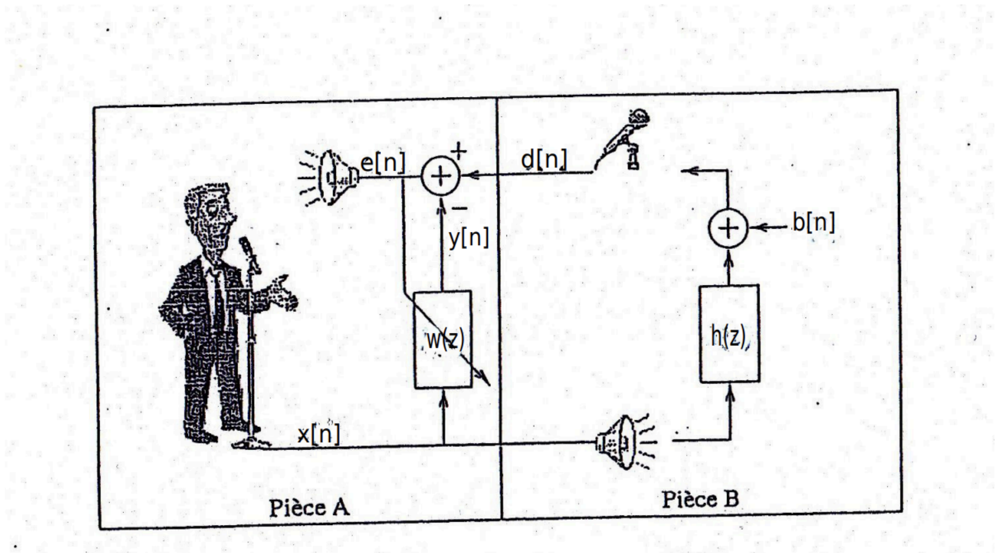
DU Jack
LAROYE Ambroise
SIA

TP2 : Méthodes de Signal Avancées Annulation d'Écho Acoustique

Introduction :



Un exemple simple d'annulation d'écho acoustique est donné dans le cas d'une téléconférence dans laquelle un locuteur parle dans la pièce A et le haut-parleur de la pièce B émet le signal x[n]. Le microphone de la pièce B reçoit un version filtrée et bruité de x[n]. Étant directement relié au haut-parleur de la pièce A, le locuteur va donc s'entendre parler. Pour éviter cela, on estime de manière adaptative le filtre h par le filtre w et on envoie sur le haut-parleur de la pièce A uniquement l'erreur commise e[n].

Objectifs : Dans ce TP nous cherchons à adapter les paramètres d'optimisation de l'algorithme LMS. Par analogie nous chercherons à comprendre **comment marche l'algorithme de la descente de gradient utilisé dans l'apprentissage profond**.

DU Jack
LAROYE Ambroise
SIA

1. Implémentation de l'algorithme LMS

1. Rappeler les équations de l'algorithme LMS.

L'idée est d'otenir $J(\mathbf{w}) \to \tilde{J}(\mathbf{w}) = |e_c(n,\omega)|^2$ (fonction de la réalisation w) tout en gardant la mémoire des instants précédents.

— $\mathbf{w}^{(0)}$ initialisation

— à l'instant n : $\overline{\mathbf{w}}^{(\mathbf{n})} = \overline{\mathbf{w}}^{(\mathbf{n-1})} - \mu grad_{\mathbf{w}} \tilde{J}(\mathbf{w})\Big|_{\mathbf{w}=\mathbf{w}^{n-1}}$

Pour une réalisation donnée :

$$\tilde{J}(\mathbf{w}) = \mathbf{w}^{\mathbf{t}}\mathbf{x}(\mathbf{n},\omega)\mathbf{x}^{\dagger}(\mathbf{n},\omega)\overline{\mathbf{w}} - \mathbf{w}^{\mathbf{t}}\mathbf{x}(\mathbf{n},\omega)\overline{d_c(n,\omega)} - d_c(n,\omega)\mathbf{x}^{\dagger}(\mathbf{n},\omega)\overline{\mathbf{w}} + |d_c(n,\omega)|^2$$

$$\begin{aligned} grad_{\overline{\mathbf{w}}} J(\tilde{w}) &= 2(\mathbf{x}(n,\omega)\mathbf{x}^{\dagger}(n,\omega) - \mathbf{x}(n,\omega)\overline{d_c(n,\omega)}) \\ &= 2(R_x\overline{\mathbf{w}} - \gamma_{\mathbf{x},\mathbf{d}}) \end{aligned}$$

$$\begin{aligned} grad_{\overline{\mathbf{w}}} J(\tilde{w}) &= 2\mathbf{x}(n,\omega)(\mathbf{x}^{\dagger}(n,\omega) - \overline{d_c(n,\omega)}) \\ &= \overline{\mathbf{w}^t\mathbf{x}(n,\omega) - d_c(n,\omega)} \end{aligned}$$

Algorithme Least Mean Square (LMS) s'exprime donc tel que :
à l'instant n :

$$\begin{aligned} e_c^{(n)} &= \mathbf{w}^{(n-1)t}\mathbf{x}(n,\omega) - d_c(n,\omega) \\ \overline{\mathbf{w}}^{(n)} &= \overline{\mathbf{w}}^{(n-1)} - \mu\overline{e_c^{(n)}(n,\omega)}\mathbf{x}(n,\omega) \end{aligned}$$

Il y a une mise à jour dans la direction du vecteur de régression.

Concernant les performances, il faut trouver un compromis sur le pas μ :

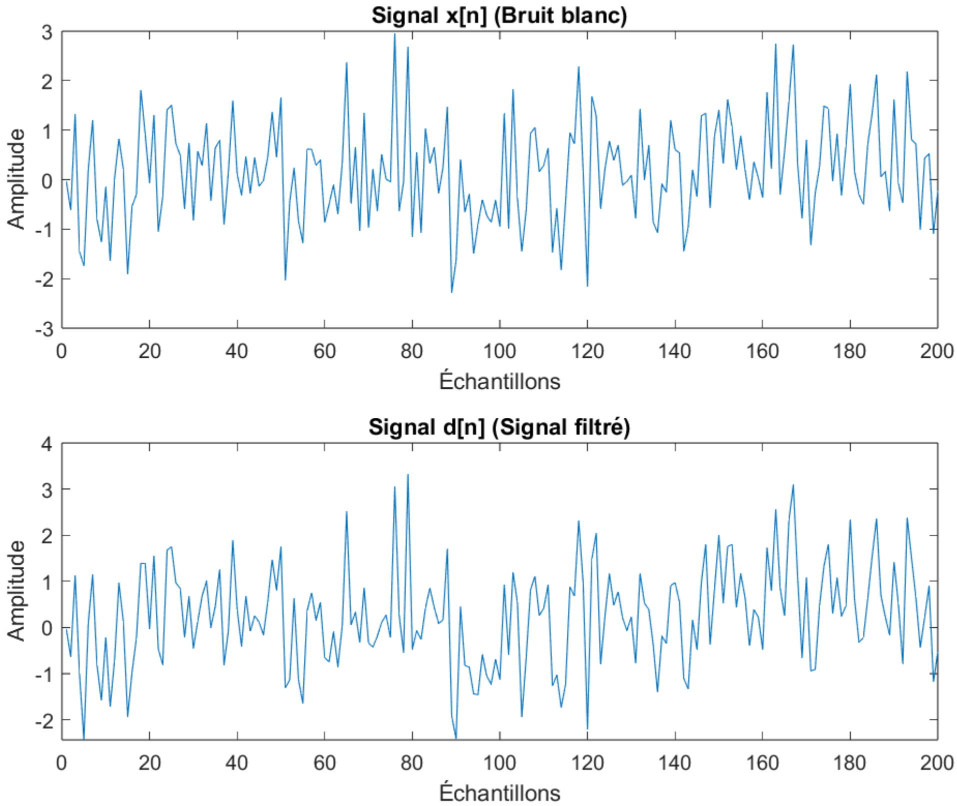
1. μ très faible : convergence lente, moins d'adaptation aux variations des statiques des signaux.
2. μ assez grand mais pas trop, pour qu'il y ait toujours convergence : bruit trop important
3. μ trop grand : divergence

Dans notre cas nous prendrons $\mu = 0.01$

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 = $[1 \quad 0.3 \quad -0.1 \; 0.2]^t$ supposé inconnu.

Génération de x et d :



L'effet du filtrage par h du bruit blanc amplifie l'amplitude négative du signal x[n] (bruit blanc).

3. Mise en oeuvre de l'algorithme LMS

Dans la mise en œuvre de la fonction qui permet de calculer la sortie y, l'erreur e et les coefficient du filtre w.

Principe de la fonction :

Nous prenons une fenêtre de x[n] sur lequel nous appliquons :

$$y(n, w) = w^t x(n, w)$$

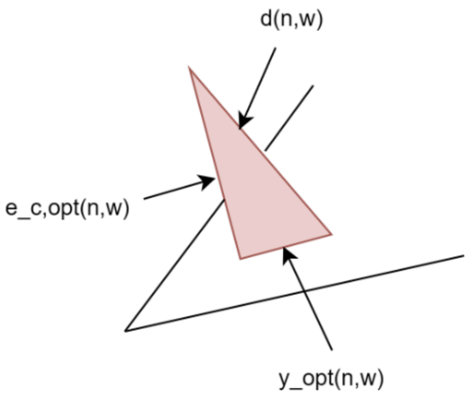
Nous avons : $e(n, w) = d(n, w) - y(n, w)$

et $\overline{\mathbf{w}}^{(n)} = \overline{\mathbf{w}}^{(n-1)} - \mu \overline{e_c^{(n)}(n, \omega)} \mathbf{x}(n, \omega)$

DU Jack
LAROYE Ambroise
SIA

4. Validation de l'algorithme LMS

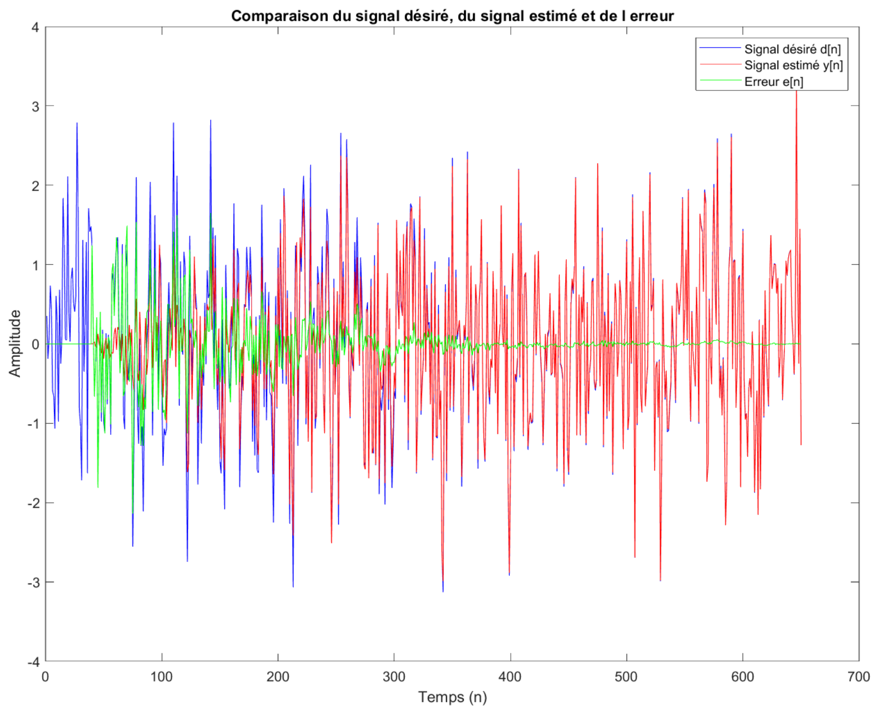
Définition de w_{opt} :



$\forall k \in D \subset Z$

$$e_{c,opt}(n,\omega) \perp x_n \Leftrightarrow \Gamma_{opt,x}(n, n-k) = E[e_{c,opt}(n,\omega)\overline{x_c(n-k,w)}] = 0$$

$$y_{opt}(n,\omega) = \sum_k w_{opt,k}x(n-k,\omega) = x(n+1,\omega)$$

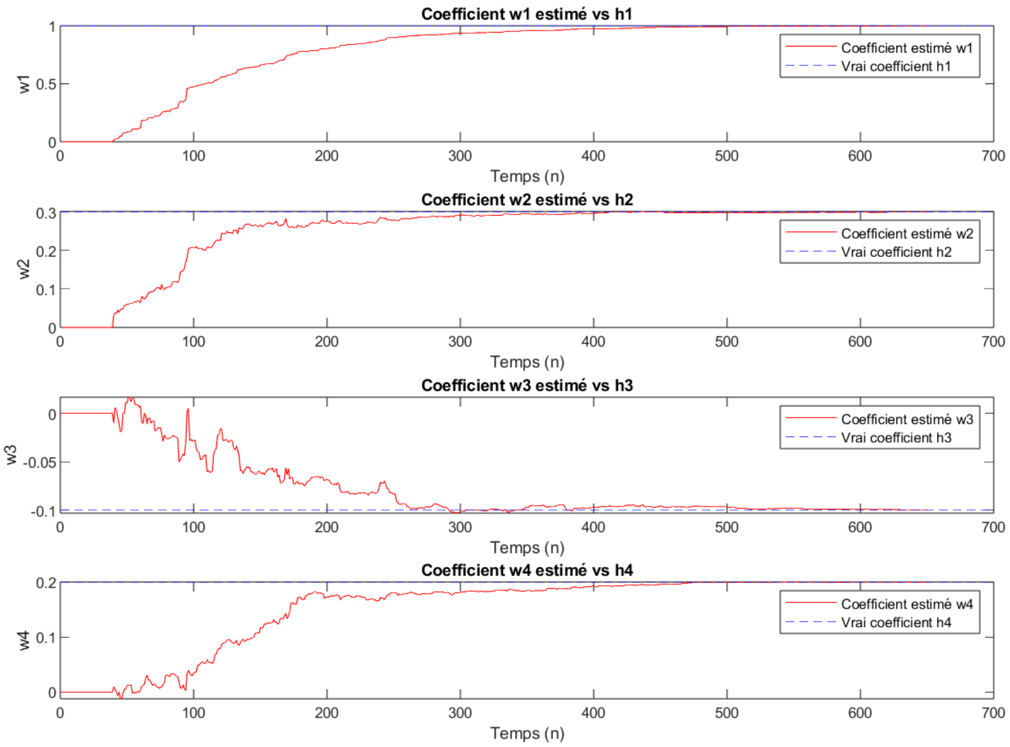


Nous obtenons bien le résultat attendu par l'algorithme LMS.
L'erreur tend vers 0, ce qui confirme que y et d se superposent à un moment donné. On a bien le signal estimé tend vers la signal désiré.

<u>Conclusion</u> : Algorithme LMS **VALIDÉ** ✅

DU Jack
LAROYE Ambroise
SIA

Tracé des coefficients et les estimation obtenue par le LMS:



Nous observons qu'il y a convergence des coefficients vers les valeurs optimales données au début du TP.

Nous pouvons relever les asymptotes suivantes :

$w_1 \ -> \ 1$

$w_2 \ -> \ 0.3$

$w_3 \ -> \ -0.1$

$w_4 \ -> \ 0.2$

Soit le **filtre optimal** défini par w_{opt} **=** h = $[1 \quad 0.3 \quad -0.1 \ \ 0.2]^t$

DU Jack
LAROYE Ambroise
SIA

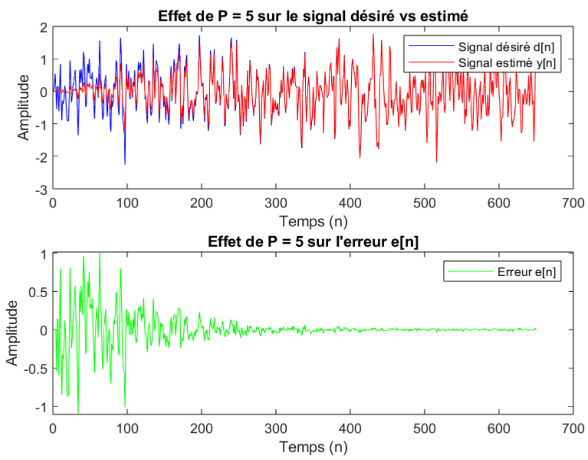
5. Test de l’algorithme LMS

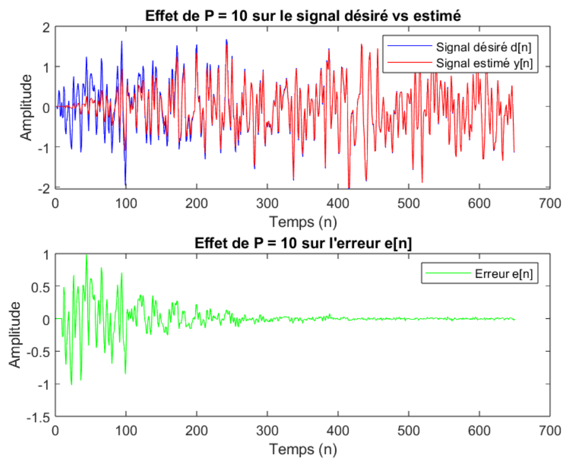
Dans cette partie nous cherchons à évaluer les performances et la robustesse de notre algorithme. Pour cela nous ajoutons un bruit b[n] au signal x[n] filtré.

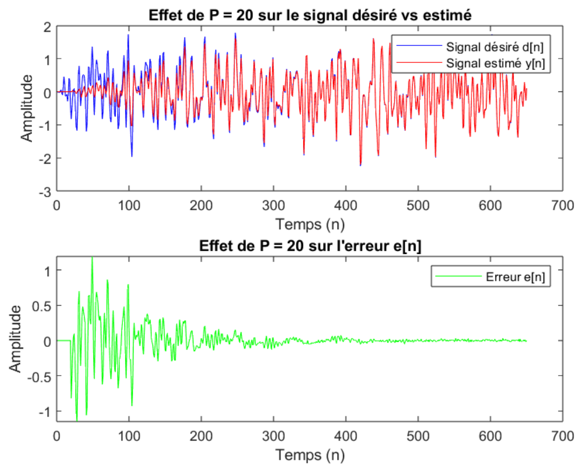
Pour différentes valeurs de P :

Ce que nous pouvons en tirer : Lorsque **P augmente**

Précision	plus précis (une meilleure résolution fréquentielle)
Temps de convergence	convergence plus lente (lié à la complexité)
Complexité	augmente (en effet il y a plus de coefficient)



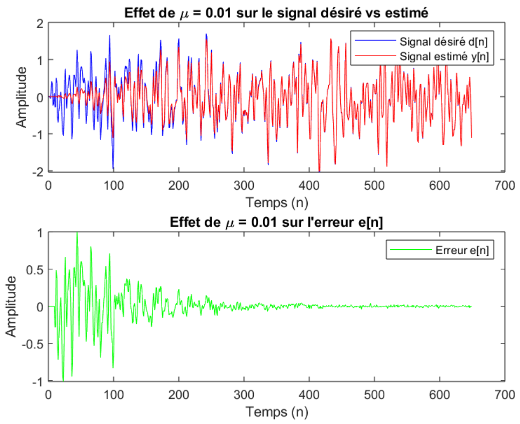


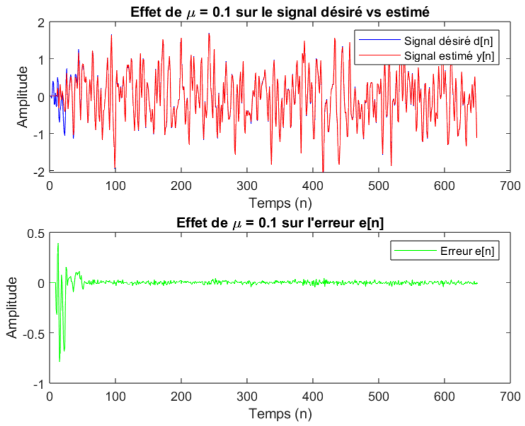


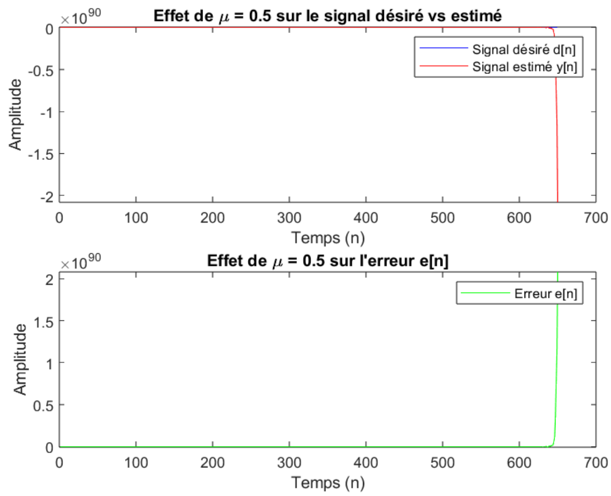
Le choix le plus judicieux pour P est un choix qui prend en compte les critères précédents. Ainsi un ordre de filtre **P=10** semble adapté.

DU Jack
LAROYE Ambroise
SIA

Pour différentes valeurs de μ :







Comme évoqué dans l'algorithme LMS :

Concernant les performances, il faut trouver un compromis sur le pas μ :

1. μ très faible : convergence lente, moins d'adaptation aux variations des statiques des signaux.
2. μ assez grand mais pas trop, pour qu'il y ait toujours convergence : bruit trop important
3. μ trop grand : divergence

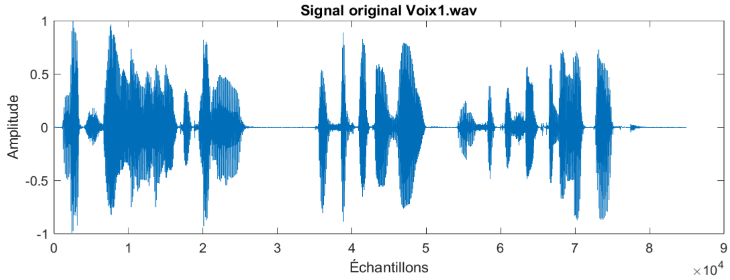
Ici nous mettons en application ces 3 cas. En effet pour μ = 0.01 le temps de convergence est plus élevé, et pour μ = 0.5 il y a **divergence**.

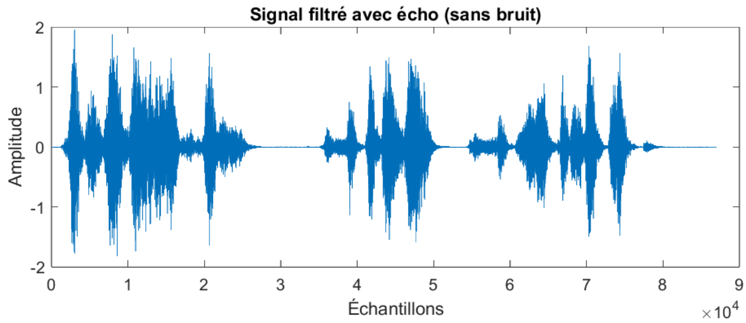
→ La valeur μ **= 0.1** semble être un bon compromis.

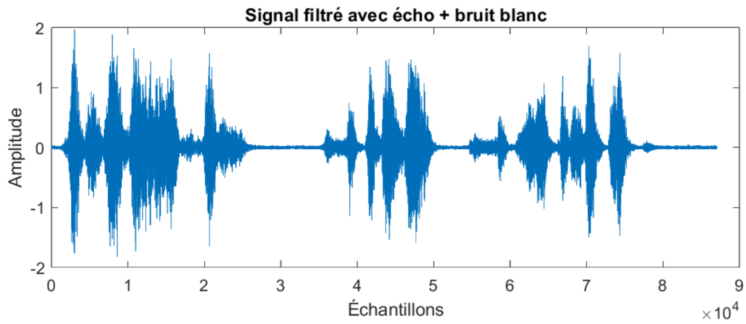
DU Jack
LAROYE Ambroise
SIA

1. Application

1. Signal audio avec une voix







Dans les figures précédentes nous affichons les différentes étapes qui nous permettent d'obtenir un annulation d'écho grâce à l'algorithme LMS.

En faisant varier les paramètres P et μ nous écoutons les différences et essayons de choisir les valeurs pour lesquels l'écho est au annulé au mieux. Et avec une retranscription idéale et sans bruit. Nous obtenons alors les paramètres de l'algorithme LMS : **P = 15 et μ = 0.1**.
Il s'agit bien de ce que nous avions avancé dans la partie précédente.

2. Signal audio avec deux voix

Dans cette partie nous nous concentrons sur 2 signaux :
- voix éloigné à laquelle nous ajoutons un écho (1)
- voix proche que l'on cherche à isoler (2)

DU Jack
LAROYE Ambroise
SIA

Lorsque nous superposons les deux voix, il semble très difficile de percevoir le de comprendre la voix 1 ou 2. L'objectif va donc être d'appliquer l'algorithme LMS de manière à isoler la voix 2.

Résultat :
Nous arrivons bien à atténuer l'écho et la voix 1. Nous pouvons tout de même remarquer que lorsque la voix féminine (proche) est isolée nous pouvons entendre des atténuations d'amplitude par moment. Cela correspond au moment où la voix avec écho au loin est annulée.

Conclusion :
En simulant un dialogue entre 2 personnes nous arrivons bien à isoler la voix avec écho malgré de légères pertes dans la voix isolé 2.

<u>Conclusion</u>

L'algorithme LMS est un choix largement utilisé en traitement du signal notamment pour les scénarios de filtrage adaptatif comme nous venons de le voir. Dans notre cas, nous nous sommes intéressés à l'annulation d'écho.

L'algorithme LMS est un algorithme relativement rapide et peu coûteux en termes de calcul, ce qui est plutôt pratique pour le temps réel comme vu dans l'exemple de ce TP.

Nous avons vu comment obtenir progressivement les coefficients pour annuler les composantes de d'écho dans le signal observé. Et comment choisir le P et μ, en fonction d'un trade off entre rapidité et performance.

Cependant l'algorithme LMS présente des limites d'utilisation. Nous avons vu dans le cours que si l'écho présente des non-linéarité ou des réverbération complexes l'algorithme LMS n'est plus suffisant et il nous faudra utiliser des filtre plus avancé comme le RLS.

L'étude de l'algorithme LMS, nous a également permis de mieux appréhender comment marche l'algorithme de la descente de gradient utilisé en l'apprentissage profond.