# TP1 : Annulation d'écho acoustique

# LMS

13.10.2025

## I. IMPLÉMENTATION DE L'ALGORITHME LMS

### 1) Préparation

L'algorithme LMS ajuste les coefficients du filtre pour minimiser l'erreur quadratique moyenne.

Les équations de base de l'algorithme LMS sont les suivantes :

LMS.

• Signal d'entrée.

$$x(n,w),\ n \in [\![ 0, N-1 ]\!].$$

• Initialisation :

$$\underline{w}^{(0)} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \updownarrow M$$

• Itérations :

$\forall n \in [\![ M-1, N-1 ]\!]$

$$\rightarrow \underline{x}(n) = \begin{pmatrix} x(n) \\ \vdots \\ x(n-M+1) \end{pmatrix} \updownarrow M$$

$$\rightarrow e^{(m)} = \overbrace{\underline{w}^{(m)+} \underline{x}(n)}^{y(n)} - d(n)$$

$$\rightarrow \underline{w}^{(m+1)} = \underline{w}^{(m)} - \mu \bar{e}^{(n)} \underline{x}(n)$$

Ziqian Peng, Kadidiatou Diallo, 3A SIA

## 2) Génération de signaux test

Dans un premier temps le signal x est un bruit blanc et le signal représentant d est obtenu par filtrage de x par le filtre de réponse impulsionnelle finie h = [1 0.3 − 0.1 0.2] supposé inconnu.

## 3) Mise en œuvre de l’algorithme LMS

- La fonction algolms prend en entrée x, d, M l’ordre du spectre et l’écart-type mu et qui renvoie, pour chaque itération : le filtre W, le signal de sortie y et l’erreur e.
- Ici nous avons défini un matrice de dimension MxN, où chaque colonne représente une mise à jour du filtre. Cela nous permettra de visualiser l’évolution des coefficients au fil des itérations.

## 4) Validation de l’algorithme LMS

Considérer le signal généré au point 2 et utiliser l’algorithme LMS pour calculer le filtre et le signal de sortie. Que doit être wopt ?

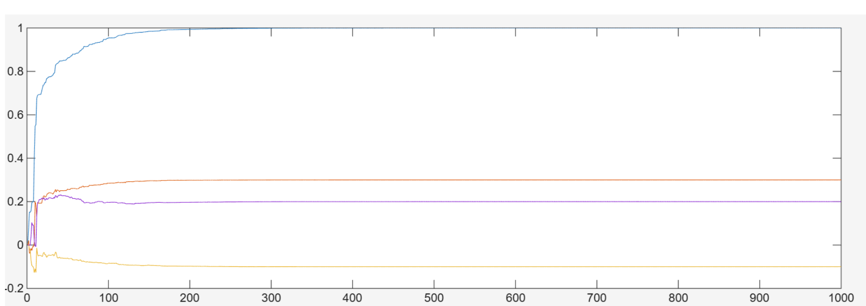
- Ici il n’y a aucune partie du signal d’entrée qui n’est pas corrélée à x (pas de bruit) donc d est entièrement déterminé par les valeurs passées de x, donc on s’attend à une erreur nulle ainsi que des coefficients après convergence qui soient égaux aux coefficients utilisés pour générer d, à savoir ceux de h (cas **Wopt** = α vu en classe).

- Lorsqu’on lit les valeurs de la dernière colonne de W, c’est bien ce qu’on obtient :

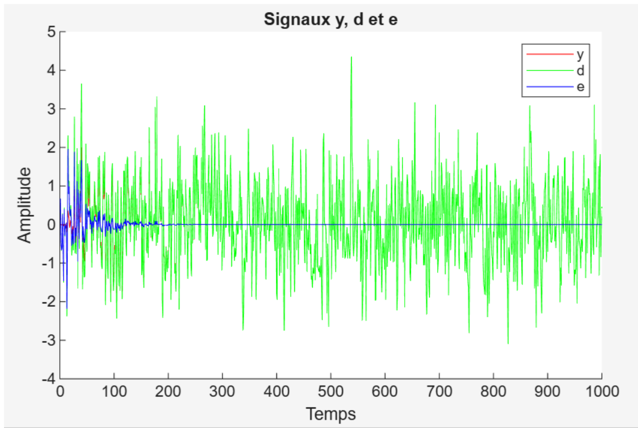
```
wfin =

    1.0000
    0.3000
   -0.1000
    0.2000
```

- Ensuite, on trace l’évolution des coefficients au fil des itérations. Le tracé permet de nous assurer de la convergence pour valider notre fonction LMS, et donne une idée de la vitesse de convergence pour les paramètres choisis pour commencer, M=4 et mu = 0.02.

  Ici on voit bien que tous les coefficients convergent, à partir de 100 itérations, vers les coefficients de h.



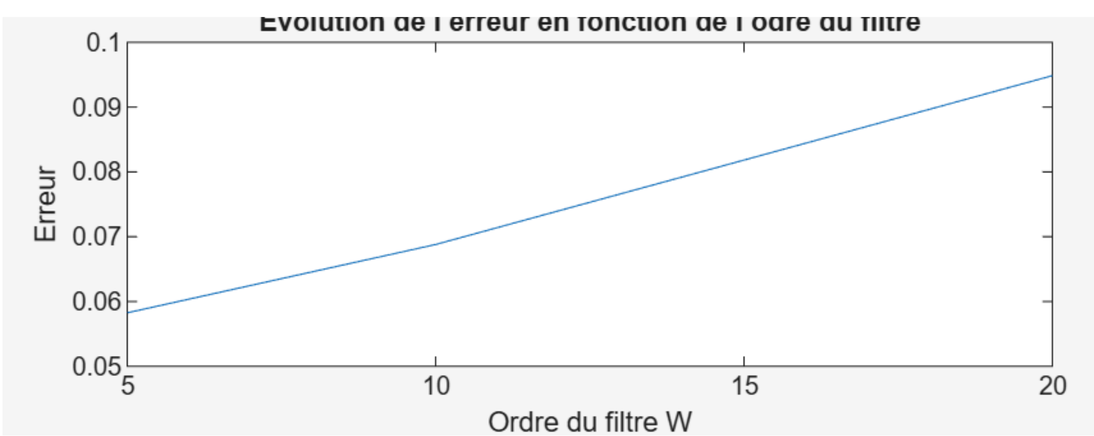
- Nous avons tracé aussi la sortie du filtre, la référence et l'erreur sur la même figure. On voit bien que l'erreur converge vers 0 (attendu puisque la référence d est entièrement corrélée avec les valeurs passées de x ! donc le filtre parvient à recréer entièrement d à partir des x).




## 5) Test de l'algorithme LMS

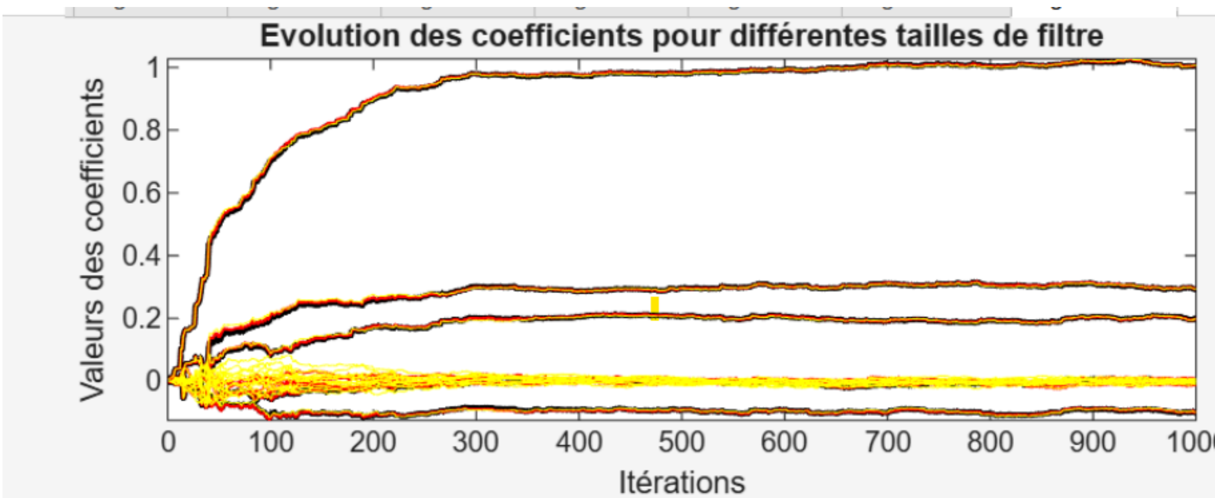
Cette fois, on crée d à partir de x plus un bruit. Donc l'erreur ne sera pas nulle.

- On trace l'évolution de l'erreur pour différents ordre du filtre W :



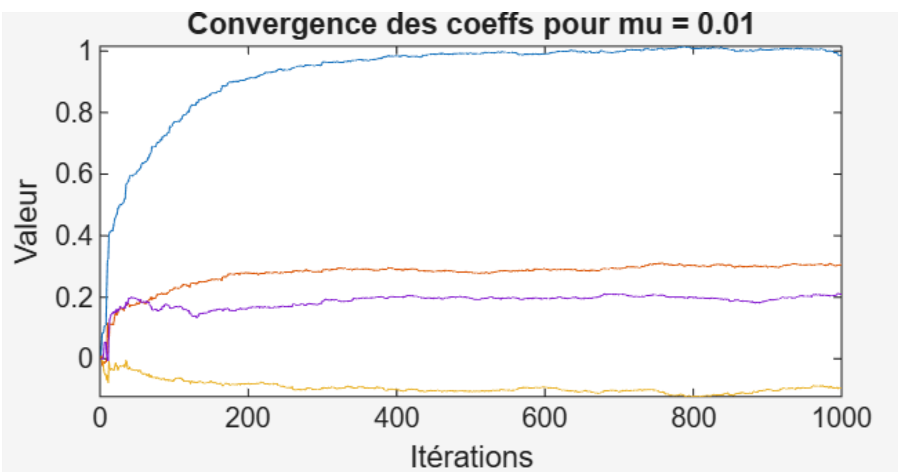

- Pour commencer on peut remarquer que l'erreur, bien que assez faible, n'est pas nulle puisque cette fois d n'est pas entièrement déterminé par les valeurs passées de x. Ensuite, on observe que pour un ordre qui varie entre 5 et 20, l'erreur augmente avec l'ordre du filtre.
- Ici d a été construit à partir d'un filtre de 4 coefficients. Donc l'ordre 5 suffit largement pour reconstruire d, les coefficients supérieurs à 4 seront négligeables. Mais nous avons du mal à comprendre l'augmentation de l'erreur.

- On trace aussi l'évolution des coefficient pour comparer ce qu'il se produit pour différentes tailles de filtre (en noir 5, en rouge 10 et en jaune 20)

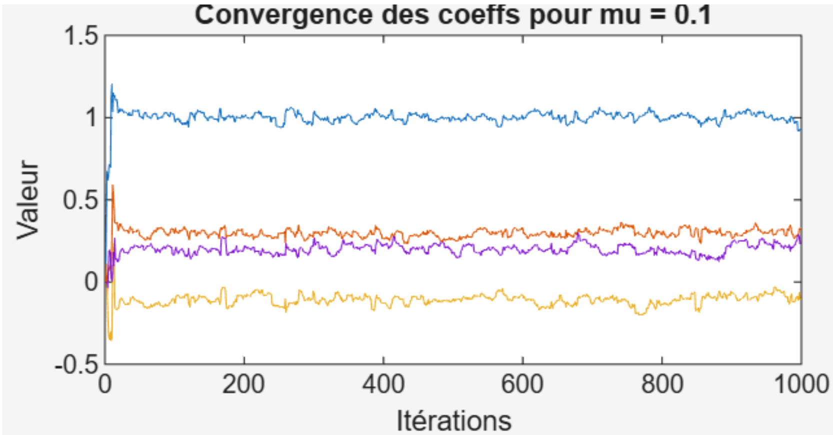



- Les tracés confirment notre intuition : au delà du quatrième indice, les coefficients convergent très rapidement pour vers 0 (on le voit sur la figure, les traités en rouge et jaune au voisinage de 0 correspondent aux coefficients des W d'indice supérieur à 5). Donc il est pas étonnant non plus d'observer que l'ordre du filtre n'impacte pas la vitesse de convergence : les tracés se superposent pour tous les ordres, et les coefficients convergent de la même façon, avec la même vitesse de convergence.

- Pour M fixé, on étudie l'effet du pas sur la vitesse de convergence.

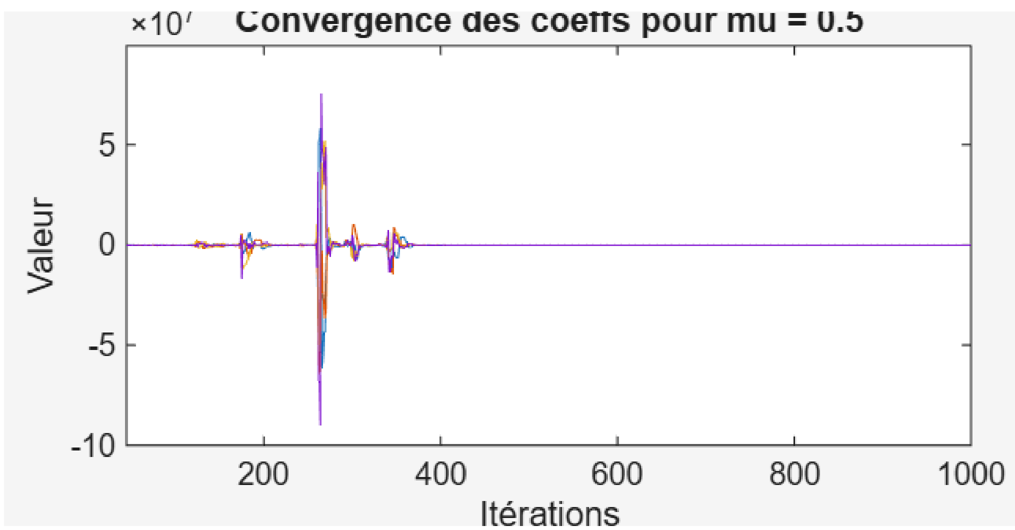



On remarque pour des pas de 0.01, la convergence se fait au bout de 200 itérations, mais il y a peu de gigue statistique.




Pour un pas de 0.1, la convergence intervient avant les 200 itérations, on peut dire qu'il y a une convergence moyenne, mais pas une véritable convergence car il y a beaucoup de gigue statistique.

- Cette observation est cohérente puisque nous avons vu en cours que la gigue est proportionnelle au pas. Il y a donc un compromis à faire entre le temps de convergence (qui diminue en augmentant le pas à condition de respecter les conditions nécessaires de convergence) et la gigue (qui augmente en augmentant le pas).
- Par contre, pour mu = 0.5, les coefficients n'ont plus trop de sens. J'imagine que le pas est trop grand pour trouver le minimum?



# II. Application