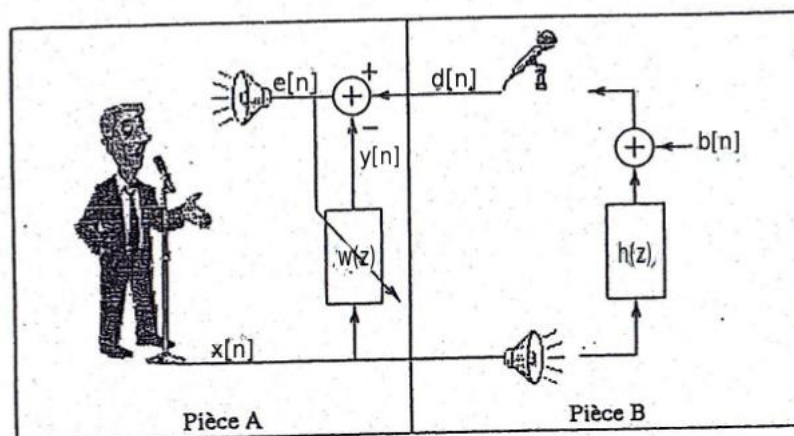


# TP : SIGNAL AVANCE

## Annulation d'Écho Acoustique

### Table des matières

|  |          |
|--|----------|
| <b>PARTIE 1 : IMPLEMENTATION DE L'ALGORITHME LMS .....</b> | <b>2</b> |
| SOUS-PARTIE 1.1 : PREPARATION .....                        | 2        |
| SOUS-PARTIE 1.2 : GENERATION DE SIGNAUX TESTS .....        | 2        |
| SOUS-PARTIE 1.3 : MISE EN ŒUVRE DE L'ALGORITHME LMS .....  | 3        |
| SOUS-PARTIE 1.4 : VALIDATION DE L'ALGORITHME LMS.....      | 3        |
| SOUS-PARTIE 1.5 : TEST DE L'ALGORITHME LMS.....            | 5        |
| <b>PARTIE 2 : APPLICATION .....</b>                        | <b>7</b> |
| SOUS-PARTIE 2.1 : SIGNAL AUDIO AVEC UNE VOIX.....          | 7        |
| SOUS-PARTIE 2.2 : SIGNAL AUDIO AVEC DEUX VOIX .....        | 8        |



Problème de Wiener étudié durant ce TP : Annulation d'écho via LMS

# Partie 1 : Implémentation de l'algorithme LMS

## Sous-partie 1.1 : Préparation

L'algorithme LMS s'implémente de la manière suivante :

### Initialisation :

- $\underline{W}^{(0)}$  (vecteur colonne de taille  $M$ )

### Mise à jour à chaque instant $n$ :

- $e_n^{(n)} = \underline{W}^{(n)\dagger} \underline{x}_n(\omega) - d_c(n, \omega)$  (scalaire)
- $\underline{W}^{(n+1)} = \underline{W}^{(n)} - \mu \times e_n^{(n)} \underline{x}_n(\omega)$  (vecteur colonne de taille  $M$ )

## Sous-partie 1.2 : Génération de signaux tests

Pour générer notre bruit  $x$ , on va créer un vecteur rempli par des valeurs prises aléatoirement suivant une loi normale centrée (car la voix dans le format .wave est un signal centré et symétrique vis-à-vis de 0) et de variance  $\sigma^2$  dont la valeur sera prise arbitrairement.

La génération de notre signal désiré  $d$ , on utilisera simplement la fonction *filter* entre le filtre  $h$  donné dans l'énoncé et notre bruit blanc.

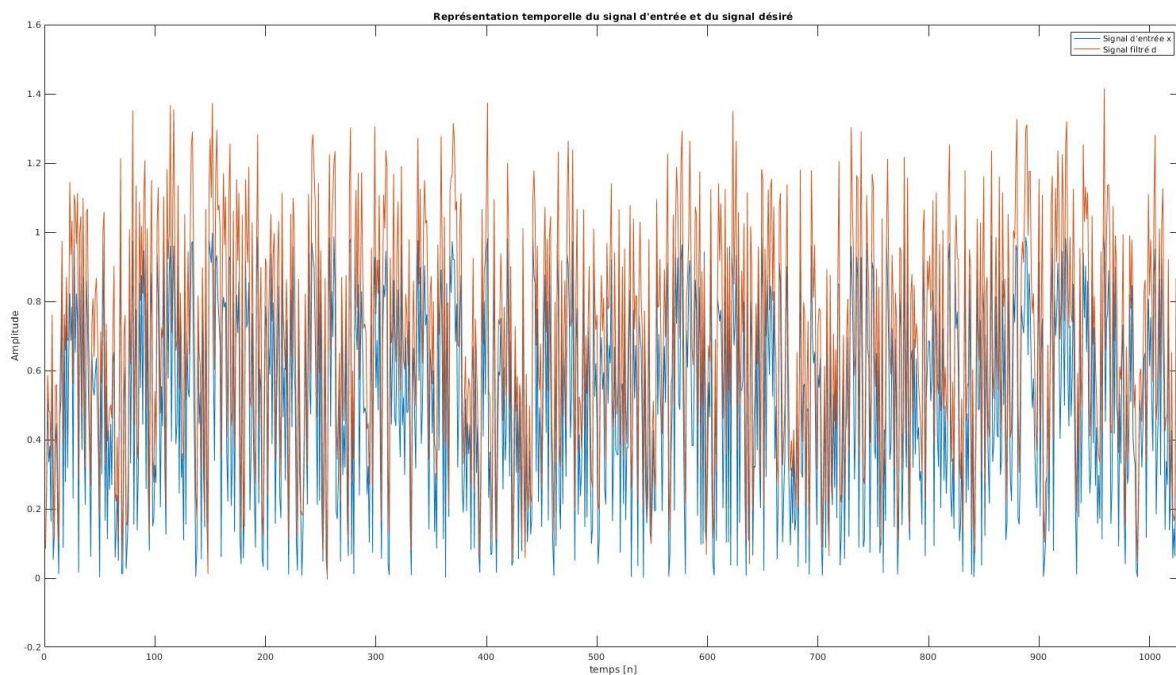


FIGURE 1 : Représentation temporelle du signal d'entrée et du signal désiré

## Sous-partie 1.3 : Mise en œuvre de l'algorithme LMS

Voir Code.

## Sous-partie 1.4 : Validation de l'algorithme LMS

Dans le cadre de notre problème, on a démontré en cours que le filtre optimal dans le cadre d'un problème de Wiener doublement infini était exactement le filtre des échos lorsque les l'interspectre entre  $x$  et  $z$  est nul i.e. l'intercorrélacion entre  $x$  et  $z$  est nulle i.e. si les deux signaux sont décorrelés. On devrait donc idéalement retrouver  $[1; 0.3; -0.1; 0.2]^t$

On observe ci-dessous la réponse de notre LMS au bruit blanc :

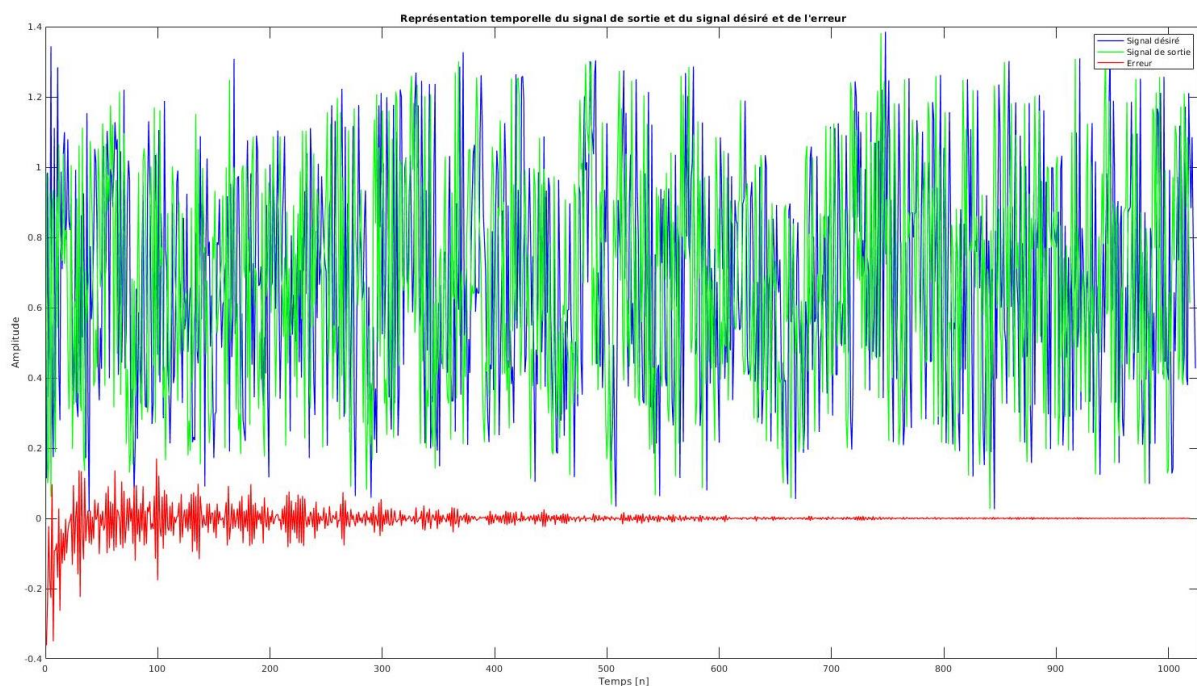


FIGURE 2 : Représentation temporelle du signal de sortie, du signal désiré et de l'erreur

On voit que le signal vert (sortie de notre LMS) qui est au départ assez différent de notre signal désiré (en bleu) finit par se superposer presque parfaitement avec ce dernier. On a vraiment l'impression à la fin que les deux signaux sont identiques mais simplement décalés d'un instant. Cela va de pair avec le fait que notre signal d'erreur tend vers 0 en oscillant beaucoup au départ puis en se stabilisant complètement au bout de quelques centaines de  $n$ .

Certes la convergence des signaux est une convergence « en moyenne » puisque l'erreur est une espérance, cependant on peut tout de même regarder si les différents coefficients  $w^i$  convergent vers les  $h^i$ . Centrer notre  $d$  a posé un problème pour la convergence de nos coefficients. Après avoir essayé plusieurs valeurs de  $\mu$  et de  $P$ , nos courbes ne convergeaient pas vers les valeurs de  $h$ . Enlever le centrage a résolu le problème.

## SA : ANNULATION D'ECHO ACOUSTIQUE

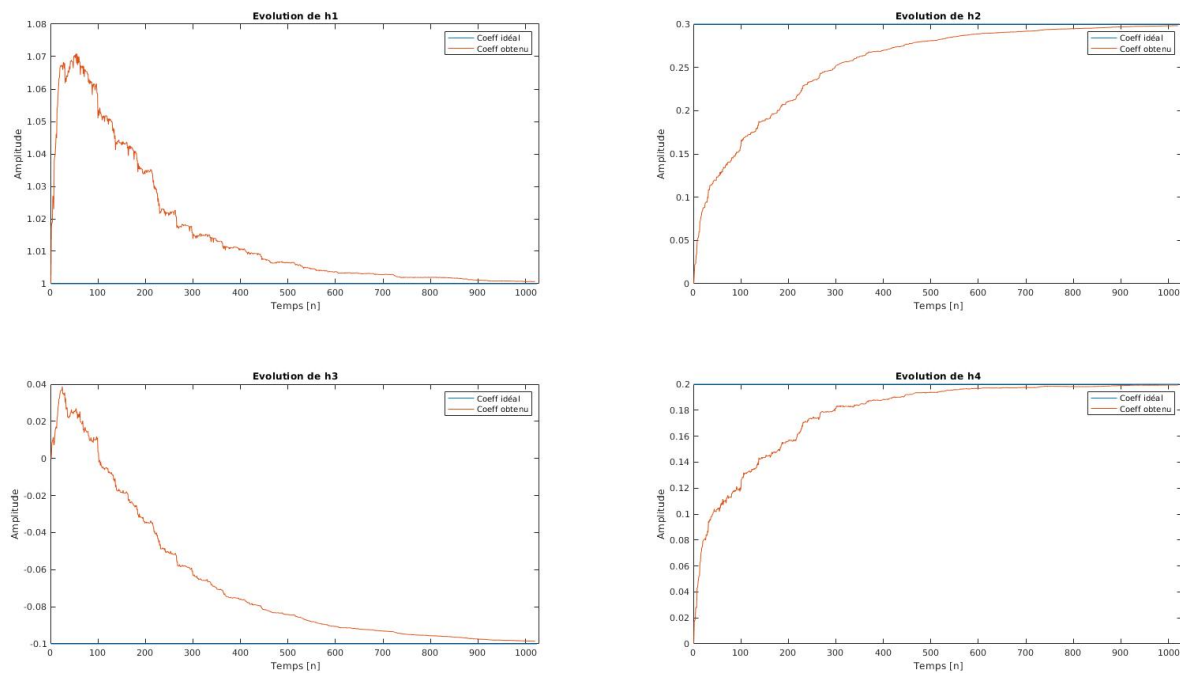


FIGURE 3 : Regard sur la convergence des coefficients de notre filtre estimé

On peut observer sur ces 4 graphes que les valeurs de poids de notre algorithme tendent à converger plus ou moins rapidement vers les valeurs « théoriques » annoncée en début de réponse. Cependant le fait que la convergence de  $w$  vers  $w^{opt}$  ne soit qu'en moyenne fait que notre convergence n'est pas uniforme (pas très lisse, avec plusieurs sauts...). On a également essayé de demander au LMS de prédire un filtre d'ordre plus important et pour tous les coefficients supplémentaires, l'algorithme fait converger les poids superflus vers 0 comme on peut l'observer sur le graphe suivant.

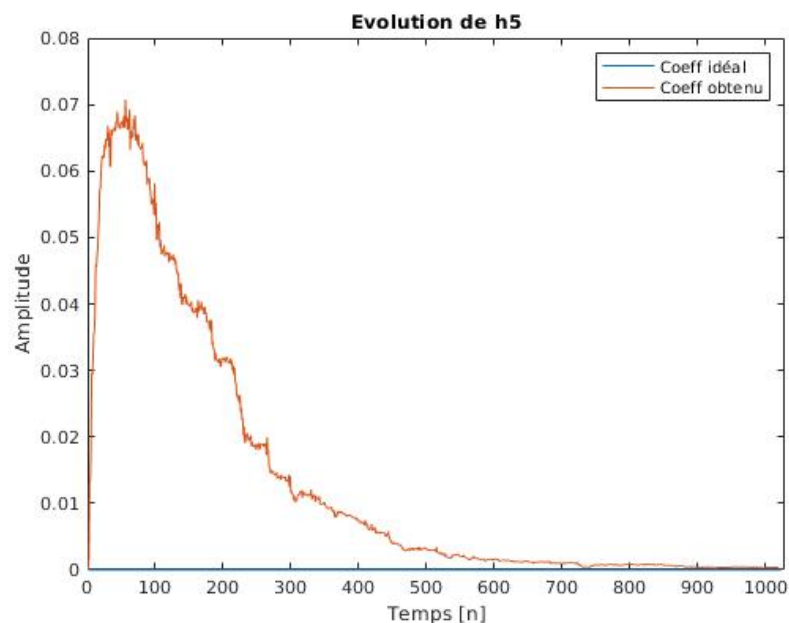


FIGURE 4 : Convergence de coefficients superflus

## Sous-partie 1.5 : Test de l'algorithme LMS

On initialisera notre vecteur  $W^{(0)}$  aléatoirement avec des valeurs assez faibles tout de même mais on fera attention à ne pas initialiser avec uniquement des zéros pour des questions d'efficacité (mettre uniquement un coefficient à 1 aurait pu faire l'affaire également). On utilise notre algorithme LMS et on cherche à observer l'effet de l'ordre du filtre ou du pas d'apprentissage sur sa sortie.

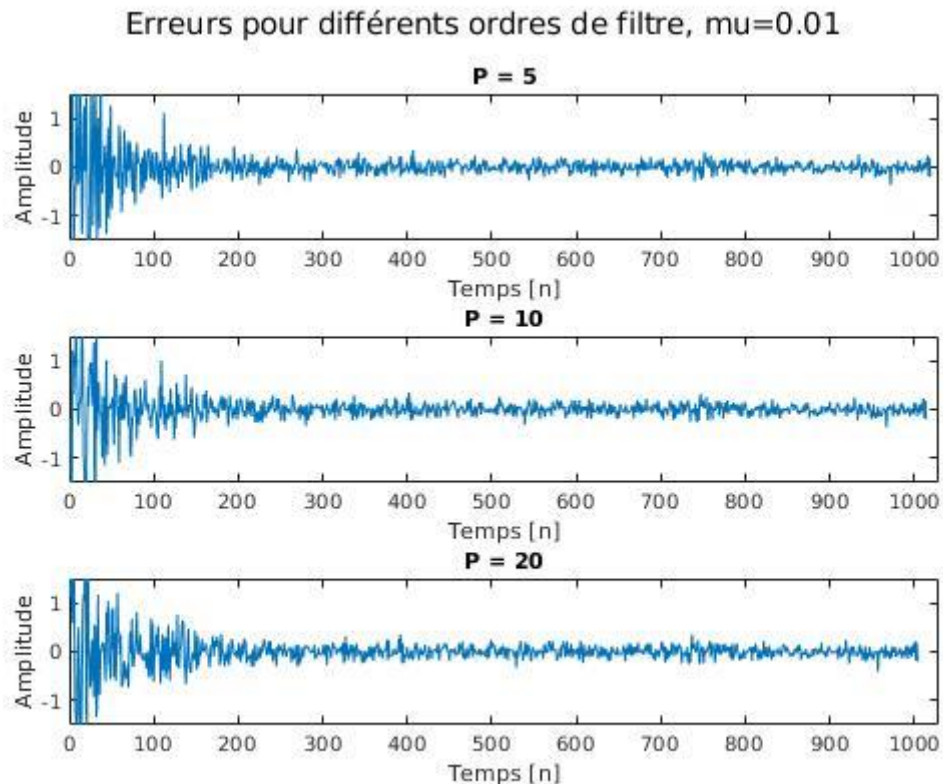


FIGURE 5 : Observation de l'effet de l'ordre  $P$  du filtre sur le signal d'erreur

Il aurait peut-être fallu faire les tests avec un pas d'apprentissage plus élevé car ici on n'observe aucun changement majeur lorsque  $P$  évolue. Cependant si on se réfère au cours on peut affirmer qu'en plus de jouer sur la complexité de l'algorithme, augmenter  $P$  pourrait conduire à la divergence de notre algorithme (on a pu en faire les frais dans la dernière partie en essayant de faire un filtre avec  $P = 1024$ ).



## SA : ANNULATION D'ECHO ACOUSTIQUE

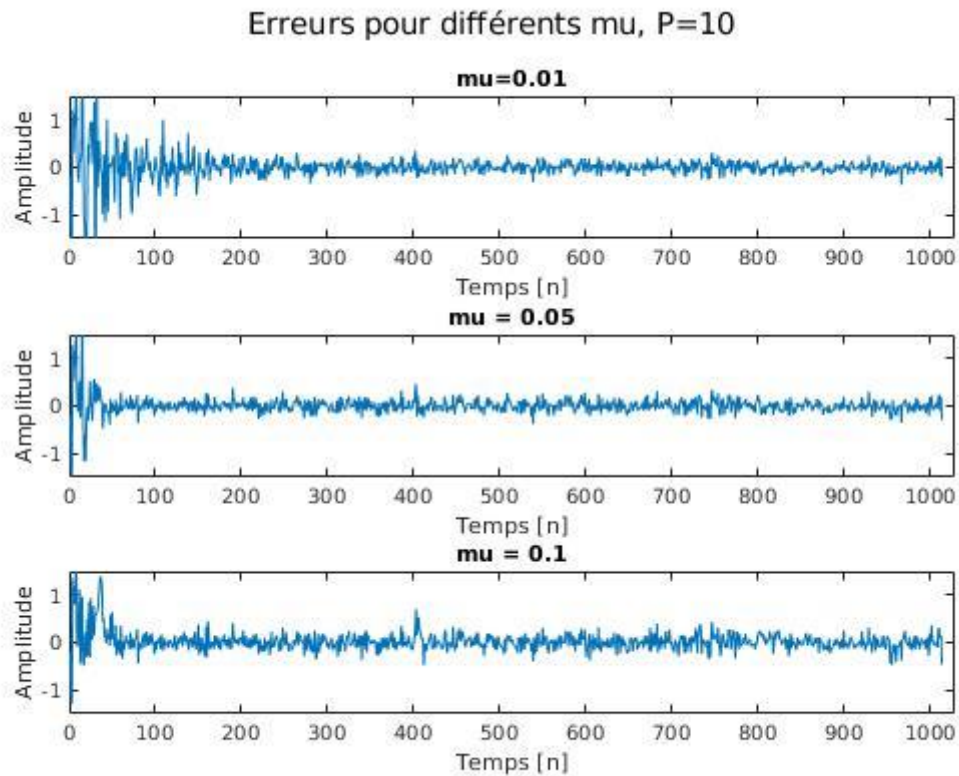


FIGURE 6 : Observation de l'effet du pas d'apprentissage sur le signal d'erreur

On remarque qu'augmenter  $\mu$  c'est accélérer la vitesse de convergence de notre algorithme. Cependant, un  $\mu$  trop élevé peut conduire à une divergence de l'algorithme. Ce n'est pas le cas ici car on traite le cas d'un bruit blanc et que pour voir un phénomène de divergence il faudrait peut-être aller vers  $\mu = 2$  pour que cela ait véritablement un effet (car les valeurs propres d'un bruit blanc sont relativement faibles ce qui laisse une grande liberté sur le choix de la valeur du pas d'apprentissage).

## Partie 2 : Application

### Sous-partie 2.1 : Signal audio avec une voix

On réalise dans cette partie l'annulation d'écho d'un signal voisé mélangé à un bruit blanc (qui recouvre presque totalement le signal voisé). On obtient les résultats ci-dessous après avoir appliqué l'algorithme LMS avec pour entrées :

- $x$  = écho
- $d$  = signal voisé sans écho ni bruit
- $P = 10$
- $\mu = 0.001$

Comparaison du signal d'entrée et le signal d'erreur renvoyé par le LMS

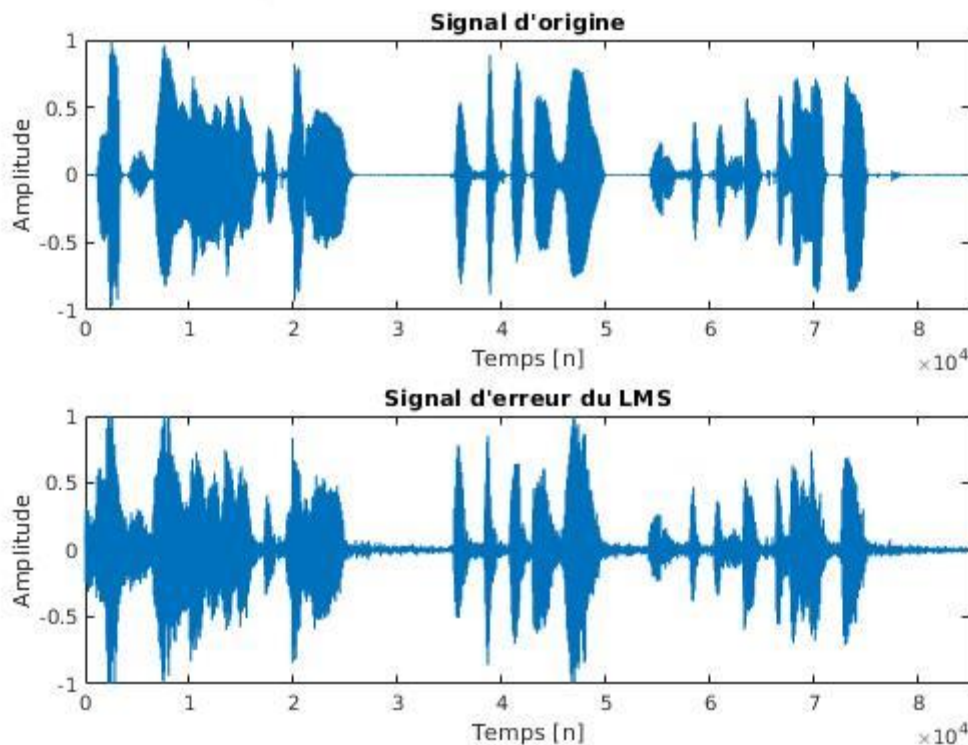


FIGURE 7 : Comparaison du signal d'entrée et le signal d'erreur renvoyé par le LMS

On remarque qu'au facteur de bruit rajouté près, le signal d'erreur renvoyé par notre LMS est exactement l'opposé du signal d'origine ce qui n'est pas vraiment problématique pour notre oreille qui arrive parfaitement à reconnaître le signal d'entrée dans le signal de sortie. C'est parfaitement logique étant donné la définition de notre erreur  $e_n = y_n - d_n$ . Pour parfaire notre travail on pourrait essayer de mieux approximer le filtre (augmenter  $P$ ) ou diminuer  $\mu$  pour finir d'atténuer le bruit blanc. Cependant tout cela allongerait grandement le temps de calcul...

## Sous-partie 2.2 : Signal audio avec deux voix

La situation simulée est la suivante :

- Steve parle dans son micro au point A.
- Ce signal acoustique est transmis au haut-parleur de Jim et l'environnement au point B du haut-parleur filtre ce signal par exemple en entraînant un effet d'écho.
- Le micro de Jim reçoit donc à la fois la réponse de Jim et le signal de Steve filtré par l'environnement au point B.

Dans cette situation, si aucun filtrage supplémentaire n'était appliqué avant restitution dans le haut-parleur de Steve, ce dernier s'entendrait parler. Pour éviter ce désagrément et le potentiel effet Larsen qui peut en découler, nous allons utiliser l'algorithme LMS de manière à estimer le filtre  $h$  de la pièce B avec un filtre  $w$ . Le signal que l'on voudrait retourner dans le haut-parleur étant  $d[n] - (h * x)[n]$  pour ne garder que la voix de Jim, on va effectivement y renvoyer  $d[n] - (w * x)[n] = -e[n]$ .

On va essayer plusieurs valeurs de  $P$  et  $\mu$  pour maximiser la qualité du signal de retour. Le résultat le plus probant a été trouvé pour  $P = 500$  et  $\mu = 0.01$ .

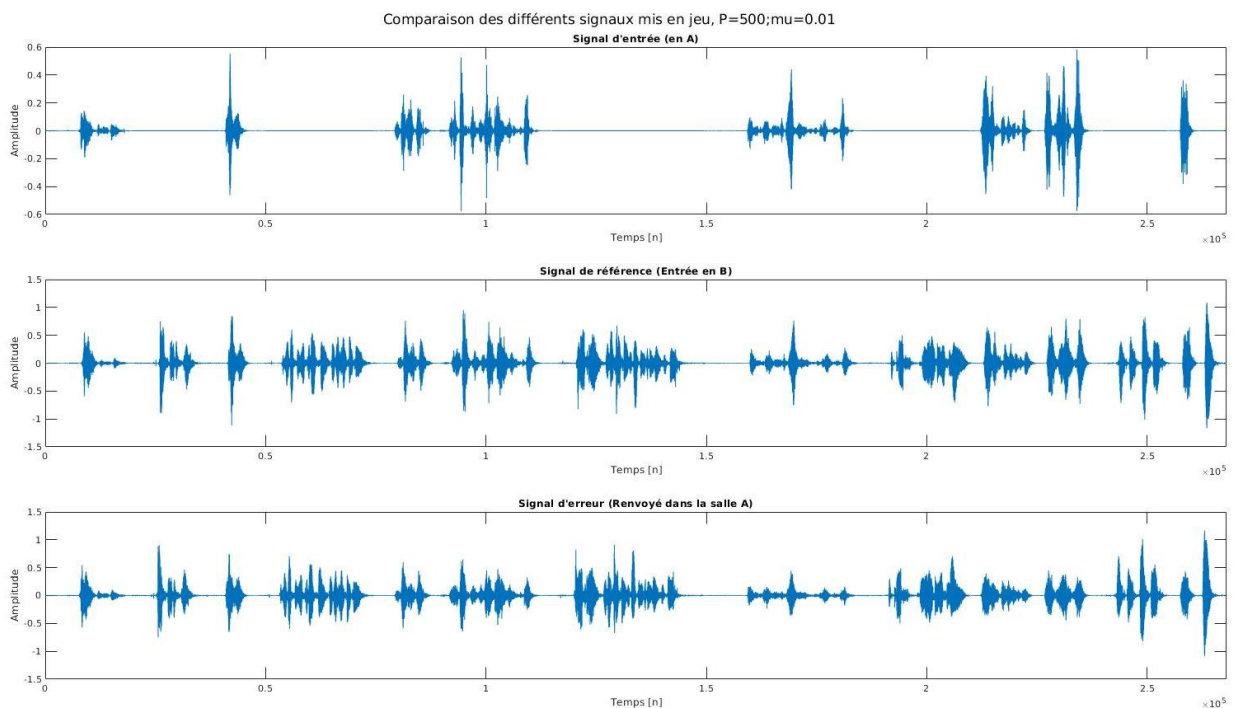


FIGURE 8 : Comparaison des signaux mis en jeu pour le couple d'hyperparamètres choisi

On peut voir que le résultat n'est pas parfait à tout instant (visible dans les différences d'amplitude). On pourrait sans doute encore diminuer  $\mu$  pour améliorer cela mais il faudrait peut-être un nombre trop important d'échantillons avant que notre système converge



SA : ANNULATION D'ECHO ACOUSTIQUE

correctement. C'est dans cette optique de compromis que l'on a décidé de s'arrêter sur ces choix d'hyperparamètres.