit. Tester l'algorithme MCR.

On considère le problème de soustraction de bruit vu en TD :

On dispose d'un signal de parole (désolée ce n'est pas le début d'un conte merveilleux mais bon...) perturbé par un cri d'insecte (et là je vous ai gatés c'est une decticelle bariolée).

On dispose aussi du signal de l'insecte seul, pris par le deuxième microphone tout près de celui-ci.

Il s'agit de mettre en application vos connaissances en filtrage adaptatif pour restituer un signal de parole propre à partir des deux signaux disponibles. On utilise le schéma de principe suivant.



Le programme fourni *soustraction_bruit.m* effectue les opérations suivantes

- Il charge les signaux dans les vecteurs u et d . Vous pouvez les écouter à l'aide de la fonction `soundsc`
- Il calcule la puissance du signal u qui peut être utilisée pour déterminer le pas d'adaptation de l'algorithme
- Il appelle la fonction *algo_LMS* que vous devrez écrire
- Il appelle la fonction *algo_MCR* gracieusement fournie

Travail à effectuer

- 1) Ecrire l'algorithme du LMS sous forme d'une fonction matlab *algo_LMS.m* qui prend en entrée les signaux u_n et d_n (vecteurs) ainsi que le pas d'adaptation μ et la dimension M du filtre et fournit en sortie l'erreur e_n (vecteur) et la suite des filtres $\underline{w}^{(n)}$ (matrice) obtenus au cours de l'adaptation
- 2) Tester votre algorithme en jouant sur le pas d'adaptation μ (fixe).
 - a. Tracer l'erreur e_n , les coefficients du filtre $\underline{w}^{(n)}$ et la norme de l'erreur sur le filtre $\underline{w}^{(n)} - \underline{w}_{opt}$ sur le filtre
 - b. Quelle est la valeur maximale de ce pas pour que l'algorithme converge ?
 - c. Quelle est d'après vous la valeur optimale de ce pas ?
- 3) Tester votre algorithme avec un nombre de coefficients de filtre M différent de l'optimum $M=3$ ($M=2$ et $M=5$ par exemple)
- 4) Etudier et commenter l'algorithme MCR fourni
- 5) Tester cet algorithme en jouant sur le facteur d'oubli λ
 - a. Tracer l'erreur e_n , les coefficients du filtre $\underline{w}^{(n)}$ et la norme de l'erreur sur le filtre $\underline{w}^{(n)} - \underline{w}_{opt}$ sur le filtre
 - b. est il nécessaire ici d'introduire un facteur d'oubli ? pourquoi ?
- 6) Tester cet algorithme pour faire de la prédiction linéaire d'ordre $M=10$ sur le signal de parole $s = e$ (de l'étape précédente). Le filtre optimal vous paraît il stationnaire dans ce cas ?