

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي  
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البلدية  
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا  
Faculté de Technologie

قسم الإلكترونيك  
Département d'Électronique



## Mémoire de Master

Filière Télécommunication

Spécialité Système de télécommunication

présenté par

AMEUR Loubna

&

FARDJALLAH Asmaa

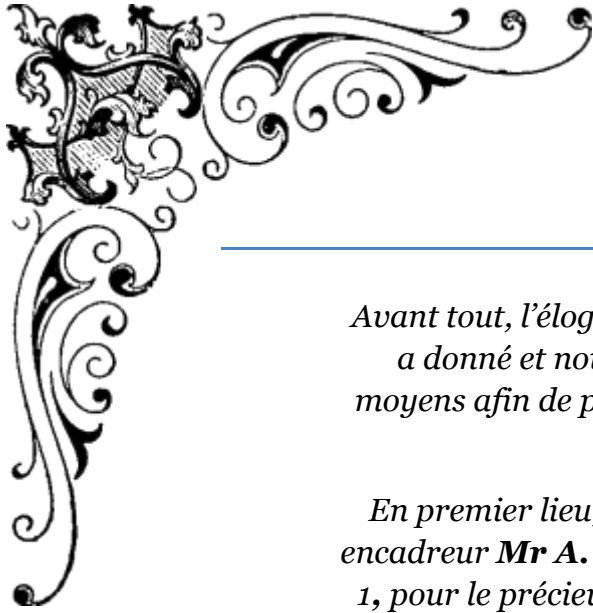
---

# Algorithmes VSS-NLMS avec l'estimation de la variance du bruit pour l'identification de réponses impulsionnelles acoustique.

---

Proposé par: MR BENALLAL Ahmed & GURUANI Imen

Année Universitaire 2020-2021



## Remerciements

---

*Avant tout, l'éloge à Dieu tout puissant pour tout ce qu'il nous a donné et nous avoir accordé la force, le courage et les moyens afin de pouvoir accomplir ce modeste travail « **Dieu merci** ».*

*En premier lieu, nous souhaitons vivement remercier notre encadreur **Mr A. BENALAL** professeur à l'université de Blida 1, pour la précieuse assistance, sa disponibilité et son soutien qu'il nous accordé tout au long de ce projet.*

*Un grand remerciement pour notre Co-promoteur **Mme GURUANI** pour le temps qu'elle a consacré et pour les précieuses informations qu'elle nous prodigués avec intérêt et compréhension.*

*Notre gratitude va aussi à nous **Enseignants de départements d'électronique** pour tout le savoir qu'ils ont su nous transmettre durant ces deux dernières années.*

*Nous remercions très vivement les membres du jury pour l'honneur et l'intérêt qu'ils nous ont accordé en acceptant d'examiner et d'évaluer ce mémoire.*

*Enfin un grand merci à nos parents, nos collègues et tous ceux qui ont contribué de près ou de loin à la réalisation du présent travail.*

## ***Dédicace***

*Avec un énorme plaisir, un cœur ouvert et une joie immense, je dédie ce modeste travail tout d'abord :*

***A mes très chers parents,*** *Je vous dois ce que je suis aujourd'hui grâce a votre amour, a votre patience et vos innombrables sacrifices.*

*Que dieu, le tout puissant, vous préserve et vous procure santé et longue vie*

***A mes chers frères,*** *pour leur aide leur encouragement et leur amour, puisse dieu donner une vie pleine de succès, de joie et de bonheur*

***A mon cher binôme,*** *AMEUR Loubna*

***A mes amis*** *qui m'ont toujours aidé et encouragé, qui étaient toujours à mes côtés, et qui mont accompagnaient durant mon chemin d'études supérieures.*

*Que dieux les protégés tous.*

**ASMAA**



## *Dédicaces*

***A mes parents ;** merci de me dire que le ciel est la limite et que je peux être ce que je veux être dans cette vie, et pour m'avoir transformé en cette personne que je suis aujourd'hui, je n'aurais jamais été là sans votre constante croire en moi et vouloir me voir grandir de plus en plus haut, réaliser tous mes rêves et me guider à travers tout cela. Je ne peux pas vous remercier assez, nous avons vécu tellement de choses ensemble et jamais vous ne m'avez abandonné, je ne pourrai jamais vous rembourser ce que vous avez fait pour moi, je vous aime sur la lune et à l'arrière*

***À mes deux familles, AMEUR et BOUGHADOU.** Vous me regardais devenir cette personne et tu étais fier de moi à travers tout cela, je suis impatient de vivre de nombreuses aventures avec vous, je t'aime et je suis fier de t'appeler ma famille.*

---

## ملخص:

الهدف من هذا المشروع هو تقديم حل لمشكلة وجود تباين الضوضاء في معادلات التحديث للخطوة المتغيرة لخوارزميات VSS-NLMS. في علم خوارزميات النغمة المتغيرة، استخدم المؤلفون تباين الضوضاء النظري المحسوب من SNR..

كلمات المفاتيح: إلغاء الصدى الصوتي، الترشيح التكيفي، حجم الخطوة المتغير، NLMS، الضوضاء المضافة، اختلاف الضوضاء

---

**Résumé :** Le but de ce projet est de fournir une solution au problème de la présence de la variance du bruit dans les équations de mise à jour du pas variable d'algorithmes VSS-NLMS. Dans la scientifique des algorithmes à pas variable, les auteurs utilisent une variance de bruit théorique calculée à partir du SNR.

**Mots clés :** Annulation d'écho acoustique, filtrage adaptatif, variable step size, NLMS, bruit additif, variation du bruit.

---

## Abstract:

The aim of this project is to provide a solution to the problem of the presence of noise variance in the updating equations of the variable pitch of VSS-NLMS algorithms. In the science of variable pitch algorithms, the authors use a theoretical noise variance calculated from the SNR.

**Keywords:** Acoustic echo cancellation, adaptive filtering, variable step size, NLMS, additive noise, noise variation.

---

## Listes des acronymes et abréviations

### ❖ Notation

[.] Opération de transposition.

[.]<sup>-1</sup> Opération de matrice inverse.

|.| Opération valeur d'absolue.

$E(.)$  Espérance mathématique.

$e(.)$  Opération d'exponentiel.

$\text{Log}_{10} (.)$  Logarithme à base 10.

$\text{trace} (.)$  Trace d'une matrice.

$\|.\|$  La norme 2.

$\nabla$  Vecteur de gradient.

$\partial (.)$  Dérivé partiel.

$y(n)$  : Sortie du filtre adaptatif.

$\mathbf{x}(n)$  : Signal d'entrée (signal de locuteur lointain).

$e(n)$  : Signal d'erreur.

$\mu$  : Pas d'adaptation.

$\mu(n)$  : Pas d'adaptation variable de NLMS

$J$  : Fonction de coût.

$\text{Trace}(R)$  : désigne la somme des éléments de la diagonale de la matrice d'autocréation.

$v(n)$  : Signal de locuteur proche.

$\hat{y}(n)$  : Signal d'écho estimé.

$b(n)$  : Signal de bruit ambiant.

$\mathbf{w}(n)$  : système inconnu.

$d(n)$  : signal désiré (écho bruité),

$N$  : Nombre d'itération totale

$L$  : Longueur du filtre adaptatif

$\delta$  : Paramètres de régularisation

$\lambda$  : facteur d'oubli, scalaire

$\sigma_x^2$  : L'énergie du signal d'entrée  $\mathbf{x}(n)$

$W$  : les coefficients du Filtre Adaptatif

$\beta$  : L'énergie de l'erreur

$T_s$  : période de convergence initiale.

## ❖ **Abbreviations**

**AEA** : **A**nnulation **d'****E**cho **A**coustique.

**EQM** : Erreur **q**uadratique **m**oyenne

**EQMM** : Erreur **Q**uadratique **M**oyenne **M**inimale.

**FIR** : **F**inite **I**mpulse **R**esponse (**R**éponse **I**mpulsionnelle **f**inie).

**LMS** : **L**east **M**ean **S**quare (Gradient Stochastique).

**MSE** : **M**ean **S**quare **E**rror.

**NLMS** : **N**ormalised **L**MS (Gradient Stochastique Normalisé).

**RI** : **R**éponse **I**mpulsionnelle.

**SNR** : **S**ignal to **N**oise **R**atio (Rapport Signal sur Bruit).

**USASI** : **U**SA **S**tandard **I**nstitute.

**VSS-NLMS**: **v**ariable **s**tep **s**ize **N**ormalized **L**MS.

## Table des matières

# Sommaire

### INTRODUCTION GENERALE

#### CHAPITRE 1 : ANNULATION D'ECHO PAR FILTRAGE ADAPTATIF

1.1 INTRODUCTION.....	3
1.2 L'ECHO ACOUSTIQUE.....	4
1.2.1 Les sources d'écho .....	4
1.2.2 Exemple de l'écho acoustique.....	5
1.2.3 Annulation d'écho acoustique.....	6
1.3 LE FILTRAGE ADAPTATIF.....	8
1.3.1 Principe de base du filtre adaptatif.....	8
1.3.2 Les applications du filtrage adaptatif.....	9
1.3.3 Filtre optimal de Wiener : .....	10
1.3.4 Critères de comparaison des algorithmes adaptatifs : .....	12
1.4 ALGORITHME ADAPTATIF.....	12
1.4.1 Choix de l'algorithme : .....	12
1.4.2 Algorithme du gradient stochastique LMS : .....	13
1.4.3 Algorithme du gradient stochastique normalisé NLMS : .....	15
1.5 CONCLUSION.....	16

#### CHAPITRE 2 : ALGORITHMES NLMS A PAS VARIABLES VSS ET LES 4 METHODES D'ESTIMATIONS DE LA VARIANCE DU BRUIT

2.1 INTRODUCTION.....	17
2.2 PRINCIPE d'UN VSS.....	17
2.3 ALGORITHMES VSS-NLMS POUR L'ANNULATION D'ECHO ACOUSTIQUE.....	18
2.3.1 Algorithme VSS-NLMS non paramétrique de BENESTY 2006 (VSS-NLMS NP).....	18
2.3.2 Nouvel Algorithme VSS-NLMS de IQBAL 2008 (NVSS-NLMS) .....	20
2.3.3 Algorithme VSS-NLMS 2020 (VSS-NLMS-2020) .....	22
2.4 LES METHODES D'ESTIMATION DE LA PUISSANCE DU BRUIT ADDITIF.....	24
2.5 CONCLUSION.....	34

#### CHAPITRE 3 : RESULTATS DE SIMULATIONS

3.1 INTRODUCTION.....	35
3.2 PRESENTATION DES SIGNAUX DE SIMULATIONS.....	35
3.3 DESCRIPTION DES CRITERES DE PERFORMANCE : .....	38
3.4 RESULTATS DE SIMULATIONS DES 3 ALGORITHMES VSS-NLMS AVEC LES 4 METHODES D'ESTIMATIONS DE LA VARIANCE DU BRUIT .....	38



3.4.1 Algorithme VSS-NLMS non paramétrique de BENESTY 2006 (VSS-NLMS NP) .....	39
3.4.2 Algorithme VSS-NLMS 2020 (VSS-NLMS-2020).....	50
3.5 COMPARAISON LES 4 METHODES D'ESTIMATIONS DE LA VARIANCE DU BRUIT POUR CHAQUE ALGORITHME VSS-NLMS.....	63
3.6 COMPARAISON DES 2 ALGORITHMES VSS-NLMS AVEC LES 4 METHODES D'ESTIMATIONS DE LA VARIANCE DU BRUIT .....	65
3.7 CONCLUSION.....	67
CONCLUSION GENERALE.....	68
BIBLIOGRAPHIES :.....	69

## Liste des figures

<b>Figure 1.1</b> : Génération d'un écho acoustique.....	4
<b>Figure 1.2</b> : Exemple de réponse impulsionnelle du canal acoustique.....	5
<b>Figure 1.3</b> : Composantes intervenant dans la transmission du signal.....	6
<b>Figure 1.4</b> : Principe de l'annulation d'écho acoustique.....	7
<b>Figure 1.5</b> : Principe d'un filtre adaptatif. ....	9
<b>Figure 1.6</b> : schéma principale du filtre Wiener.....	10
<b>Figure 1.7</b> : Schéma représentatif de l'algorithme LMS.....	13
<b>Figure 3.1</b> : Schéma de génération des signaux.....	35
<b>Figure 3. 2a</b> : Réponse impulsionnelle mesurée ( L=256) .....	36
<b>Figure 3.2b</b> : Réponse impulsionnelle mesurée (L=512). ....	36
<b>Figure 3.3</b> : Signal USASI.....	36
<b>Figure 3.4</b> : signal bbg1AR20. ....	37
<b>Figure 3.5</b> signal parole .....	37
<b>Figure 3. 6</b> : Variation linéaire du canal.....	38
<b>Figure 3.7</b> : signal USASI, cas stationnaire, SNR=20dB, L=256.....	38
<b>Figure 3.8</b> : signal USASI, cas non stationnaire, SNR=20dB, L=256.....	38
<b>Figure 3.9</b> : signal USASI, cas stationnaire, SNR=20dB, L=256.....	40
<b>Figure 3.10</b> : signal USASI, cas non stationnaire, SNR=20dB, L=256 .....	40
<b>Figure 3.11</b> signal USASI, cas stationnaire, SNR=15dB, L=256, $T_s=20*L$ .....	41
<b>Figure 3.12</b> : signal USASI, cas stationnaire, SNR=30dB, L=256, $T_s=30*L$ .....	41
<b>Figure 3.13</b> : signal USASI, cas non stationnaire SNR=15dB, L=256, $T_s=20*L$ .....	41
<b>Figure 3.14</b> : signaux USASI, cas non Stationnaire, SNR=30dB, L=256, $T_s=30*L$ .....	41
<b>Figure 3.15</b> : signal USASI, cas non Stationnaire, SNR=50dB, L=256, $T_s=50*L$ .....	42
<b>Figure 3.16</b> : signal parole, cas Stationnaire, SNR=20dB, L=256 .....	42
<b>Figure 3.17</b> : signal parole, cas Stationnaire SNR=20dB, L=512.....	42

<b>Figure 3.18</b> : signal USASI, cas stationnaire, SNR=15dB, L=256 Ts=20*L .....	43
<b>Figure 3.19</b> : signal USASI, cas stationnaire, SNR=30dB, L=256, Ts=20*L .....	43
<b>Figure 3.20</b> : signal USASI, cas non stationnaire, SNR=15dB, L=256, Ts=20*L.....	44
<b>Figure 3.21</b> : signal USASI, cas non stationnaire, SNR=30dB, L=256, Ts=30*L.....	44
<b>Figure 3.22</b> : signal parole, cas stationnaire, SNR=15dB, L=256 .....	44
<b>Figure 3.23</b> : signal parole, cas stationnaire, SNR=15, L=512.....	44
<b>Figure 3.24</b> : signal USASI, cas stationnaire, SNR=15dB, L=256, Ts=15*L .....	45
<b>Figure 3.25</b> : signal USASI, cas stationnaire, SNR=30dB, L=256, Ts=30*L.....	45
<b>Figure 3.26</b> : signal USASI, cas non stationnaire SNR=15dB, L=256, Ts=15*L.....	46
<b>Figure 3.27</b> : signal USASI, cas non stationnaire, SNR=30dB, L=256, Ts=30*L.....	46
<b>Figure 3.28</b> : signal BBG1AR20, cas non stationnaire SNR=15dB, L=256, Ts=30*L.....	46
<b>Figure 3.29</b> : signal parole, cas stationnaire, SNR=15dB, L=256.....	47
<b>Figure 3.30</b> : signal parole, cas non stationnaire SNR=15 L=256.....	47
<b>Figure 3.31</b> : signal USASI, cas stationnaire, SNR=20dB, L=512, Ts=20*L.....	48
<b>Figure 3.32</b> : signal USASI, cas stationnaire, SNR=30dB, L=512, Ts=30*L .....	48
<b>Figure 3.33</b> : signal BBG1AR20, cas stationnaire, SNR=20dB,L=256,Ts=30*L.....	48
<b>Figure 3.34</b> : signal USASI, cas non stationnaire SNR=20dB, L=256, Ts=20*L .....	49
<b>Figure 3.35</b> : signal USASI, cas non stationnaire, SNR=20dB, L=256, Ts=30*L .....	49
<b>Figure 3.36</b> : signal parole, cas stationnaire, SNR=20dB, L=256 .....	49
<b>Figure 3.37</b> : signal parole, cas non stationnaire, SNR=20, L=256.....	49
<b>Figure 3.38</b> : signal USASI, cas stationnaire, SNR=15dB, L=256, Ts=20*L.....	50
<b>Figure 3.39</b> : signal USASI, cas stationnaire, SNR=30dB, L=256,Ts=30*L.....	50
<b>Figure 3.40</b> : signal USASI, cas stationnaire, SNR=50dB, L=256, Ts=60*L.....	51
<b>Figure 3.41</b> : signal USASI, cas non stationnaire, SNR=15dB, L=256, Ts=20*L.....	51
<b>Figure 3.42</b> : signal USASI, cas non stationnaire, SNR=30dB, L=256, Ts=30*L.....	51
<b>Figure 3.43</b> : signal parole, cas stationnaire, SNR=15dB, L=256 .....	52
<b>Figure 3.44</b> : signal parole, cas stationnaire, SNR=15dB, L=512.....	52
<b>Figure 3.45</b> : signal USASI, cas stationnaire, SNR=15dB, L=256,Ts=20*L.....	52
<b>Figure 3.46</b> : signal USASI, cas stationnaire, SNR=30dB,L=256,Ts=30*L.....	52

<b>Figure 3.47</b> : signal USASI, cas stationnaire, SNR=50dB,L=256,Ts=50*L .....	53
<b>Figure 3.48</b> : signal BBG1AR20, cas stationnaire ,SNR=15dB,L=256,Ts=30*L.....	53
<b>Figure 3.49</b> :signal BBG1AR20, cas stationnaire ,SNR=30dB,L=256,Ts=110*L .....	53
<b>Figure 3.50</b> : signal BBG1AR20, cas non stationnaire, SNR=15dB,L=256,Ts=30*L.....	54
<b>Figure 3.51</b> : signal BBG1AR20 ,cas non stationnaire ,SNR=30dB,L=256,Ts=100*L.....	54
<b>Figure 3.52</b> : signal BBG1AR20, cas non stationnaire, SNR=50dB,L=256 ,Ts=110*L.....	54
<b>Figure 3.53</b> : signal parole, cas stationnaire, SNR=15dB,L=256.....	55
<b>Figure 3.54</b> : signal parole, cas stationnaire ,SNR=15dB,L=512.....	55
<b>Figure 3.55</b> : signal USASI , cas stationnaire ,SNR=15dB,L=256,Ts=20*L.....	56
<b>Figure 3.56</b> :signal USASI ,cas stationnaire ,SNR=30dB,L=256,Ts=25*L.....	56
<b>Figure 3.57</b> : signal USASI, cas stationnaire, SNR=50dB,L=256,Ts=50*L.....	56
<b>Figure 3.58</b> : signal BBG1AR20, cas stationnaire, SNR=15dB, L=256, Ts=50*L.....	56
<b>Figure 3.59</b> : signal BBG1AR20 , cas stationnaire ,SNR=30dB,L=256,Ts=110*L.....	56
<b>Figure 3.60</b> : signal USASI, cas non stationnaire, SNR=15dB, L=256 ,Ts=20*L .....	57
<b>Figure 3.61</b> : signal USASI, cas non stationnaire ,SNR=30dB,L=256,Ts=30*L.....	57
<b>Figure 3.62</b> : signal parole, cas stationnaire, SNR=15dB,L=256.....	57
<b>Figure 3.63</b> : signal parole, cas stationnaire ,SNR=15dB,L=512.....	57
<b>Figure 3.64</b> : signal USASI, cas stationnaire, SNR=15dB,L=256,Ts=15* L .....	58
<b>Figure 3.65</b> :signal USASI, cas stationnaire ,SNR=30dB,L=256,Ts=25*L.....	58
<b>Figure 3.66</b> : signal USASI, cas stationnaire, SNR=50dB,L=256,Ts=50*L.....	58
<b>Figure 3.67</b> :signal USASI, cas stationnaire ,SNR=15dB,L=512,Ts=15*L.....	58
<b>Figure 3.68</b> : signal USASI, cas non stationnaire, SNR=15dB,L=256 ,Ts=20*L .....	59
<b>Figure 3.69</b> : signal USASI, cas non stationnaire ,SNR=30dB,L=256,Ts=25*L.....	59
<b>Figure 3.70</b> : signal BBG1AR20, cas non stationnaire, SNR=15dB,L=256,Ts=30*L.....	59
<b>Figure 3.71</b> : signal parole, cas stationnaire, SNR=15dB,L=256.....	60
<b>Figure 3.72</b> : signal parole, cas stationnaire ,SNR=15dB,L=512.....	60
<b>Figure 3.73</b> : signal USASI, cas stationnaire, SNR=15dB, L=256.....	63
<b>Figure 3.74</b> : signal USASI, cas stationnaire, SNR=15dB, L=256.....	63
<b>Figure 3.75</b> : signal USASI, cas stationnaire, SNR=15dB, L=256.....	63
<b>Figure 3.76</b> : signal USASI, cas stationnaire, SNR=15dB, L=256.....	64
<b>Figure 3.77</b> : signal USASI, cas stationnaire, SNR=15dB, L=256.....	64

<b>Figure 3.78</b> : signal USASI, cas stationnaire, SNR=15dB, L=256.....	64
<b>Figure 3.79</b> : signal USASI, cas stationnaire, SNR=15dB, L=256.....	65
<b>Figure 3.80</b> : signal USASI, cas stationnaire, SNR=15dB, L=256.....	65
<b>Figure 3.81</b> : signal USASI, cas stationnaire, SNR=15dB, L=256.....	66
<b>Figure3.82</b> : signal USASI, cas stationnaire, SNR=15dB, L=256.....	66

.

## Liste des tableaux

<b>Tableau 1.1.</b> L'algorithme LMS. ....	14
<b>Tableau 1.2.</b> L'algorithme NLMS. ....	16
<b>Tableau 2.1.</b> NP-VSS-NLMS.....	20
<b>Tableau 2.2.</b> NVSS-NLMS-IQBAL avec méthode 0.....	21
<b>Tableau 2.3.</b> VSS-NLMS-2020 avec méthode 0.....	23
<b>Tableau 2.4.</b> NP-VSS-NLMS avec M1.....	26
<b>Tableau 2.5.</b> VSS-NLMS-2020 avec M1.....	27
<b>Tableau 2.6.</b> NP-VSS-NLMS-2020 avec M2.....	28
<b>Tableau 2.7.</b> VSS-NLMS-2020 avec M2.....	29
<b>Tableau 2.8.</b> NP-VSS-NLMS avec M3.....	30
<b>Tableau 2.9.</b> VSS-NLMS-2020 avec M3.....	31
<b>Tableau 2.10.</b> NP-VSS-NLMS avec M4.....	32
<b>Tableau 2.11.</b> VSS-NLMS-2020 avec M4.....	33
<b>Tableau 3.1.</b> Comparassions des 4 méthode d'estimation avec M0 pour NP-VSS-NLMS.....	61
<b>Tableau 3.2.</b> Comparassions des 4 méthode d'estimation avec M0 pour VSS-NLMS-2020 .....	62

# Introduction générale

---

La parole moyenne de communication privilégié entre les humains, constitue une grande partie des messages transmis en télécommunications un phénomène d'écho, qui est la réverbération du signal à l'émission, pose généralement un problème dans toutes les communications de type "PC a téléphoné " ou " Téléphone a Téléphone". Certains nouveaux services de télécommunications ont considéré le milieu acoustique (salle, cabine téléphonique, habitacle d'une voiture etc...) comme faisant partie de la chaîne de communication. Ces nouveaux services correspondant à l'apparition des postes à haut-parleur, des postes main-libres et des systèmes de téléconférence (audio et visioconférence). Toutefois, la qualité de la communication est souvent fortement dégradée par le phénomène d'écho acoustique.

L'annulation d'écho acoustique consiste à identifier le chemin d'écho entre le haut-parleur et le microphone, ceci permet une évaluation de l'écho qui est alors soustraite du signal à envoyer. Cette méthode apparaît bien adaptée à la résolution du problème d'écho puisqu'elle permet une communication bidirectionnelle simultanée. Le canal acoustique, composée essentiellement en première approximation de retards et d'atténuation, est généralement modélisé par un filtre de réponse impulsionnelle finie (RIF) de taille  $L$ . L'annulation d'écho acoustique est un problème d'identification d'un système linéaire (le canal acoustique) [1]. On doit donc utiliser des algorithmes d'identification adaptative qui soient robustes aux bruits perturbateurs en sortie. L'objectif dans ce Projet est faire un bilan des algorithmes est de fournir une solution au problème de la présence de la variance du bruit dans les équations de mise à jour du pas variable d'algorithmes VSS-NLMS. Ainsi que de nouvelles propositions pour estimation cette variance de manière continue

Dans le chapitre 1, traite le problème d'écho acoustique et les algorithmes utilisées pour l'annulation d'écho acoustique par le filtrage adaptatif en se basant sur le filtre de Wiener.

Le chapitre 2, concerne les algorithmes à pas variable VSS-NLMS pour l'annulation d'écho avec les méthodes d'estimation de la puissance de variance de bruit.

Notre intérêt est de performer la vitesse de convergence, le niveau de l'erreur final et la capacité de poursuite des variations du canal.

Dans le chapitre 3, nous présentons des résultats de simulations qui comparent les performances des algorithmes adaptatifs avec les méthodes d'estimation de bruit et présentés dans les chapitres précédents, ces performances sont basées sur la vitesse de convergence, l'énergie de l'erreur de filtrage final et la capacité de poursuite. Cette étude comparative est consacrée aux cas stationnaire et non stationnaire c'est-à-dire nous testons les algorithmes pour un canal acoustique stationnaire et non stationnaire. Nous terminons notre travail par une conclusion générale



# Chapitre 1 : Annulation d'écho par filtrage adaptatif

---

## 1.1 INTRODUCTION

Dans le système de communication mains-libres et audioconférences (téléconférence, conférence de bureau ...), on trouve le phénomène d'écho acoustique est provoqué à partir du couplage entre le haut-parleur et le microphone, ce phénomène cause une mauvaise qualité de communication, pour cela on a besoin des techniques afin de résoudre ce problème et d'améliorer la qualité de communication, le plus courant est le filtre adaptatif. Nous allons décrire les différentes classes d'algorithmes adaptatifs, nous intéressons aux algorithmes adaptatifs qui sont de bons candidats, comme le LMS et NLMS, ainsi que l'algorithme NLMS à pas variable (VSS-NLMS) représente une solution plus fiable qui assure un démarrage rapide et assure une erreur en régime permanent assez faible.

## 1.2 L'ECHO ACOUSTIQUE

L'écho acoustique est un phénomène dans lequel une version retardée, distordue d'un son est réfléchié et renvoyé vers la source. Ce problème entraînera une mauvaise qualité du signal vocal, par conséquent, les interlocuteurs ne peuvent pas entendre clairement le contenu de la conversation, même si les informations importantes ont été perdues. Cet écho acoustique est en effet le bruit qui est créé par la réflexion des ondes sonores par le mur et les autres choses existant dans le même système.

L'objectif principal pour les ingénieurs est l'annulation de cet écho acoustique, fournit un environnement sans écho pour les locuteurs pendant la conversation [1].

Le problème d'écho acoustique dérangera la conversation des personnes, réduira la qualité du système, c'est un problème commun des réseaux de communication [1].

L'origine d'écho acoustique provient de l'utilisation de nouveaux systèmes de télécommunication dits "mains libres". Le couplage acoustique provoque certains effets indésirables comme le phénomène de l'écho acoustique ou encore l'instabilité de la boucle de communication [3].

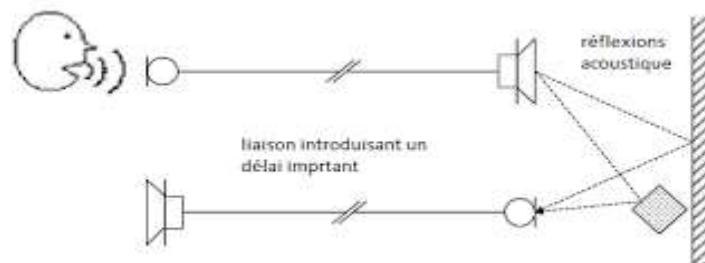
L'écho acoustique est provoqué par la transmission du signal émis par le haut-parleur et reçu par le microphone : cette transmission est composée d'un trajet direct et de

multiples réflexions captées par le microphone, et par conséquent seront renvoyer vers le locuteur qui a prononcé la parole dans une salle distante qu'est son propre signal, donc la propagation acoustique d'une onde sonore à l'intérieur d'un volume donné provoque l'écho acoustique.

### 1.2.1 Les sources d'écho

Les sources de génération de l'écho peuvent être d'origine électrique liées aux impédances du canal de transmission et liées au problème du gain rencontré pendant le processus de l'amplification du signal pour établir des communications lointaines.

Il existe deux sources typiques de l'écho acoustique : isolation et réflexion. L'écho d'isolation acoustique ou couplage acoustique est généré lorsque le haut-parleur et le microphone sont mal isolés l'un de l'autre. La deuxième forme d'écho acoustique est appelée écho acoustique ambient, ce type d'écho est généré lorsqu'une conversation téléphonique a lieu dans un environnement à réflexion acoustique. Dans ce cas, le microphone capte le premier flux de données audio d'origine, suivi par la parole qui est réfléchié par les murs et les parois. En télécommunications ; il est souvent nécessaire d'éliminer des échos gênants.

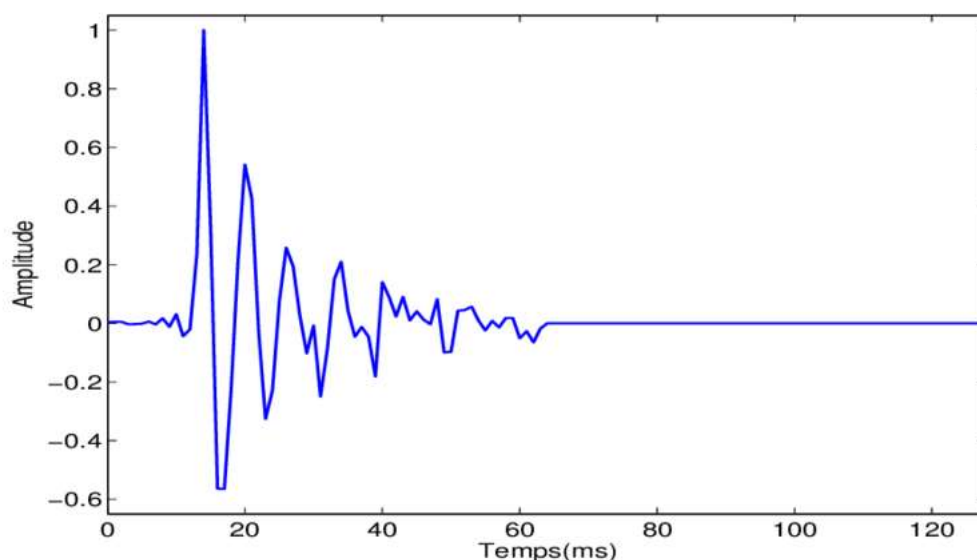


**Figure 1.1** : Génération d'un écho acoustique.[3]

## 1.2.2 Exemple de l'écho acoustique :

Les exemples les plus standards d'application de téléconférences sont par exemple les téléconférences (audio et vision), une salle spécialement conçue pour cette application est généralement utilisée. Lorsqu'un son est émis à l'intérieur d'une salle (ou d'une voiture). Il subit des transformations physiques qui peuvent être comprises grâce aux principes de l'acoustique des salles, des interprétations théoriques précisées peuvent être obtenues en faisant appel aux domaines de l'acoustique géométrique, ondulatoire et statistique. Néanmoins, le phénomène physique peut-être décrit simplement et successivement comme suit, une onde sonore, émise par un émetteur, se propage suivant les lois de l'acoustique vers un récepteur, au cours de son trajet, l'onde subit l'influence de l'environnement acoustique dans lequel elle se propage. Le phénomène se résume, dans le cas d'une propagation dans un espace libre, l'absorption d'une onde sonore par l'air dépend des paramètres (température, pression atmosphérique, etc...) variant lentement dans le temps par rapport à l'échelle de stationnarité du signal sonore. S'ajoutent à cela des phénomènes de réflexion, diffraction, diffusion, et absorption provoquée par les parois et obstacles présents dans l'espace clos. Le trajet de propagation d'une onde sonore est appelé canal acoustique, ce dernier dépend directement de ces différents paramètres.

La réponse impulsionnelle d'un canal acoustique se présente sous la forme d'une onde directe et d'une succession d'ondes réfléchies par les parois d'une salle particulière (figure 1.2).



**Figure 1.2 :** Exemple de réponse impulsionnelle du canal acoustique[2]

Les ondes se propagent en trajet direct (le trajet le plus court emprunté par l'onde sonore) jusqu'à ce qu'elles rencontrent un obstacle sur lequel elles se réfléchissent tout en perdant l'énergie. En réalité, la réponse impulsionnelle de couplage acoustique

est d'une durée infinie mais elle est généralement admise sur son support temporel significatif est de l'ordre de 50 à 100 ms dans une voiture, et de 250 ms à 300 ms dans une salle de téléconférence.

Le système binaural d'audition humaine gère remarquablement l'effet de réverbération (voir figure 1.3), lorsque les personnes en conversations sont dans la même salle même très réfléchissante.

Ce n'est pas le cas si les mêmes personnes sont en salles différentes et utilisent un haut-parleur pour la conversation. L'écho acoustique devient très gênant et inquiétant et donc doit être enlevé.

L'acoustique de la salle affectera toujours le son et peut entraver la communication.

En outre, un sifflement peut se produire si le microphone est placé trop près du haut-parleur (ce phénomène est connu par l'effet Larsen), il doit être éliminé par les annuleurs d'écho acoustique. La salle acoustique est une question complexe, mais elle a été largement étudiée en profondeur à partir de deux perspectives théoriques et pratiques.

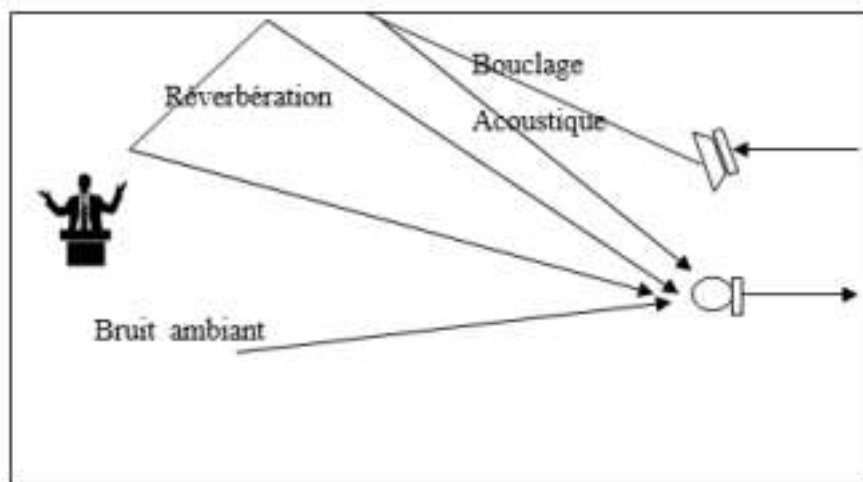


Figure 1.3 : Composantes intervenant dans la transmission du signal[2].

### 1.2.3 Annulation d'écho acoustique

L'annulation d'écho acoustique consiste à utiliser un filtre qui génère une estimée de l'écho réel, la différence entre les deux est l'écho résiduel (erreur d'estimation), le but est de minimiser l'erreur d'estimation pour réduire au mieux l'écho, alors que la technique d'annulation d'écho devient un problème de minimisation ceci nous permet d'utiliser des filtres optimums décrits dans la section suivante.

Le microphone (Room 2) dans la figure 1.4 capte à la fois la parole de locuteur proche (double parole) et la parole du locuteur distant (signal écho), à cause du couplage acoustique. Un système d'annulation d'écho estime l'écho acoustique entre l'entrée (signal envoyé dans le haut-parleur) et la sortie (signal récupéré par le microphone), puis il retranche cette estimation du signal local de parole. En effet, l'annulation de cet écho acoustique est basée sur l'estimation du canal acoustique entre le haut-parleur et le microphone par des techniques d'identification adaptative de la réponse impulsionnelle.

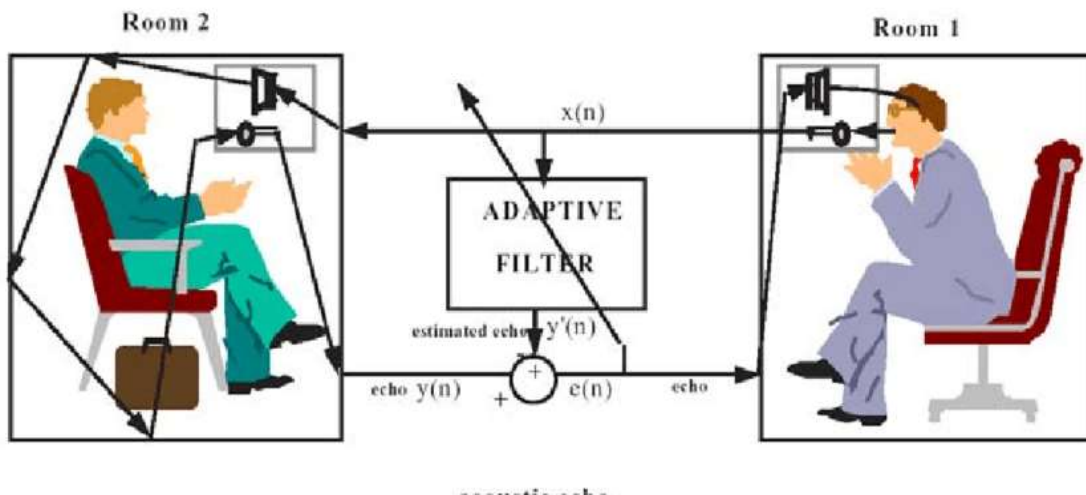


Figure 1.4 : Principe de l'annulation d'écho acoustique[2].

✓ **Filtre de réponse impulsionnelle finie (RIF) :**

Le filtre RIF ou FIR (en anglais) comprend une série de retards, de multiplicateurs et d'additionneurs ; à une entrée  $x(n)$ , et une sortie  $\hat{y}(n)$ . La sortie est exprimée sous la forme d'une combinaison linéaire d'échantillons d'entrée [1] :

$$\hat{y}(n) = \sum_{i=0}^{L-1} h_i(n)x(n - i) \quad (1.1)$$

Où  $h_i(n)$  sont les coefficients de filtre,  $L$  est la longueur du filtre ;  $\hat{y}(n)$  est donc la Convolution (produit intérieur) des deux vecteurs  $h(n)$  et  $x(n)$ . L'avantage significatif que les filtres FIR ont sur les autres structures sont importants que leurs fonctions de transfert ne contiennent que des zéros [1] :

$$H(z) = \sum_{k=0}^M b_k z^{-k} \quad (1.2)$$

✓ **La raison d'utiliser RIF que RII :**

On utilise un filtre FIR parce que les RI acoustique sont approximativement FIR

### 1.3 LE FILTRAGE ADAPTATIF

Un filtrage est rendu adaptatif si ses paramètres, les coefficients, sont modifiés selon un critère donné dès qu'une nouvelle valeur du signal devient disponible, ces modifications doivent suivre l'évolution des systèmes dans leur environnement aussi rapidement que possible. Le filtrage adaptatif est généralement associé avec un fonctionnement en temps réel (Dans le cas où les coefficients du filtre seraient variables dans le temps).

$x(n)$  : signal de l'entrée.

Avec :

$$x(n) = [x(n) \ x(n - 1) \dots x(n - L + 1)]^{-1} \quad (1.3)$$

$\hat{y}(n)$  : Le signal de sortie du filtre.

$d(n)$ : Écho bruité ou signal désiré.

Avec :

$$\begin{aligned} d(n) &= \mathbf{x}^T(n)\mathbf{h}(n) + v(n) \\ d(n) &= y(n) + v(n) \end{aligned} \quad (1.4)$$

$v(n)$ : signal de bruit additif de moyenne nulle.

$\mathbf{h}(n)$  : La réponse impulsionnelle de système (haut-parleur – microphone)

$$\mathbf{h}(n) = [h_0(n)h_1(n) \dots h_{L-1}(n)]^T \quad (1.5)$$

$\hat{\mathbf{h}}(n)$ : La réponse impulsionnelle de filtre adaptatif

Avec :

$$\hat{\mathbf{h}}(n) = [\hat{h}_0(n), \hat{h}_1(n) \dots \hat{h}_{l-1}(n)]^T \quad (1.6)$$

#### 1.3.1 Applications du filtrage adaptatif :

Le filtrage adaptatif est un outil puissant en traitement du signal, communications numériques, contrôle automatique. Les applications sont diverses mais présentent les caractéristiques suivantes : on dispose d'une entrée  $x(n)$ , ainsi que de la réponse désirée (référence)  $d(n)$  et l'erreur  $e(n)$ , qui est la différence entre  $d(n)$  et la sortie du filtre  $\hat{y}(n)$ , sert à contrôler (adapter) les valeurs des coefficients du filtre. Ce qui

différence essentiellement les applications provient de la façon de définir la réponse désirée  $d(n)$ .

On peut distinguer quatre grandes classes d'applications [4]:

- Identification de systèmes.
- La prédiction.
- La modelisation inverse.
- L'annulation d'interférences (réduction du bruit, annulation d'écho).

### 1.3.2 Principe de base du filtre adaptatif :

Le principe d'un filtre adaptatif est de rapprocher le signal de sortie du filtre (les coefficients variables) le mieux que possible d'un signal référence (désiré).

La technique de filtrage adaptatif se décompose en deux étapes (voir figure 1.5), une étape de filtrage qui permet d'obtenir une estimation du signal inconnu en convolant le signal d'entrée  $x(n)$  avec le coefficient du filtre adaptatif  $\hat{h}(n)$ .

L'erreur d'estimation  $e(n) = d(n) - \hat{y}(n)$  est ensuite utilisée dans la partie adaptation pour mettre jour les coefficients du filtre.

Une étape d'adaptation qui permet d'ajuster les coefficients du filtre adaptatif  $h$  suivant un algorithme donné [3].

A chaque itération les coefficients du filtre varient du signal d'erreur  $e(n)$  et ce pour faire diminuer la différence entre la sortie du filtre  $\hat{y}(n)$  et le signal désirée  $d(n)$ . Le signal d'erreur diminue jusqu'à atteindre dans certain cas une valeur nulle. À ce moment les coefficients du filtre adaptatif cessent de s'adapter.

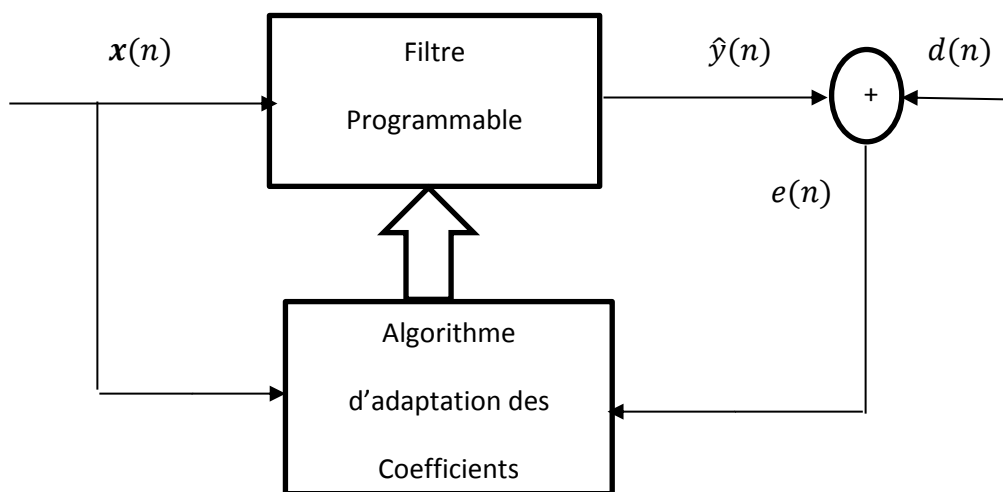


Figure 1.5: Principe d'un filtre adaptatif.

Où :

$x(n)$ : Le signal d'entrée.

$\hat{y}(n)$ : La sortie du filtre numérique.

$e(n)$ : Le signal d'erreur.

$d(n)$ : Le signal désiré (référence).

Les filtres adaptatifs peuvent être classés en fonction des choix qui sont faits sur les points suivants :

- le critère d'optimisation.
- l'algorithme de mise à jour des coefficients.
- la structure du filtre programmable.
- le type de signal traité, mono ou multidimensionnel.

Il existe deux classes importantes de filtres linéaires optimaux :

- filtrage de Wiener (ou les signaux considérés  $d(n)$  et  $x(n)$  sont stationnaires).
- filtrage de Kalman (qui est une généralisation du filtre de Wiener valable aussi dans le cas de processus (ou de signaux) non stationnaires).

### 1.3.3 Filtre optimal de Wiener :

Le filtrage de Wiener, qui doit son nom au scientifique Norbert Wiener, est adéquat pour les situations dans lesquelles le signal ou le bruit sont stationnaires (Voir figure 1.6)

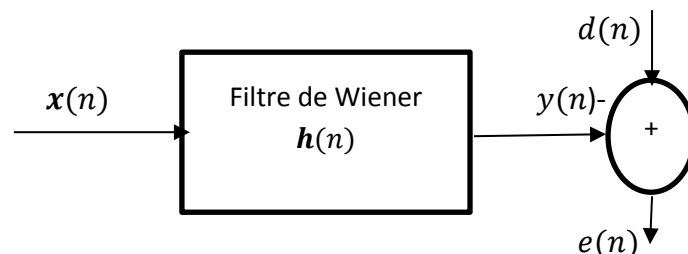


Figure 1.6 : schéma principale du filtre Wiener



Le principe du filtre de Wiener est de trouver en sortie une réponse  $\hat{y}(n)$  la plus proche possible d'une réponse désirée  $d(n)$ , lorsque l'entrée  $x(n)$  est perturbée par un bruit [3], comme le montre la figure 1.6.

La sortie du filtre  $\hat{y}(n)$  s'écrit :

$$\hat{y}(n) = \mathbf{h}^T \mathbf{x}(n) \quad (1.7)$$

Où  $\mathbf{h}$  est un vecteur des coefficients du filtre.

Le signal d'erreur est défini par (l'erreur entre la réponse désirée  $d(n)$  et la sortie de filtre  $\hat{y}(n)$ ) :

$$e(n) = d(n) - \hat{y}(n) \quad (1.8)$$

Les algorithmes adaptatifs permettent l'estimation du filtre adaptatif par le vecteur  $\mathbf{h}$ , de taille  $L$  à l'aide d'un critère basé sur l'erreur d'estimation à priori.

Cette erreur d'estimation, appelée précédemment signal de différence, est donnée par l'équation (1.6).

Il consiste alors à minimiser l'erreur quadratique moyenne entre le signal désiré et la sortie du filtre linéaire.

Erreur quadratique moyenne (EQM) :

$$j = E\{e^2(n)\} \quad (1.9)$$

La minimisation de  $J$  par rapport au vecteur  $\mathbf{h}$  donne :

$$E\{e_{min}(n) \mathbf{x}(n)\} = \mathbf{0}_{L \times 1} \quad (1.10)$$

C'est le principe d'orthogonalité On en déduit l'équation de Wiener-Hopf :

$$\mathbf{R} \mathbf{h}_{opt} = \mathbf{p} \quad (1.11)$$

Le vecteur optimum,  $\mathbf{h}_{opt}$  est celui qui annule le gradient du critère est donnée par :

$$\mathbf{h}_{opt} = \arg \min J(\mathbf{h}). \quad (1.12)$$

L'équation a une forme quadratique alors sa solution optimale est quand la dérivée de  $J_n(\mathbf{h})$  est nulle :

$$\nabla_{\mathbf{h}} J_n(\mathbf{h}) = \frac{\partial J_n}{\partial \mathbf{h}} = 0 \quad (1.13)$$

$$2\mathbf{R}_{L \times L} \mathbf{h}(n) - 2\mathbf{p}_{L \times 1} = 0 \quad (1.14)$$

Sachant que :

$$\mathbf{R}_{L \times L} = E\{\mathbf{x}^T(n)\mathbf{x}(n)\} \quad (1.15)$$

$\mathbf{R}_{L \times L}$ : est la matrice d'autocorrélation du vecteur  $\mathbf{x}(n)$

$\mathbf{p}_{L \times 1} = E\{d(n)\mathbf{x}(n)\}$  : est Le vecteur d'inter-corrélation du signal désiré et le signal d'entrée.

Si la matrice d'autocorrélation est inversible alors le filtre optimal est donné par l'expression suivante :

$$\mathbf{h}_{L,opt} = \mathbf{R}_{L \times L}^{-1} \mathbf{p}_{L \times 1} \quad (1.16)$$

Alors pour atteindre cet objectif, il faut connaître les propriétés statistiques  $\mathbf{R}_{L \times L}$  et  $\mathbf{p}_{L \times 1}$

## 1.4 Critères de comparaison des algorithmes adaptatifs

- Taux de convergence : nombre d'itérations pour converger suffisamment près de solution de Wiener.
- Désajustement : moyen d'ensembles de l'erreur quadratique final. (Erreur quadratique minimale obtenue avec Wiener).
- Robustesse : résistance au mauvais conditionnement des données.
- Complexité : nombre d'opérations par itération + place mémoire nécessaire (programme et donnée).
- Structure : aspect hardware, complexité de l'implantation matériel.
- Stabilité numérique : influence des erreurs de quadratique problème de la propagation des erreurs.
- 

## 1.5 ALGORITHME ADAPTATIF

Un algorithme est une spécification d'un schéma de calcul sous forme d'une suite finie d'opération élémentaire obéissant à un enchaînement déterminé qui permet de résoudre le problème.

### 1.5.1 Choix de l'algorithme :

Le choix de l'algorithme se fera en fonction des critères suivants [4] :

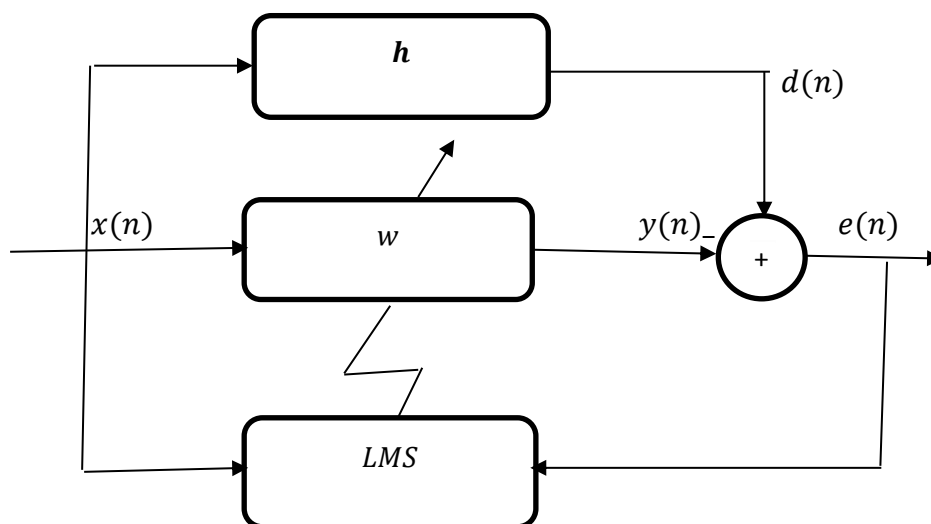
- La rapidité de convergence qui sera le nombre d'itérations nécessaires pour converger assez près de la solution optimale.
- La mesure de cette "proximité" entre cette solution optimale et la solution obtenue.
- La capacité de poursuite (tracking) des variations (non-stationnarités) du système.

- La robustesse au bruit.
- La complexité des calculs.
- La structure (modularité, parallélisme, ...).
- Les propriétés numériques (stabilité et précision) dans le cas d'une précision limitée sur les données et les coefficients du filtre.

### 1.5.2 Algorithme du gradient stochastique LMS

L'algorithme LMS (least-mean-square), conçu par Widrow et Hoff en 1959, l'un des premiers algorithmes adaptatifs les plus populaires pour le calcul des coefficients d'un filtre adaptatif à cause de sa robustesse et sa simplicité de calcul, c'est un algorithme qui minimise l'erreur quadratique moyenne entre un signal désiré et celui d'estimation, c'est-à-dire l'erreur. se base sur une estimation simple du gradient.

La figure (1.7) représente un schéma du principe de l'algorithme LMS :



**Figure 1.7** : Schéma représentatif de l'algorithme LMS.

Où :

$x(n)$  : signal d'entrée.

$d(n)$  : signal désiré (référence).

$h$  : système inconnu de longueur  $L$ .

$w$  : filtre d'estimation.

$y(n)$  : la sortie du filtre.

$e(n)$  : le signal d'erreur.

La sortie du filtre :

$$y(n) = \mathbf{w}^T(n)\mathbf{x}(n) \quad (1.17)$$

L'équation du signal d'erreur est donnée par :

$$e(n) = d(n) - y(n) \quad (1.18)$$

$$e(n) = d(n) - \mathbf{w}^T(n)\mathbf{x}(n) \quad (1.19)$$

L'équation de la mise à jour des filtres est donnée par :

$$\mathbf{h}(n) = \mathbf{h}(n-1) - \nabla_{\mathbf{h}}(e(n))e(n) \quad (1.20)$$

$\nabla_{\mathbf{h}}$ : Représente le gradient mais dans un sens du minimum

On obtient :

$$\mathbf{h}(n) = \mathbf{h}(n-1) + 2\mathbf{x}(n)e(n) \quad (1.21)$$

Cette relation est bien celle du LMS, on rajoute un paramètre  $\mu_0$  pour contrôler la vitesse de convergence, donc le LMS est géré par les deux équations :

1- Erreur du filtrage :

$$e(n) = d(n) - \mathbf{h}^T(n)\mathbf{x}(n) \quad (1.22)$$

2- Adaptation du filtrage :

$$\mathbf{h}(n) = \mathbf{h}(n-1) + \mu_0\mathbf{x}(n)e(n) \quad (1.23)$$

Avec :  $\mu_0$  est le pas d'adaptation de l'algorithme (qui dépend de l'énergie de  $\mathbf{x}(n)$ ).

La condition nécessaire et suffisante de convergence de l'algorithme (LMS) est :

$$0 < \mu_0 < \frac{2}{\lambda_{max}} \quad (1.24)$$

L'algorithme LMS est très simple : il nécessite seulement  $2L + 1$  multiplications et  $2L$  additions par itération, une étude plus précise (la moyenne quadratique), mais qui repose également sur des hypothèses contestables conduit à la condition plus contraignante :

$$0 < \mu_0 < \frac{2}{\text{Trace}(R)} = \frac{2}{L\sigma_x^2} \quad (1.25)$$

$L$  : est le nombre de coefficients du filtre adaptatif.

$\text{Trace}(R)$  : désigne la somme des éléments de la diagonale de la matrice d'autocorrélation du signal d'entrée  $R$  et  $\sigma_x^2$  désigne l'énergie du signal d'entrée  $\mathbf{x}(n)$ .

Cet algorithme n'est pas recommandé pour l'annulation d'écho en raison de la grande variation d'énergie contenue dans la voix.

Ont résumé l'algorithme LMS dans **la table 1.1** comme suite :

<p><b>Initialisation</b> : <math>\mathbf{w}_{L,0} = \mathbf{0}_L</math> ; <math>L</math> : la taille du filtre</p> <p><b>Conditions de convergence</b> : <math>0 &lt; \mu_0 &lt; \frac{2}{\text{Trace}(R)} = \frac{2}{L\sigma_x^2}</math></p> <p><b>Pour</b> <math>n = 1, 2, 3 \dots</math> (itérations)</p> <p><b>Signal d'entrée</b> : <math>\mathbf{x}(n) = [x(n) \ x(n-1) \ \dots \ x(n-L+1)]^T</math></p> <p><b>Erreur de filtrage</b> : <math>e(n) = d(n) - \mathbf{h}^T(n)\mathbf{x}(n)</math></p> <p><b>Equation de mise à jour</b> <math>\mathbf{h}(n) = \mathbf{h}(n-1) + \mu_0\mathbf{x}(n)e(n)</math></p> <p><b>Fin</b></p>
---

**Tableau 1.1.** L'algorithme LMS.

### 1.5.3 Algorithme du gradient stochastique normalisé NLMS

L'algorithme de gradient stochastique normalisé NLMS est une version modifiée de l'algorithme LMS, dont le gain d'adaptation est normalisé par l'énergie de signal d'entrée de  $\mathbf{x}(n)$ . Pour des signaux non stationnaire, l'énergie de signal  $\mathbf{x}(n)$  varie avec le temps, l'algorithme LMS aura du mal à fonctionner correctement puisque  $\mu_0$  constant, donc le nouveau pas d'adaptation est donné par :

$$\mu = \frac{\mu_0}{\mathbf{x}^T(n)\mathbf{x}(n)} = \frac{\mu_0}{\sum_{i=0}^{L-1} x^2(n-i)} = \frac{\mu_0}{L\sigma_x^2} \quad (1.26)$$

$\mu$  : C'est le pas d'adaptation de l'algorithme NLMS.

Pour la stabilité du NLMS il faut que :  $0 < \mu_0 < 2$

L'équation de la mise à jour des coefficients du filtre adaptatif par l'algorithme NLMS est donnée par :

$$\mathbf{h}(n) = \mathbf{h}(n-1) + \mu_0 \frac{\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{x}(n)} \quad (1.27)$$

Pour éviter les difficultés numériques (division par zéros), on introduit un scalaire de régularisation  $\delta$ .

$$\mathbf{h}(n) = \mathbf{h}(n - 1) + \mu_0 \frac{x(n)e(n)}{x^T(n)x(n) + \delta_{NLMS}} \quad (1.28)$$

L'algorithme LMS est très simple, mais reste inefficace en raison des grandes variations d'énergie. Ces variations d'énergie tendent à faire diverger les coefficients du filtre adaptatif. Pour cela, nous avons introduit le gradient normalisé NLMS.

Pour équilibrer le conflit entre les différents aspects du filtrage adaptatif tels que la vitesse de convergence et l'erreur quadratique moyenne en régime permanent de l'algorithme NLMS, un contrôle de pas d'adaptation est nécessaire.

On résume l'algorithme NLMS dans **la table 1.2** comme suite :

<p><b>Initialisation</b> : <math>\mathbf{h}_{L,0} = \mathbf{0}_L</math> ; L : la taille du filtre</p> <p><b>Conditions de convergence</b> : <math>0 &lt; \mu &lt; 2</math></p> <p><b>Erreur de filtrage</b> : <math>e(n) = d(n) - \mathbf{h}^T(n)\mathbf{x}(n)</math></p> <p><b>Equation de mise à jour</b> : <math>\mathbf{h}(n) = \mathbf{h}(n - 1) + \mu \frac{x(n)e(n)}{x^T(n)x(n) + \delta_{NLMS}}</math></p> <p><b>Fin</b></p>
--

**Tableau 1.2** . L'algorithme NLMS.

## 1.6 CONCLUSION

Dans ce chapitre, nous avons présenté la problématique de notre PFE, en commençant par des généralités sur le problème d'écho acoustique et l'annulation de ce dernier, en identifiant progressivement la Réponse Impulsionnelle Finie (FIR) à partir des algorithmes de type moindres carrés, par la suite nous avons expliqué le principe du filtrage adaptatif et le filtre Wiener qui répond au problème de minimisation du critère d'erreur choisi. Nous avons rappelé les algorithmes adaptatifs LMS, NLMS. Dans le chapitre suivant nous présentons deux familles d'algorithmes à pas d'adaptation variable et les méthodes d'estimations de la variance du bruit.

# Chapitre 2 : Algorithmes NLMS à pas variables VSS et les quatre méthodes d'estimation de variance du bruit

---

## 2.1 Introduction

Malgré l'introduction de la normalisation dans l'algorithme NLMS, ceci présente toujours des insuffisances causées principalement par le pas d'adaptation fixe (FSS NLMS Fixed Step Size NLMS). Plusieurs algorithmes de type NLMS à pas d'adaptation variable VSS-NLMS (Variable Step Size-NLMS) ont été proposés pour réduire l'erreur en régime permanent. Dans ce chapitre, nous présentons d'abord quelques algorithmes de type NLMS à pas variable existants VSS-NLMS. Par la suite nous représentons les méthodes d'estimation de la variance du bruit additif pour un fonctionnement autonome des algorithmes.

## 2.2 Le principe d'un algorithme NLMS à pas variable (VSS-NLMS)

L'algorithme NLMS est l'un des algorithmes adaptatifs les plus populaires. Dans cet algorithme, on trouve plusieurs critères (vitesse de convergence, capacité de poursuite, robustesse, la valeur finale de MSE...etc.), qui sont régies par un paramètre de pas d'adaptation fixe pour contrôler ses performances. Le compromis de NLMS doit être géré par le bon choix des valeurs de pas d'adaptation [5] : une grande valeur permet une vitesse de convergence rapide et une capacité de poursuite accrue, une valeur faible permet de minimiser l'erreur quadratique moyenne MSE en régime permanent et une robustesse au bruit.

Par contre les algorithmes VSS à pas variable permettent de résoudre le compromis entre la vitesse de convergence et l'erreur finale. On utilise une grande valeur de  $\mu$  dans les premières étapes de la convergence afin d'avoir une vitesse de convergence maximale, d'autre part, une petite valeur de  $\mu$  est utilisée après la convergence, qui correspond à une faible erreur en régime permanent [5].

## 2.3 Algorithme VSS-NLMS pour l'annulation d'écho acoustique

Plusieurs algorithmes de type VSS-NLMS ont été proposés dans la littérature [10]. Dans cette partie, nous présentons quelques algorithmes type NLMS à pas variable existants (VSS-NLMS).

### 2.3.1 L'algorithme non paramétrique (NP\_VSS\_NLMS) de BENESTY 2006

L'algorithme non paramétrique est l'une des solutions les plus intéressantes. Il a été développé dans un contexte d'identification de système [5], permet un contrôle robuste de l'algorithme NLMS pour donner des bonnes performances.

On considère le signal de sortie suivant :

$$d(n) = \mathbf{h}^T \mathbf{x}(n) + v(n) \quad (2.1)$$

Où :

$d(n)$ : Signal souhaité (écho bruité).

$v(n)$  : Signal de bruit.

$\mathbf{x}(n)$  : vecteur de signal d'entrée.

$n$  : est l'indice de temps

$T$  : L'exposant qu'il désigne la transposition

$$\mathbf{h} = [h_1 \ h_2 \ \dots \ h_{L-1}]^T \quad (2.2)$$

$\mathbf{h}$  est le système inconnu de longueur  $L$  que l'on va essayer d'identifier avec un filtre adaptatif  $\hat{\mathbf{h}}(n)$ .

Le signal distant  $\mathbf{x}(n)$  traverse la réponse impulsionnelle du trajet d'écho  $\mathbf{h}$ , ce qui donne le signal d'écho  $y(n)$ . Ce dernier est ajouté au bruit de fond  $v(n)$ , ce qui donne le signal souhaité  $d(n)$ .

La sortie du filtre adaptatif  $\hat{\mathbf{h}}(n)$ , fournit une réplique de l'écho, qui est en outre soustraite du signal souhaité.

$$\hat{y}(n) = \hat{\mathbf{h}}^T(n) \mathbf{x}(n) \quad (2.3)$$



L'équation de mise à jour de l'algorithme VSS-NLMS non paramétrique (NPVSS-NLMS) s'écrit alors :

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu_{NPVSS}(n)\mathbf{x}(n)e(n) \quad (2.4)$$

Où :

$\mu_{NPVSS}$  : est le pas d'adaptation variable, qui régit la stabilité, la vitesse de convergence et le mauvais réglage de l'algorithme.

Le pas variable est donné par la relation suivante :

$$\mu_{NPVSS}(n) = \begin{cases} \beta(n) & \text{si } \hat{\sigma}_e(n) \geq \sigma_v \\ 0 & \text{sinon} \end{cases} \quad (2.5)$$

Sachant que :

$$\beta(n) = [\delta + \mathbf{x}(n)\mathbf{x}^T(n)]^{-1} \left[ 1 - \frac{\sigma_v}{\epsilon + \hat{\sigma}_e(n)} \right] \quad (2.6)$$

Où,

$\sigma_v^2$  : est la puissance du bruit connue ou estimée

$\delta$  : est une faible constante de régularisation

La puissance de l'erreur de filtrage utilisé par l'algorithme dans (2.6) est donnée par la relation suivante :

$$\hat{\sigma}_e^2(n) = \lambda \hat{\sigma}_e^2(n-1) + (1-\lambda)e(n)^2 \quad (2.7)$$

$\lambda$  : est un facteur d'oubli défini par :

$$\lambda = 1 - \frac{k}{L} \quad k \geq 2 \quad (2.8)$$

Il est clair que  $\sigma_e(n) \geq \sigma_v$ , ce qui implique  $\mu_{NPVSS} \geq 0$ , en pratique, la quantité  $\hat{\sigma}_e^2(n)$  est estimée par la formule (2.7). Cette estimation pourrait entraîner une amplitude inférieure à l'estimation de la puissance du bruit  $\sigma_v^2$  ce qui rendrait  $\mu_{NPVSS}(n)$  négatif. Dans cette situation et afin de résoudre ce problème, les auteurs proposent de fixer  $\mu_{NPVSS}(n) = 0$  [6].

L'algorithme NPVSS-NLMS-2006 est donné ci-dessous dans la table 2.1 :

Etapes	Algorithme	Description
Paramètre :	$\hat{\mathbf{h}}(0) = 0$  $\lambda = 1 - \frac{1}{KL}$ Avec $K \geq 1$ $\mu_{\min} = 0.001$ $\mu_{\max} = \begin{cases} 0.8 & \text{if SNR} < 20 \\ 1 & \text{allieurs} \end{cases}$	$L$ : est la longueur du filter adaptatif  $n$ : le nombre d'itération qui correspond à l'index de temps.  $\delta, \varepsilon > 0$ : constants $\hat{\sigma}_e^2$ , Puissance estimée.  $\sigma_v^2$ , puissance de bruit connue
Erreur :	$e(n) = d(n) - \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n)$	$\lambda$ : facteur d'oubli
Mise-à-jour :	$\hat{\sigma}_e^2(n) = \lambda \hat{\sigma}_e^2(n-1) + (1 - \lambda) e^2(n)$  $\beta(n) = [\delta + \mathbf{x}^T(n)\mathbf{x}(n)]^{-1} [1 - \frac{\sigma_v}{\varepsilon + \hat{\sigma}_e(n)}]$  $\mu_{\text{NPVSS}}(n) = \begin{cases} \beta(n) & \text{if } \hat{\sigma}_e > \sigma_v \\ 0 & \text{allieurs} \end{cases}$  $\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu_{\text{NPVSS}}(n) \mathbf{x}(n) e(n)$	
	<b>Fin</b>	

**Tableau 2.1.** NP-VSS-NLMS

### 2.3.2 Nouvel algorithmes VSS\_ NLMS de Iqbal 2008 (NVSS-NLMS

**IQBAL) :**

Dans cette section, nous introduisons un deuxième algorithme VSS NLMS à pas variable.

Les équations de mise à jour sont données par [8] :

$$e(n) = d(n) - \mathbf{h}^T(n-1)\mathbf{x}(n) \quad (2.9)$$

$$\mathbf{h}(n) = \mathbf{h}(n-1) + \mu_{VSS}(n) \frac{\mathbf{x}(n)e(n)}{\|\mathbf{x}(n)\|^2 + C_0} \quad (2.10)$$

On définit le pas variable comme suit :

$$\mu_{VSS}(n) = \frac{\xi(n)}{\xi(n) + \gamma(n)} \quad (2.11)$$

Ou  $\xi(n)$  et  $\gamma(n)$  sont définie dans les deux équations suivantes :

$$\xi(n) = \left| \frac{r_{ed}(n) - \sigma_e^2(n)}{\sigma_d^2(n) - r_{ed}(n)} \right| \quad (2.12)$$

$$\gamma(n) = \sigma_e^2(n) - \frac{1}{\sigma_x^2(n)} r_{ex}(n)^T r_{ex}(n) \quad (2.13)$$

Les estimateurs récursifs des variables,  $r_{ed}(n)$ ,  $\sigma_e^2(n)$ ,  $\sigma_d^2(n)$ ,  $r_{ex}(n)$  et  $\sigma_x^2(n)$  sont définies dans l'algorithme Non paramétrique étendu de IQBAL.

Novel algorithme VSS-NLMS de Iqbal 2008 avec la méthode théorique M0 est donné ci-dessous dans la table 2.2 :

Etapes	Algorithme	Description
Paramètre :	$\hat{h}(0) = 0$ $\lambda = 1 - \frac{1}{KL}$ avec $K \geq 1$ $\mu_{min} = 0.001$ $\mu_{max} = 1$	$L$ : est la longueur du filtre adaptatif $n$ : le nombre d'itération qui correspond à l'index de temps $\delta, c_0 > 0$ : constants $\hat{\sigma}_e^2$ , Puissance estimée. $\sigma_v^2$ , puissance de bruit connue
Erreur :	$e(n) = d(n) - \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n)$	$\lambda$ : facteur d'oubli
Mise-à-jour :	$\hat{\sigma}_e^2(n) = \lambda \hat{\sigma}_e^2(n-1) + (1-\lambda)e^2(n)$ $\hat{\sigma}_x^2(n) = \lambda \hat{\sigma}_x^2(n-1) + (1-\lambda)\mathbf{x}(n)^2$ $\hat{\sigma}_m^2(n) = \lambda \hat{\sigma}_m^2(n-1) + (1-\lambda)\hat{y}^2(n)$ $r_{ex}(n) = \lambda r_{ex}(n-1) + (1-\lambda)\mathbf{x}(n)e(n)$ $r_{ed}(n) = \lambda r_{ed}(n-1) + (1-\lambda)e(n)\hat{y}(n)$ $\gamma(n) = \sigma_e^2(n) - \frac{r_{ex}(n)^T r_{ex}(n)}{\sigma_x^2(n)}$ $\xi(n) = \frac{ r_{ed}(n) - \sigma_e^2(n) }{ \sigma_d^2(n) - r_{ed}(n) }$ $\mu_{NVSS}(n) = \frac{\xi(n)}{\xi(n) + \gamma(n)}$ $\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu_{NVSS}(n) \frac{\mathbf{x}(n) e(n)}{ \mathbf{x}(n) ^2 + c_0}$	
<b>Fin</b>		

**Tableau 2.2.** NVSS-NLMS-IQBAL avec méthode 0

### 2.3.3 Algorithme VSS-NLMS 2020 :

Dans cette section, nous introduisons un autre algorithme VSS-NLMS de pas variable, qui est basé sur un critère simple de variation du pas d'adaptation donné par le rapport des énergies suivant [10] :

$$\mu_{VSS}(n) = \alpha \left| \frac{\sigma_d^2(n) - \sigma_{\hat{y}}^2(n)}{\sigma_v^2} \right| \quad (2.14)$$

Avec :

$\sigma_d^2(n)$ : L'énergie du signal désiré (écho bruité),  $d(n) = y(n) + v(n)$ .

$\sigma_{\hat{y}}^2(n)$ : L'énergie de l'écho estimé.

$\sigma_v^2$ : L'énergie du bruit additif estimé ou supposé connu. Par la suite, on suppose cette quantité calculée par la formule théorique du SNR.

$\alpha$ : Une constante comprise en 0 et 1 qui permet de régler le pas VSS. Elle est fixée ici 0.1.

Dans le cas stationnaire, en supposant  $y(n)$  et  $v(n)$  indépendants,  $\sigma_d^2(n)$  s'écrit :

$$\sigma_d^2(n) = \sigma_y^2(n) + \sigma_v^2 \quad (2.15)$$

En supposant que le filtre a suffisamment convergé ; c'est-à-dire  $\hat{y}(n) \approx y(n)$ , le pas variable (2.14) s'écrit :

$$\mu_{VSS}(n) \approx \alpha \left| \frac{\sigma_y^2(n) + \sigma_v^2 - \sigma_{\hat{y}}^2(n)}{\sigma_v^2} \right| = \alpha \ll 1 \quad (2.16)$$

Avec l'arrivée d'une non stationnarité, le signal écho sans bruit est perturbé et s'écrit :

$$y(n) + \Delta y(n) \quad (2.17)$$

Avec :

$\Delta y(n)$  : est une perturbation due à la non stationnarité

$y(n)$  : reste proche de  $\hat{y}(n)$ .

Le signal désiré  $d(n)$  s'écrit alors :

$$d(n) = y(n) + \Delta y(n) + v(n) \quad (2.18)$$

En supposant les trois composants de cette expression mutuellement indépendantes, l'énergie de  $d(n)$  s'écrit :

$$\sigma_d^2(n) = \sigma_y^2(n) + \sigma_{\Delta y}^2(n) + \sigma_v^2 \quad (2.19)$$

En utilisant (2.19) et avec l'hypothèse  $\sigma_y^2(n) \approx \hat{\sigma}_y^2(n)$ , le pas VSS s'écrit :

$$\begin{aligned} \mu_{VSS}(n) &= \alpha \left| \frac{\sigma_y^2(n) + \sigma_{\Delta y}^2(n) + \sigma_v^2 - \hat{\sigma}_y^2(n)}{\sigma_v^2} \right| \\ &= \alpha \left| 1 - \frac{\sigma_{\Delta y}^2(n)}{\sigma_v^2} \right| > \alpha \end{aligned} \quad (2.20)$$

Selon le rapport perturbation à bruit le pas variable VSS augmente et sera limité par une limite maximale  $\mu_{max} = 1$ .

Ainsi l'algorithme proposé s'écrit :

$$\mathbf{h}(n) = \mathbf{h}(n-1) + \frac{\mu_{VSS}(n)}{\|\mathbf{x}(n)\|^2 + c_0} \mathbf{x}(n)e(n) \quad (2.21)$$

Le pas  $\mu_{VSS}(n)$  s'écrit :

$$\mu_{VSS}(n) = \alpha \left| \frac{\sigma_d^2(n) - \hat{\sigma}_y^2(n)}{\sigma_v^2 + \delta} \right| \quad (2.22)$$

Où :

$\delta$  : une faible constante qui limite les valeurs de  $\sigma_v^2$  en cas de SNR très faibles. Les estimateurs récursifs de (2.22) sont :

$$\hat{\sigma}_y^2(n) = \lambda \hat{\sigma}_y^2(n-1) + (1-\lambda) \hat{y}^2(n) \quad (2.23)$$

$$\sigma_d^2(n) = \lambda \sigma_d^2(n-1) + (1-\lambda) d^2(n) \quad (2.24)$$

Le pas variable  $\mu_{VSS}(n)$  est limité dans l'intervalle  $[\mu_{min}, \mu_{max}]$

Avec :

$\lambda$  : est un facteur d'oubli

$c_0$  : une faible constante de régularisation.

L'algorithme VSS-NLMS 2020 avec la méthode théorique M0 est donné ci-dessous dans la table 2.3 :

Étapes	Algorithme	Description
Paramètre :	$\hat{\mathbf{h}}(0) = 0$ $\lambda = 1 - \frac{1}{KL}$ avec $K \geq 1$ $\mu_{min} = 0.001$	$L$ : est la longueur du filtre adaptatif $n$ : le nombre d'itération qui correspond à l'index de temps.

$$\mu_{max} = \begin{cases} 0.8 & \text{if } SNR < 20 \\ 1 & \text{allieurs} \end{cases} \quad \delta, c_0 > 0 : \text{constants}$$

$$\alpha = 0.1 \quad \sigma_v^2, \text{ puissance de bruit connue ou estimé}$$

Erreur :  $e(n) = d(n) - \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n)$   $\lambda$  : facteur d'oubli

Mise-à-jour :  $\hat{\sigma}_d^2(n) = \lambda \hat{\sigma}_d^2(n-1) + (1 - \lambda) d(n)^2$

$$\hat{\sigma}_y^2(n) = \lambda \hat{\sigma}_y^2(n-1) + (1 - \lambda) y^2(n)$$

$$\mu_{PROVSS}(n) = \alpha * \left| \frac{\hat{\sigma}_d^2(n) - \hat{\sigma}_y^2(n)}{\sigma_v^2 + \delta} \right|$$

$$\mu_{PROVSS}(n) = \begin{cases} \mu_{max} & \text{if } \mu_{PROVSS}(n) > \mu_{max} \\ \mu_{min} & \text{if } \mu_{PROVSS}(n) < \mu_{min} \end{cases}$$

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu_{PROVSS}(n) \frac{\mathbf{x}(n) e(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + c_0}$$

Fin

Tableau 2.3.VSS-NLMS-2020 avec méthode 0

## 2.4 Les méthodes d'estimation de la puissance du bruit Additif

Dans cette étude. Nous avons utilisé plusieurs méthodes d'estimation pour définir une meilleure estimation de l'algorithme non paramétrique et l'algorithme 2020 pour le paramètre  $\sigma_v^2$ :

### La Méthode0 (M0) :

Au début nous avons calculé par la méthode théorique est nommée la méthode 0 (M0), qui est donné par la suite :

$$SNR = 10 \log_{10} \frac{\sigma_y^2}{\sigma_v^2}$$

$$\sigma_v^2 = 10^{-\frac{SNR}{10}} \cdot \sigma_y^2 \quad (2.25)$$

Nous avons utilisé cette méthode pour la comparaison avec chaque méthode d'estimation pour conclure sur la méthode d'estimation la plus proche et la plus fiable à la méthode théorique.

## La Méthode1 (M1) :

La méthode M1 est l'une des solutions les plus pratiques, qui a été proposée par Jacob Benesty en 2009 [5]. On sait que le signal souhaité du filtre adaptatif est exprimé par :

$$d(n) = y(n) + v(n) \quad (2.26)$$

Où,

$y(n)$  : Signal d'écho.

$v(n)$  : Signal de locuteur proche.

Une solution plus simple a été proposée dans [5]. Le signal d'écho et le signal proche peut être considéré comme non corrélé. Ainsi :

$$E\{d^2(n)\} = E\{y^2(n)\} + E\{v^2(n)\} \quad (2.27)$$

En supposant que le filtre adaptatif a convergé vers un certain degré, on peut utiliser l'approximation :

$$E\{y^2(n)\} \approx E\{\hat{y}^2(n)\} \quad (2.28)$$

Par conséquent, en utilisant des estimations de puissance, nous obtenons :

$$\hat{\sigma}_v^2(n) \approx |\hat{\sigma}_d^2(n) - \hat{\sigma}_y^2(n)| \quad (2.29)$$

Avec :

$$\hat{\sigma}_y^2(n) = \lambda \hat{\sigma}_y^2(n-1) + (1-\lambda) \hat{y}^2(n) \quad (2.30)$$

$$\hat{\sigma}_d^2(n) = \lambda \hat{\sigma}_d^2(n-1) + (1-\lambda) d^2(n) \quad (2.31)$$

Où  $\lambda$  est un facteur d'oubli, utilisé dans l'estimation récursive de puissance dans (2.30) et (2.31) :

$$\lambda = 1 - \frac{1}{KL} \quad k \geq 1 \quad (2.32)$$

Les valeurs absolues dans (2.29) empêchent toute déviation mineure (due à l'utilisation d'estimations de puissance) par rapport aux valeurs vraies, ce qui peut rendre le pas normalisé négatif

Les deux algorithmes NP-VSS-NLMS/VSS-NLMS-2020 appliqué par la méthode 1 d'estimation de la puissance de bruit (M1) sont résumé par les tableaux au-dessous (2.4/2.5) :

Etapes	Algorithme	Description
Paramètre :	$\hat{\mathbf{h}}(0) = 0$	$L$ : est la longueur du filtre adaptatif
temps.	$\lambda = 1 - \frac{1}{KL}$ Avec $K \geq 1$	$n$ : le nombre d'itération qui correspond à l'index de
	$\mu_{\min} = 0.001$	$\delta, \varepsilon > 0$ : constants
	$\mu_{\max} = \begin{cases} 0.8 & \text{if SNR} < 20 \\ 1 & \text{allieurs} \end{cases}$	$\hat{\sigma}_e^2$ , Puissance de bruit estimée
	$Ts = m * L$ , temps de démarrage	$\hat{\sigma}_d^2$ , puissance de signal désiré estimée
		$\hat{\sigma}_y^2$ , Puissance de la sortie du filtre estimée.
		$\lambda$ : Facteur d'oubli
Erreur :	$e(n) = d(n) - \mathbf{h}^T(n-1)\mathbf{x}(n)$	
Méthode d'estimation 1 :		
	$\hat{\sigma}_d^2(n) = \lambda' \hat{\sigma}_d^2(n-1) + (1 - \lambda') d(n)^2$	
	$\hat{\sigma}_y^2(n) = \lambda' \hat{\sigma}_y^2(n-1) + (1 - \lambda') \hat{y}^2(n)$	
	$\hat{\sigma}_v^2(n) =  \hat{\sigma}_d^2(n) - \hat{\sigma}_y^2(n) $	
Mise-à-jour :	$\hat{\sigma}_e^2(n) = \lambda \hat{\sigma}_e^2(n-1) + (1 - \lambda) e^2(n)$	
	$\beta(n) = [\delta + \mathbf{x}^T(n)\mathbf{x}(n)]^{-1} [1 - \frac{\sigma_v}{\varepsilon + \hat{\sigma}_e(n)}]$	
	$\mu_{npvss}(n) =  \beta(n) $	
	$\mu_{npvss}(n) = \begin{cases} \frac{\mu_{max}}{\delta + \mathbf{x}^T(n)\mathbf{x}(n)} & \text{if } n < Ts \\  \beta(n)  & \text{if } n > Ts \end{cases}$	
	$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu_{NPVSS}(n) \mathbf{x}(n) e(n)$	
	<b>Fin</b>	

**Tableau 2.4.** NP-VSS-NLMS avec M1



Etapes	Algorithme	Description
Paramètre : adaptatif	$\hat{h}(0) = 0$	$L$ : est la longueur du filtre
d'itération qui correspond à l'index de temps.	$\lambda = 1 - \frac{1}{KL}$ avec $K \geq 1$	$n$ : le nombre
	$\mu_{max} = \begin{cases} 0.8 & \text{if } SNR < 20 \\ 1 & \text{allieurs} \end{cases}$	$\mu_{min} = 0.001$
	$\alpha = 0.1$	$\delta, c_0 > 0$ : constants
	$Ts = m * L, m = 20,30,40$ estimée	$\sigma_v^2$ , puissance de bruit estimé
		$\hat{\sigma}_{d1}^2$ , puissance de signal désiré
		$\hat{\sigma}_{y1}^2$ , Puissance de la sortie du filtre estimée.
Erreur :	$e(n) = d(n) - \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n)$	$\lambda$ : facteur d'oubli
Méthode d'estimation 1 :		
	$\hat{\sigma}_{d1}^2(n) = \lambda' \hat{\sigma}_{d1}^2(n-1) + (1 - \lambda') d(n)^2$	
	$\hat{\sigma}_{y1}^2(n) = \lambda' \hat{\sigma}_{y1}^2(n-1) + (1 - \lambda') \hat{y}^2(n)$	
	$\hat{\sigma}_{v1}^2(n) =  \hat{\sigma}_{d1}^2(n) - \hat{\sigma}_{y1}^2(n) $	
Mise-à-jour :	$\hat{\sigma}_d^2(n) = \lambda \hat{\sigma}_d^2(n-1) + (1 - \lambda) d(n)^2$	
	$\hat{\sigma}_y^2(n) = \lambda \hat{\sigma}_y^2(n-1) + (1 - \lambda) \hat{y}^2(n)$	
	$\mu_{PROVSS}(n) = \alpha * \left  \frac{\hat{\sigma}_d^2(n) - \hat{\sigma}_y^2(n)}{\sigma_v^2 + \delta} \right $	
	$\mu_{PROVSS}(n) = \begin{cases} \mu_{max} & \text{if } \mu_{PROVSS}(n) > \mu_{max} \\ \mu_{min} & \text{if } \mu_{PROVSS}(n) < \mu_{min} \end{cases}$	
	$\mu_{PROVSS}(n) = \mu_{max}$ if $n < Ts$	
	$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu_{PROVSS}(n) \frac{\mathbf{x}(n) e(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + c_0}$	
	<b>Fin</b>	

**Tableau 2.5.** VSS-NLMS-2020 avec M1

## La Méthode2 (M2) :

La deuxième méthode a été proposée par Jacob Benesty et Silviu Ciochin en 2010[6], est en ce qui concerne l'estimation du paramètre  $\sigma_e(n)$ .

La puissance du signal d'erreur est estimée comme suit :

$$\hat{\sigma}_e^2(n) = \lambda \hat{\sigma}_e^2(n-1) + (1-\lambda)e^2(n) \quad (2.33)$$

Les deux algorithmes NP-VSS-NLMS/VSS-NLMS-2020 appliqué par la méthode 2 d'estimation de la puissance de bruit (M2) sont résumés par les tableaux au-dessous (2.6/2.7) :

Etapes	Algorithme	Description
Paramètre :	$\hat{\mathbf{h}}(0) = 0$	$L$ : est la longueur du filtre adaptatif
temps.	$\lambda = 1 - \frac{1}{KL}$ Avec $K \geq 1$	$n$ : le nombre d'itération qui correspond à l'index de
	$\mu_{\min} = 0.001$	
	$\mu_{\max} = \begin{cases} 0.8 & \text{if SNR} < 20 \\ 1 & \text{allieurs} \end{cases}$	$\delta, \varepsilon > 0$ : constants
	$Ts = m * L, gain(Nstart)$ estimée	$\hat{\sigma}_e^2, \hat{\sigma}_{ee}^2$ , Puissance de bruit
		$\lambda$ : Facteur d'oubli
Erreur :	$e(n) = d(n) - \mathbf{h}^T(n-1)\mathbf{x}(n)$	
Méthode d'estimation 2 :		
	$\hat{\sigma}_{ee}^2(n) = \lambda' \hat{\sigma}_{ee}^2(n-1) + (1-\lambda') e(n)^2$	
	$\hat{\sigma}_v^2(n) = \hat{\sigma}_{ee}^2(n)$	
Mise-à-jour :	$\hat{\sigma}_e^2(n) = \lambda \hat{\sigma}_e^2(n-1) + (1-\lambda) e^2(n)$	
	$\beta(n) = [\delta + \mathbf{x}^T(n)\mathbf{x}(n)]^{-1} [1 - \frac{\sigma_v}{\varepsilon + \hat{\sigma}_e(n)}]$	
	$\mu_{npvss}(n) =  \beta(n) $	
	$\mu_{npvss}(n) = \begin{cases} \frac{\mu_{\max}}{\delta + \mathbf{x}^T(n)\mathbf{x}(n)} & \text{if } n < Ts \\  \beta(n)  & \text{if } n > Ts \end{cases}$	
	$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu_{NPVSS}(n) \mathbf{x}(n) e(n)$	
	<b>Fin</b>	

**Tableau 2.6.** NP-VSS-NLMS-2020 avec M2

Etapes	Algorithme	Description
Paramètre :	$\hat{h}(0) = 0$	$L$ : est la longueur du filtre adaptatif
temps.	$\lambda = 1 - \frac{1}{KL}$ avec $K \geq 1$	$n$ : le nombre d'itération qui correspond à l'index de
	$\mu_{min} = 0.001$	
	$\mu_{max} = \begin{cases} 0.8 & \text{if } SNR < 20 \\ 1 & \text{allieurs} \end{cases}$	$\delta, c_0 > 0$ : constants
	$\alpha = 0.1$	$\sigma_v^2$ , puissance de bruit estimé
	$Ts = m * L, m = 20,30,40$	$\lambda$ : facteur d'oubli
Erreur :	$e(n) = d(n) - \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n)$	
Méthode d'estimation 2 :		
	$\hat{\sigma}_{ee}^2(n) = \lambda' \hat{\sigma}_{ee}^2(n-1) + (1 - \lambda') e(n)^2$	
	$\hat{\sigma}_v^2(n) = \hat{\sigma}_{ee}^2(n)$	
Mise-à-jour :	$\hat{\sigma}_d^2(n) = \lambda \hat{\sigma}_d^2(n-1) + (1 - \lambda) d(n)^2$	
	$\hat{\sigma}_y^2(n) = \lambda \hat{\sigma}_y^2(n-1) + (1 - \lambda) y^2(n)$	
	$\mu_{PROVSS}(n) = \alpha * \left  \frac{\hat{\sigma}_d^2(n) - \hat{\sigma}_y^2(n)}{\sigma_v^2 + \delta} \right $	
	$\mu_{PROVSS}(n) = \begin{cases} \mu_{max} & \text{if } \mu_{PROVSS}(n) > \mu_{max} \\ \mu_{min} & \text{if } \mu_{PROVSS}(n) < \mu_{min} \end{cases}$	
	$\mu_{PROVSS}(n) = \mu_{max} \text{ if } n < Ts$	
	$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu_{PROVSS}(n) \frac{\mathbf{x}(n) e(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + c_0}$	
	<b>Fin</b>	

**Tableau 2.7.** VSS-NLMS-2020 avec M2

### La Méthode 3 (M3) :

Cette méthode a été proposée par Mohammad Asif Iqbal en 2008 [8], c'est la méthode d'inter-corrélation qui peut être évaluée comme suit :

L'estimation de l'énergie du signal proche est donnée par la relation suivante [9] :

$$\hat{\sigma}_v^2(n) = \hat{\sigma}_e^2(n) - \frac{\hat{\mathbf{r}}_{\mathbf{x}\mathbf{e}}^T(n)\hat{\mathbf{r}}_{\mathbf{x}\mathbf{e}}(n)}{\hat{\sigma}_x^2(n) + C_0} \quad (2.34)$$

Avec :

$$\hat{\sigma}_x^2(n) = \lambda \sigma_x^2(n-1) + (1-\lambda) \mathbf{x}^2(n) \quad (2.35)$$

$$\hat{\mathbf{r}}_{xe}(n) = \lambda \hat{\mathbf{r}}_{xe}(n-1) + (1-\lambda) \mathbf{x}(n)e(n) \quad (2.36)$$

$\hat{r}_{xe}^T(n)$  : Désigne l'inter-corrélation entre  $\mathbf{x}(n)$  et  $e(n)$ .

$e(n)$  : Signal d'erreur.

$\hat{\sigma}_x^2(n)$  : est la puissance du signal d'entrée.

Les deux algorithmes NP-VSS-NLMS/VSS-NLMS-2020 appliqué par la méthode 3 d'estimation de la puissance de bruit (M3) sont résumés par les tableaux au-dessous (2.8/2.9) :

Étapes	Algorithme	Description
Paramètre :	$\hat{\mathbf{h}}(0) = 0$  $\lambda = 1 - \frac{1}{KL}$ Avec $K \geq 1$ $\mu_{\min} = 0.001$ $\mu_{\max} = \begin{cases} 0.8 & \text{if SNR} < 20 \\ 1 & \text{allieurs} \end{cases}$  $Ts = m * L, \text{ gain } (Nstart)$	$L$ : est la longueur du filtre adaptatif  $n$ : le nombre d'itération qui correspond à l'index de temps.  $\delta, \varepsilon > 0$ : constants  $\hat{\sigma}_e^2, \hat{\sigma}_{ee}^2$ , Puissance de bruit estimée  $\lambda$ : Facteur d'oubli
Erreur :	$e(n) = d(n) - \mathbf{h}^T(n-1)\mathbf{x}(n)$	

Méthode d'estimation 3 :

$$\hat{\sigma}_{ee}^2(n) = \lambda' \hat{\sigma}_{ee}^2(n-1) + (1-\lambda') e(n)^2$$

$$\hat{\sigma}_x^2(n) = \lambda' \hat{\sigma}_x^2(n-1) + (1-\lambda') \mathbf{x}(n)^2$$

$$\mathbf{R}_{xe}(n) = \lambda' \mathbf{R}_{xe}(n-1) + (1-\lambda') \mathbf{x}(n)e(n)$$

$$\hat{\sigma}_v^2(n) = \hat{\sigma}_{ee}^2(n) - \frac{\mathbf{R}_{xe}^T(n)\mathbf{R}_{xe}(n)}{\hat{\sigma}_x^2(n) + c_0}$$

Mise-à-jour :  $\hat{\sigma}_e^2(n) = \lambda \hat{\sigma}_e^2(n-1) + (1-\lambda) e^2(n)$

$$\beta(n) = [\delta + \mathbf{x}^T(n)\mathbf{x}(n)]^{-1} [1 - \frac{\sigma_v}{\varepsilon + \hat{\sigma}_e(n)}]$$

$$\mu_{npvss}(n) = |\beta(n)|$$

$$\mu_{npvss}(n) = \begin{cases} \frac{\mu_{max}}{\delta + \mathbf{x}^T(n)\mathbf{x}(n)} & \text{if } n < Ts \\ |\beta(n)| & \text{if } n > Ts \end{cases}$$

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu_{NPVSS}(n) \mathbf{x}(n) e(n)$$

Fin

**Tableau 2.8.** NP-VSS-NLMS avec M3

Etapes	Algorithme	Description
Paramètre :	$\hat{\mathbf{h}}(0) = 0$ $\lambda = 1 - \frac{1}{KL}$ avec $K \geq 1$ $\mu_{min} = 0.001$ $\mu_{max} = \begin{cases} 0.8 & \text{if } SNR < 20 \\ 1 & \text{allieurs} \end{cases}$ $\alpha = 0.1$ $Ts = m * L, m = 20,30,40$	$L$ : est la longueur du filtre adaptatif $n$ : le nombre d'itération qui correspond à l'index de temps. $\delta, c_0 > 0$ : constants $\sigma_v^2$ , puissance de bruit estimé $\lambda$ : facteur d'oubli
Erreur :	$e(n) = d(n) - \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n)$	
Méthode d'estimation 3 :	$\hat{\sigma}_{ee}^2(n) = \lambda' \hat{\sigma}_{ee}^2(n-1) + (1 - \lambda') e(n)^2$ $\hat{\sigma}_x^2(n) = \lambda' \hat{\sigma}_x^2(n-1) + (1 - \lambda') \mathbf{x}(n)^2$ $\hat{R}_{xe}(n) = \lambda' \hat{R}_{xe}(n-1) + (1 - \lambda') \mathbf{x}(n)e(n)$ $\hat{\sigma}_v^2(n) = \hat{\sigma}_{ee}^2(n) - \frac{\hat{R}_{xe}^T(n)\hat{R}_{xe}(n)}{\hat{\sigma}_x^2(n) + c_0}$	
Mise-à-jour :	$\hat{\sigma}_d^2(n) = \lambda \hat{\sigma}_d^2(n-1) + (1 - \lambda) d(n)^2$ $\hat{\sigma}_y^2(n) = \lambda \hat{\sigma}_y^2(n-1) + (1 - \lambda) y^2(n)$ $\mu_{PROVSS}(n) = \alpha * \left  \frac{\hat{\sigma}_d^2(n) - \hat{\sigma}_y^2(n)}{\sigma_v^2 + \delta} \right $ $\mu_{PROVSS}(n) = \begin{cases} \mu_{max} & \text{if } \mu_{PROVSS}(n) > \mu_{max} \\ \mu_{min} & \text{if } \mu_{PROVSS}(n) < \mu_{min} \end{cases}$ $\mu_{PROVSS}(n) = \mu_{max} \text{ if } n < Ts$ $\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu_{PROVSS}(n) \frac{\mathbf{x}(n) e(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + c_0}$	
	Fin	

**Tableau 2.9.** VSS-NLMS-2020 avec M3

**Méthode 4 (M4) :**

La quatrième méthode d'estimation a été proposée en 2019 par [10], est une méthode d'estimation par la valeur absolue de  $e(n)$  :

$$\hat{\sigma}_e(n) = \lambda \hat{\sigma}_e(n-1) + (1-\lambda)|e(n)| \quad (2.37)$$

$$\hat{\sigma}_v^2(n) = \hat{\sigma}_e^2(n)$$

Les deux algorithmes NP-VSS-NLMS/VSS-NLMS-2020 appliqué par la méthode 4 d'estimation de la puissance de bruit (M4) sont résumés par les tableaux au-dessous (2.10/2.11) :

Etapes	Algorithme	Description
Paramètre :	$\hat{\mathbf{h}}(0) = 0$  $\lambda = 1 - \frac{1}{KL}$ Avec $K \geq 1$ $\mu_{\min} = 0.001$ $\mu_{\max} = \begin{cases} 0.8 & \text{if SNR} < 20 \\ 1 & \text{allieurs} \end{cases}$  $Ts = m * L, gain (Nstart)$	$L$ : est la longueur du filtre adaptatif  $n$ : le nombre d'itération qui correspond à l'index de temps.  $\delta, \varepsilon > 0$ : constants  $\hat{\sigma}_e^2, \hat{\sigma}_{ee}^2$ , Puissance de bruit estimée  $\lambda$ : Facteur d'oubli
Erreur :	$e(n) = d(n) - \mathbf{h}^T(n-1)\mathbf{x}(n)$	
Méthode d'estimation 4 :	$\hat{\sigma}_{ee}(n) = \lambda' \hat{\sigma}_{ee}(n-1) + (1-\lambda')  e(n) $  $\hat{\sigma}_v^2(n) = \hat{\sigma}_{ee}^2(n)$	
Mise-à-jour :	$\hat{\sigma}_e^2(n) = \lambda \hat{\sigma}_e^2(n-1) + (1-\lambda) e^2(n)$  $\beta(n) = [\delta + \mathbf{x}^T(n)\mathbf{x}(n)]^{-1} [1 - \frac{\sigma_v}{\varepsilon + \hat{\sigma}_e(n)}]$  $\mu_{npvss}(n) =  \beta(n) $  $\mu_{npvss}(n) = \begin{cases} \frac{\mu_{max}}{\delta + \mathbf{x}^T(n)\mathbf{x}(n)} & \text{if } n < Ts \\  \beta(n)  & \text{if } n > Ts \end{cases}$  $\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu_{NPVSS}(n) \mathbf{x}(n) e(n)$	
<b>Fin</b>		

**Tableau 2.10.** NP-VSS-NLMS avec M4

Etapes	Algorithme	Description
Paramètre : adaptatif	$\hat{h}(0) = 0$	$L$ : est la longueur du filtre
d'itération qui correspond à l'index de temps.	$\lambda = 1 - \frac{1}{KL}$ avec $K \geq 1$	$n$ : le nombre $\mu_{min} = 0.001$
	$\mu_{max} = \begin{cases} 0.8 & \text{if } SNR < 20 \\ 1 & \text{allieurs} \end{cases}$	$\delta, c_0 > 0$ : constants
	$\alpha = 0.1$	$\sigma_v^2$ , puissance de bruit connue ou estimé
	$Ts = m * L, m = 20,30,40$	$\lambda$ : facteur d'oubli
Erreur :	$e(n) = d(n) - \hat{h}^T(n-1)\mathbf{x}(n)$	
Méthode d'estimation 4 :	$\hat{\sigma}_{ee}(n) = \lambda' \hat{\sigma}_{ee}(n)(n-1) + (1 - \lambda')  e(n) $ $\hat{\sigma}_v^2(n) = \hat{\sigma}_{ee}^2(n)$	
Mise-à-jour :	$\hat{\sigma}_d^2(n) = \lambda \hat{\sigma}_d^2(n-1) + (1 - \lambda) d(n)^2$ $\hat{\sigma}_y^2(n) = \lambda \hat{\sigma}_y^2(n-1) + (1 - \lambda) y^2(n)$ $\mu_{PROVSS}(n) = \alpha * \left  \frac{\hat{\sigma}_d^2(n) - \hat{\sigma}_y^2(n)}{\sigma_v^2 + \delta} \right $ $\mu_{PROVSS}(n) = \begin{cases} \mu_{max} & \text{if } \mu_{PROVSS}(n) > \mu_{max} \\ \mu_{min} & \text{if } \mu_{PROVSS}(n) < \mu_{min} \end{cases}$ $\mu_{PROVSS}(n) = \mu_{max} \quad \text{if } n < Ts$	
	$\hat{h}(n) = \hat{h}(n-1) + \mu_{PROVSS}(n) \frac{\mathbf{x}(n) e(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + c_0}$	
	<b>Fin</b>	

**Tableau 2.11.** VSS-NLMS-2020 avec M4

## 2.5 Conclusion

Dans ce chapitre, nous avons présentés quelques algorithmes NLMS à pas variable VSS-NLMS adaptés aux applications AEC en présence du bruit additif. Ces algorithmes sont basés sur différentes stratégies de contrôle pour ajuster leurs principaux paramètres, c'est-à-dire le pas normalisé, terme de régularisation, afin d'atteindre un compromis entre les critères de rendement (c.-à-d. vitesse de convergence et la capacité de poursuivre). Ensuite nous avons appliqué les méthodes d'estimation de bruit avec les algorithmes NP-VSS-2006 et VSS-NLMS-2020. Dans le prochain chapitre, nous voulons tester les méthodes d'estimations de la variance du bruit avec méthode théorique et les algorithmes (NP\_VSS\_NLMS 2006, VSS-NLMS-2020) pour voir quels algorithmes VSS avec les méthodes d'estimation de la variance du bruit, sont les plus robustes et capables d'atténué l'écho acoustique et en même temps poursuivre les non stationnarités du canal acoustique et pour conclure sur la méthode d'estimation la plus proche et la plus fiable à la méthode théorique.



## Chapitre 3 : RESULTATS DE SIMULATIONS

---

### 3.1 Introduction

Dans ce chapitre, nous allons tester et comparer les performances des algorithmes VSS-NLMS appliquée par les quatre méthodes d'estimation de variance du bruit présentés dans le chapitre précédent, plusieurs simulations numériques ont été réalisées à l'aide du logiciel MATLAB. Les performances qui nous intéressent dans cette étude sont exprimées en termes de vitesse de convergence et de niveau d'erreur de filtrage final en sortie par rapport à la méthode théorique. Avant la présentation des résultats de simulation de ces algorithmes VSS-NLMS, nous allons décrire les signaux utilisés dans ce travail.

### 3.2 Présentation des signaux de simulations

Les signaux de nos simulations ont été générés selon le schéma décrit sur la figure 3.1. La fréquence d'échantillonnage est de 16 kHz et l'amplitude est codée sur 16 bits.

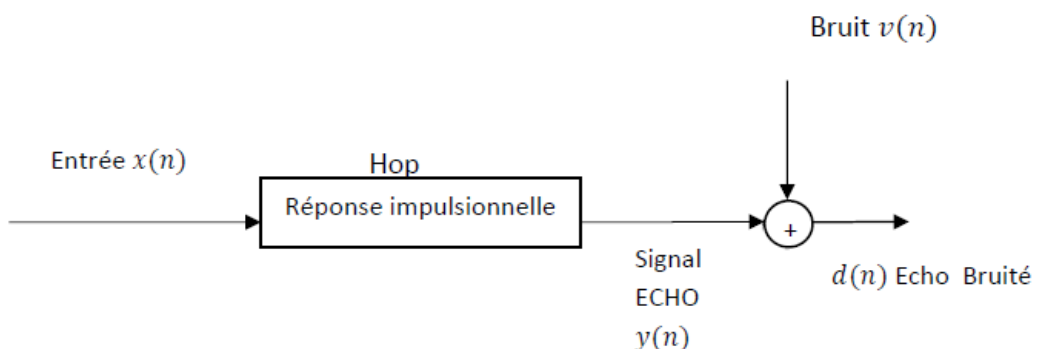
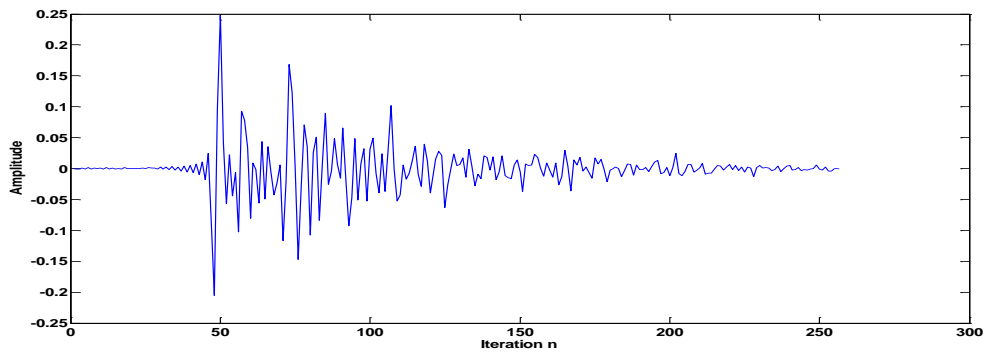
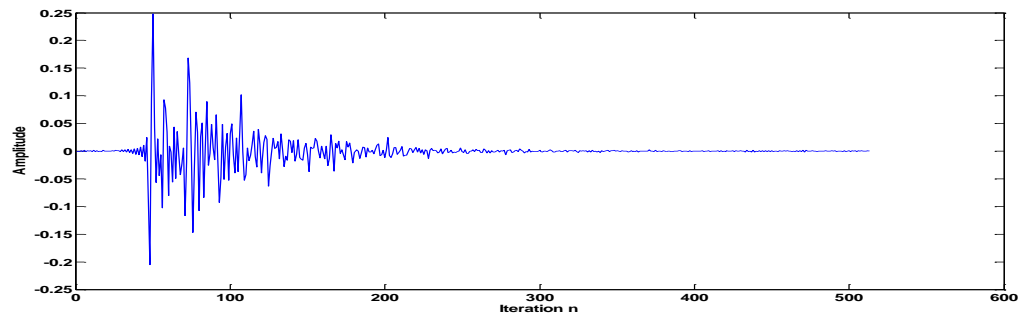


Figure 3.1 : Schéma de génération des signaux

Nous utilisons deux différents types de signaux d'entre  $x(n)$  stationnaires, qui sont ensuite filtrés par une réponse impulsionnelle mesurée dans l'habitacle d'une voiture (filtre FIR inconnu Hop) de taille maximale 1024 points tronquées à 256 et 512 points (Figures 3.2a et 3.2b), pour obtenir les signaux écho non bruité, enfin nous ajoutons un bruit  $v(n)$  pour obtenir le signal de sortie écho bruité  $d(n)$  (ou signal désiré).



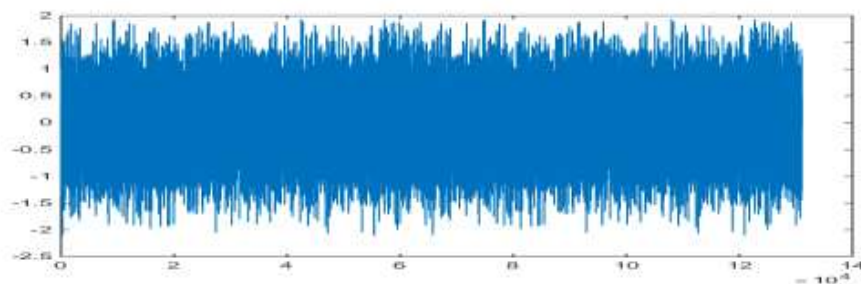
**Figure 3. 2a** : Réponse impulsionnelle mesurée ( L=256).



**Figure 3.2b** : Réponse impulsionnelle mesurée (L=512).

**a Signal USASI :**

C'est un bruit corrélé stationnaire avec un spectre équivalent au spectre moyen de la parole, appelé bruit USASI (United States of America Standards Institute) dans le domaine de l'annulation d'écho acoustique. Ce signal de moyenne nulle possède une dynamique spectrale de 32 dB.



**Figure 3.3:** Signal USASI

**b Signal BBG1AR20 :**

C'est un signal stationnaire, fortement corrélé, généré par un modèle AR d'ordre 20 déduit de l'analyse LPC de la voyelle « U ». Sa dynamique spectrale est d'environ de 42dB.

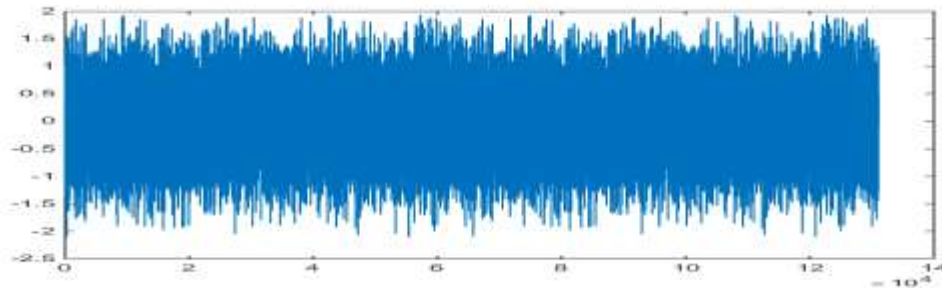


Figure 3.4 : signal bbg1AR20.

**c Signal parole :**

C'est un signal vocal typique non stationnaire échantillonnés à 16 kHz, est filtré par une vraie réponse impulsionnelle acoustique mesurée dans une vraie voiture pour obtenir les signaux souhaités.

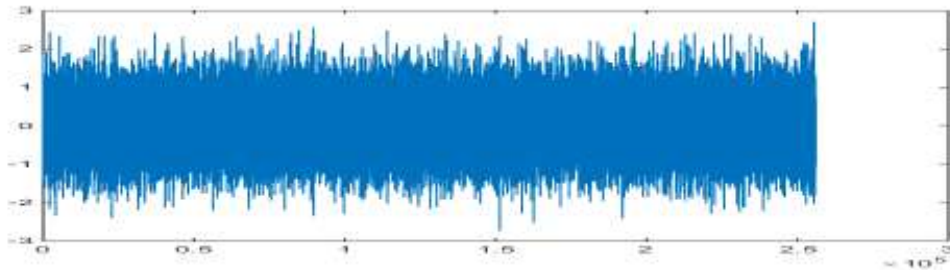


Figure 3.5 signal parole

**d Le bruit additif  $v(n)$  :**

Le bruitage du signal écho est réalisé par la fonction Matlab `awgn(y, SNR, 'measured')`.

Cette fonction ajoute au signal écho  $y(n)$  un bruit blanc Gaussien  $v(n)$  avec un rapport signal à bruit SNR en dB.

Nous avons besoin de la puissance du bruit  $\sigma_v^2$  qui sera calculée par la méthode théorique comme suit :

$$SNR = 10 \log_{10} \frac{\sigma_y^2}{\sigma_v^2} \quad (3.1)$$

$$\sigma_v^2 = 10^{-\frac{SNR}{10}} \cdot \sigma_y^2 \quad (3.2)$$

Où  $\sigma_y^2$  désigne la puissance de l'écho.

### e Non stationnarité du canal acoustique

Pour pouvoir discuter l'une des performances les plus importantes d'un algorithme adaptatif ; c'est-à-dire sa capacité à suivre les variations du système inconnu, nous avons introduit une variation artificielle réalisée par la multiplication du signal écho par un gain variable linéaire pendant une période de temps finie entre les échantillons 55000 et 80000. L'amplitude de variation est comprise entre 1 et 2.5 (figure 3.6).

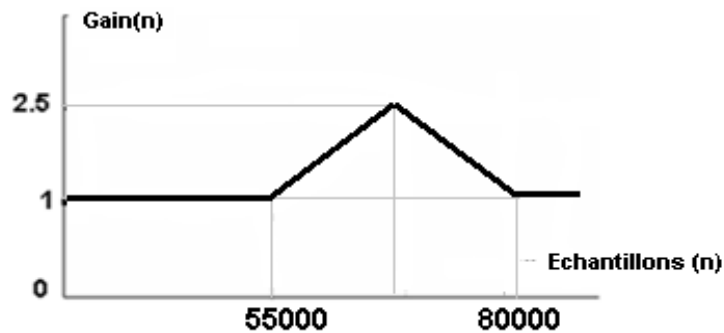


Figure 3. 6 : Variation linéaire du canal

### 3.3 Description des critères de performance :

Pour comparer les performances des algorithmes entre eux, un critère objectif couramment utilisé est l'EQM (MSE) donnée par :

$$MSE(dB) = 10\log(\langle e^2(n) \rangle) \quad (3.3)$$

Où  $\langle \cdot \rangle$  désigne une moyenne évaluée sur des moyennes temporelles de 256 ou 512 échantillons consécutifs.

Pour évaluer les algorithmes VSS, nous avons considérés trois performances différentes : la vitesse de convergence, la valeur de MSE en régime permanent et la capacité de poursuite des variations de la réponse impulsionnelle du canal.

### 3.4 Résultats de simulations des algorithmes VSS-NLMS :

Dans ce qui suit, nous allons présenter les résultats comparatifs des algorithmes VSS-NLMS et NLMS avec pas d'adaptations fixe  $\mu = 0.8$  ou  $\mu = 1$  et en utilisant trois signaux d'entrées USASI et Bbg1ar20 et signal parole dans les deux cas stationnaire et non

stationnaire pour les valeurs de SNR suivantes SNR =15dB, 20dB, 30dB et 50dB avec un système de L=256 ou L=512.

La constante de régularisation du NLMS est fixée à 0.01.

### 3.4.1 Algorithme VSS-NLMS non paramétrique de BENESTY 2006 (VSS-NLMS NP)

L'algorithme VSS-NLMS NP avec la méthode d'estimation M1 :

A l'algorithme VSS-NLMS NP décrit au chapitre précédent, on a intégré ces trois équations de la méthode d'estimation de bruit M1 comme suit :

$$\hat{\sigma}_d^2(n) = \lambda' \hat{\sigma}_d^2(n-1) + (1 - \lambda') d(n)^2$$

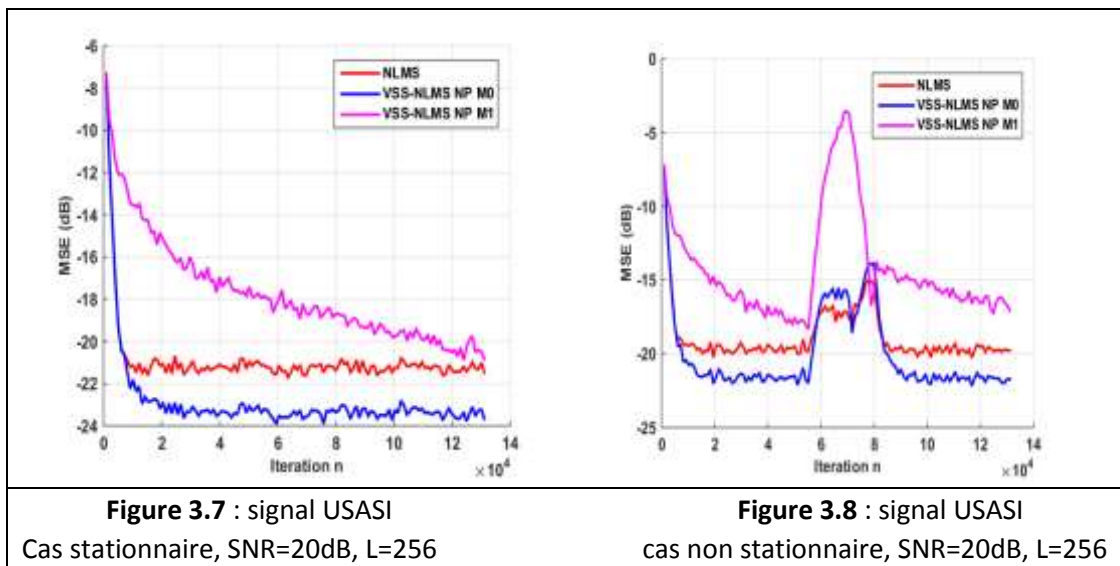
$$\hat{\sigma}_y^2(n) = \lambda' \hat{\sigma}_y^2(n-1) + (1 - \lambda') \hat{y}^2(n)$$

$$\hat{\sigma}_v^2(n) = |\hat{\sigma}_d^2(n) - \hat{\sigma}_y^2(n)|$$

Pour obtenir les résultats suivants, on a utilisé ces valeurs des paramètres :

$$\varepsilon = 0.000001, \lambda = 0.46, \lambda' = 0.9987, \hat{\sigma}_e^2(0) = 0.001, \hat{\sigma}_d^2(0) = 0.01, \hat{\sigma}_y^2(0) = 0.01, \delta = 20 * \sigma_x^2, \sigma_x^2 = 0.3272.$$

Qui l'ont appliqué avec deux types de canal : stationnaire et non stationnaire, en utilisant un signal stationnaire USASI :



**Commentaire** : Comme il est visible sur les résultats de simulation, l'algorithme VSS-NLMS NP avec la méthode d'estimation M1 ne converge pas rapidement dans les premières étapes de la convergence dans les deux cas stationnaire et non stationnaire

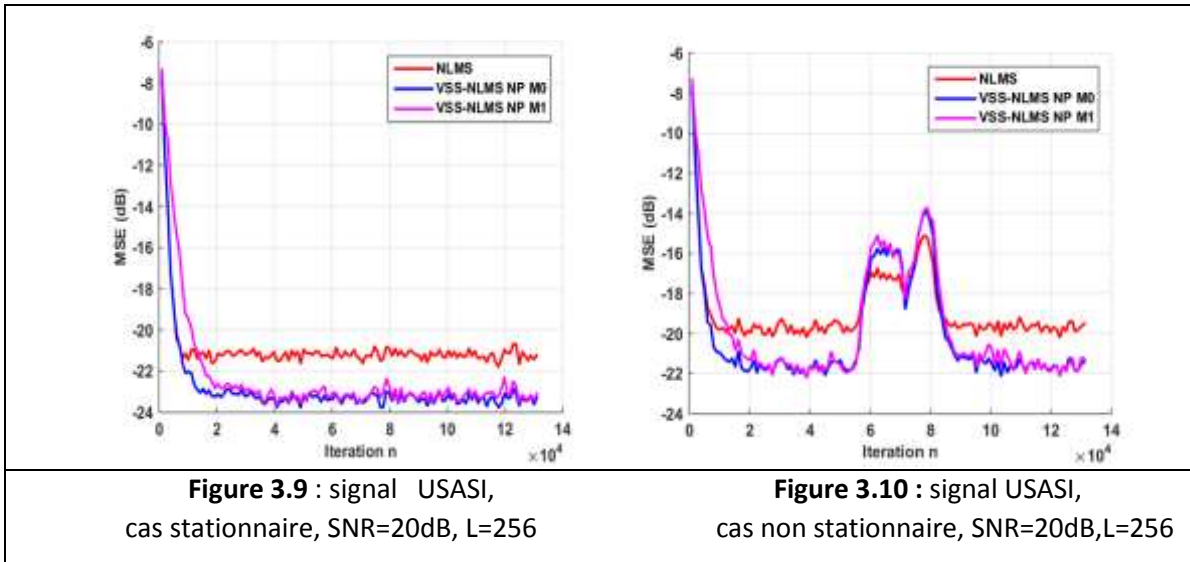
( car la condition  $\mu_{NPVSS} = 0$  bloque la mise à jour de filtre), ainsi qu'on a observé une divergence dès l'arrivée du non stationnarité.

- ✚ Pour résoudre ce problème, on a proposé au lieu de mettre  $\mu_{NPVSS} = 0$  pour  $\beta < 0$  ( $\hat{\sigma}_e > \sigma_v$ ), on va utiliser la valeur absolue de  $\beta(n)$  pour calculer le pas d'adaptation :

$$\beta(n) = [\delta + \mathbf{x}^T(n)\mathbf{x}(n)]^{-1} \left[ 1 - \frac{\sigma_v}{\varepsilon + \hat{\sigma}_e(n)} \right]$$

$$\mu_{npvss}(n) = |\beta(n)|$$

En appliquant cette proposition, on a obtenu les résultats suivants avec les mêmes paramètres utilisés précédemment :

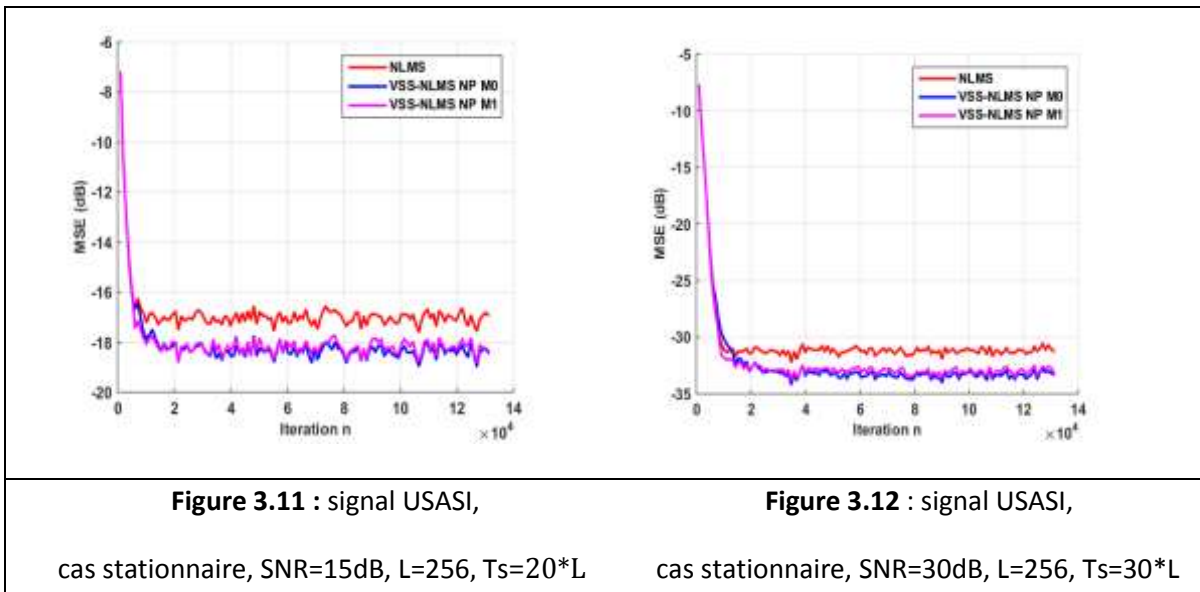


**Commentaire** : D'après les résultats obtenus dans les figures ci-dessus, on peut dire que la proposition de la valeur absolue résout le problème d'adaptation de l'algorithme NP-VSS-NLMS avec la méthode M1, mais malgré cette solution on remarque que la vitesse de convergence est ralentie dans la phase de convergence.

- ✚ Pour cela et afin de rendre la vitesse plus rapide, nous proposons d'utiliser la meilleure valeur de pas d'adaptation qui est le  $\mu_{max}$  du NP-VSS-NLMS durant la phase initiale de durée  $T_s$ , pour lui donner un certain temps afin d'atteindre les vraies valeurs de l'estimateur.

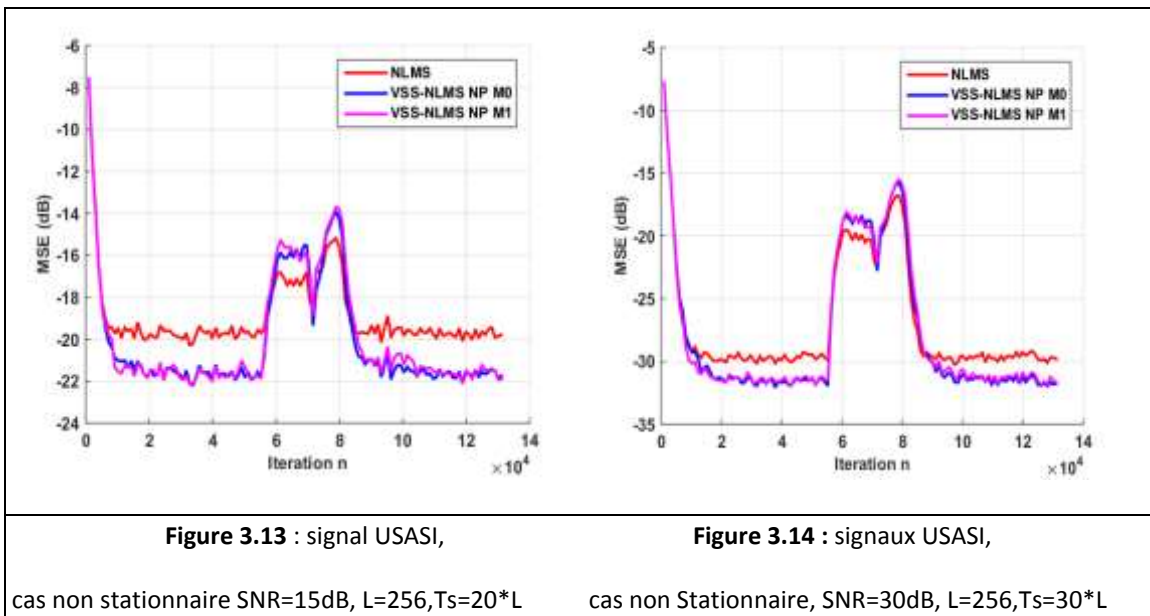
Avec l'application de toutes ces propositions quelques résultats de simulation de cet algorithme pour les deux cas stationnaire et non stationnaire sont résumés sur les figures 3.11 à 3.17.

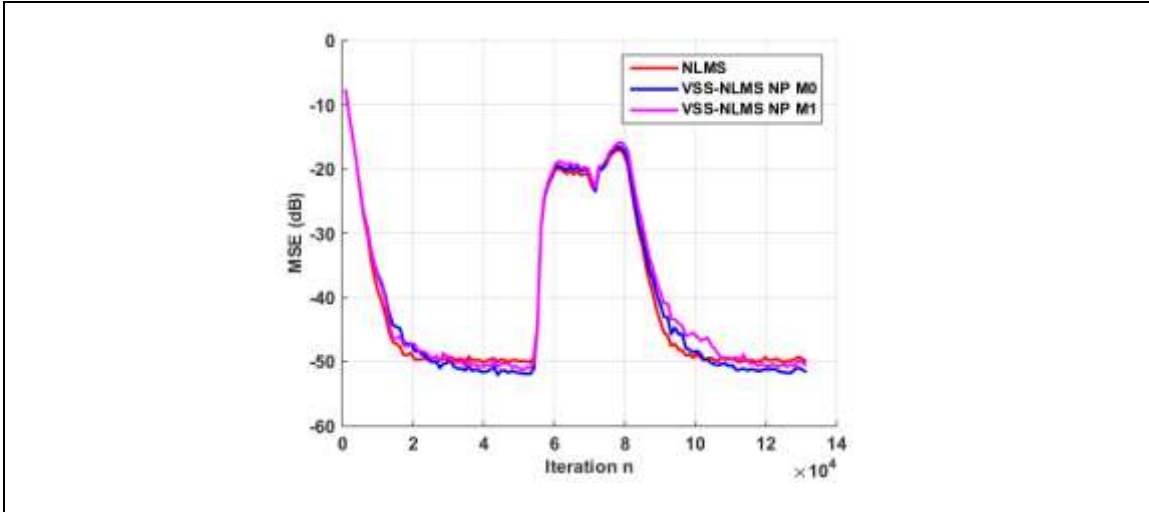
**Canal stationnaire :**



**Commentaire :** Pour le cas stationnaire, on a obtenu une vitesse de convergence similaire que celle du NP VSS-NLMS avec la méthode théorique (M0) pour les deux types de signaux. D'autre part, on remarque qu'ils ont des comportements presque identiques d'EQM en régime permanent.

**Canal non stationnaire :**



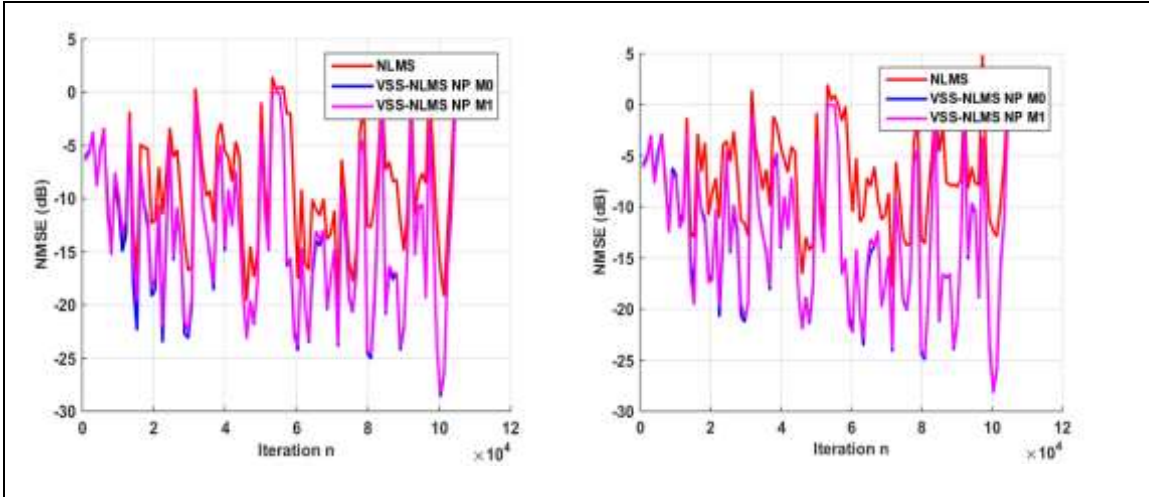


**Figure 3.15** : signal USASI, cas non Stationnaire, SNR=50dB, L=256, Ts=50\*L

**Commentaire** : Pour le cas non stationnaire , on observe qu'on a une bonne capacité de poursuite des non stationnarités, il arrive à reconverger vers une EQM finale semblables avants et après le passage de la non stationnarité du canal avec SNR de 20 et 30 dB.

**Signal parole :**

- **Canal stationnaire:**



**Figure 3.16** : signal parole,  
cas Stationnaire, SNR=20dB, L=256

**figure 3.17** : signal parole,  
cas Stationnaire SNR=20dB, L=512

**Commentaire** : Pour le cas parole, sachant que  $T_s > 0$  on a remarqué que le comportement de MSE normalisé de l'algorithme NPVSS-NLMS avec M1 est similaire avec NPVSS-NLMS avec M0 et qui sont meilleurs que le NLMS traditionnel.



Enfin, nous pouvons conclure après les modifications de l'algorithme VSS-NLMS NP avec la M1 au vu de ces performances similaires que la méthode théorique en termes de vitesse de convergence et la capacité de poursuite et l'EQM finale

### L'algorithme NPVSS-NLMS avec la méthode d'estimation M2

A l'algorithme VSS-NLMS NP décrit ci-dessus, on a intégré ces équations de la méthode d'estimation de bruit M2 comme suit :

$$\hat{\sigma}_{ee}^2(n) = \lambda' \hat{\sigma}_{ee}^2(n-1) + (1 - \lambda') e(n)^2$$

$$\hat{\sigma}_v^2(n) = \hat{\sigma}_{ee}^2(n)$$

On a obtenu les résultats par la suite, en utilisant les valeurs des paramètres suivantes :

$$\varepsilon = 0.000001, \lambda = 0.999, \lambda' = 0.85, \hat{\sigma}_e^2(0) = 0.001, \hat{\sigma}_{ee}^2(0) = 0.01, \delta = 20 * \sigma_x^2$$

Les figures 3.18 à 3.23 représentent quelques résultats obtenus de l'évaluation du critère MSE d'algorithmes NP-VSS-NLMS avec M2 appliqué par les deux propositions expliquées dans la M1 pour le cas stationnaire et non stationnaire avec le signal USASI.

#### Canal stationnaire :

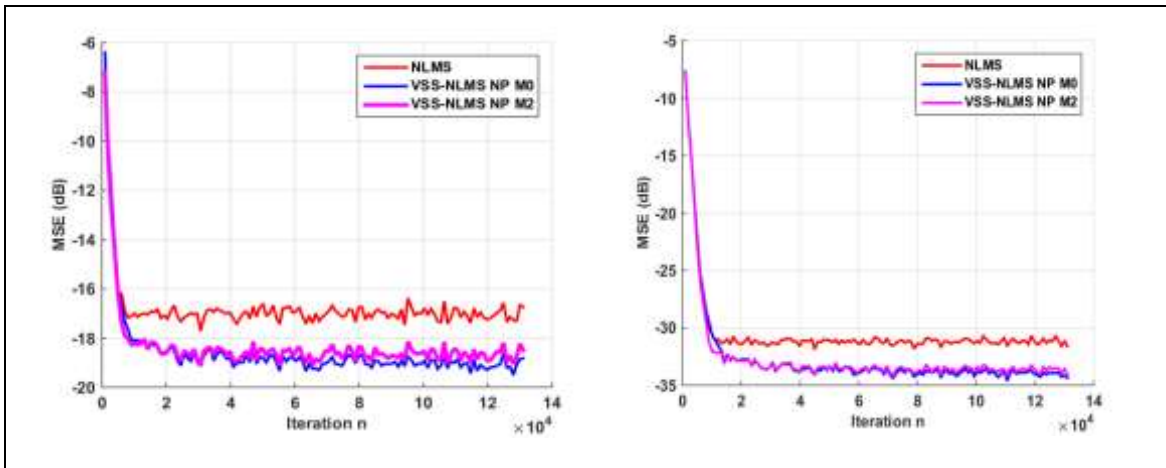


Figure 3.18 : signal USASI,

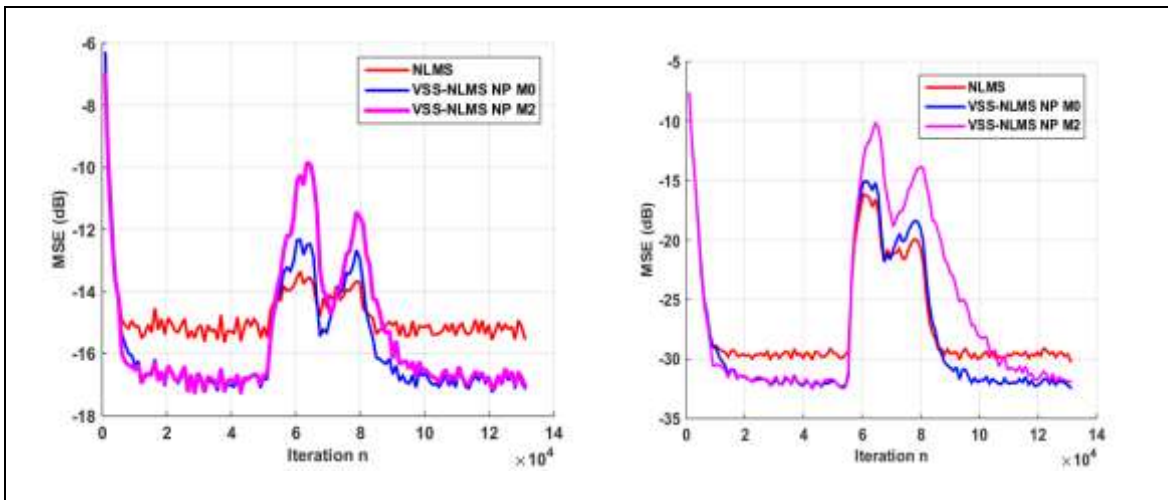
Figure 3.19: signal USASI ,

cas stationnaire, SNR=15dB, L=256, Ts=20\*L

cas stationnaire, SNR=30dB, L=256, Ts=30\*L

**Commentaire :** Ce que l'on constate sur les figures ci-dessus pour le cas stationnaire, la vitesse de convergence obtenue est équivalente à celle la méthode théorique M0, mais avec un EQM finale plus important pour un SNR=15dB.

**Canal non stationnaire :**



**Figure 3.20 :** signal USASi,

cas non stationnaire SNR=15dB, L=256, Ts=20\*L

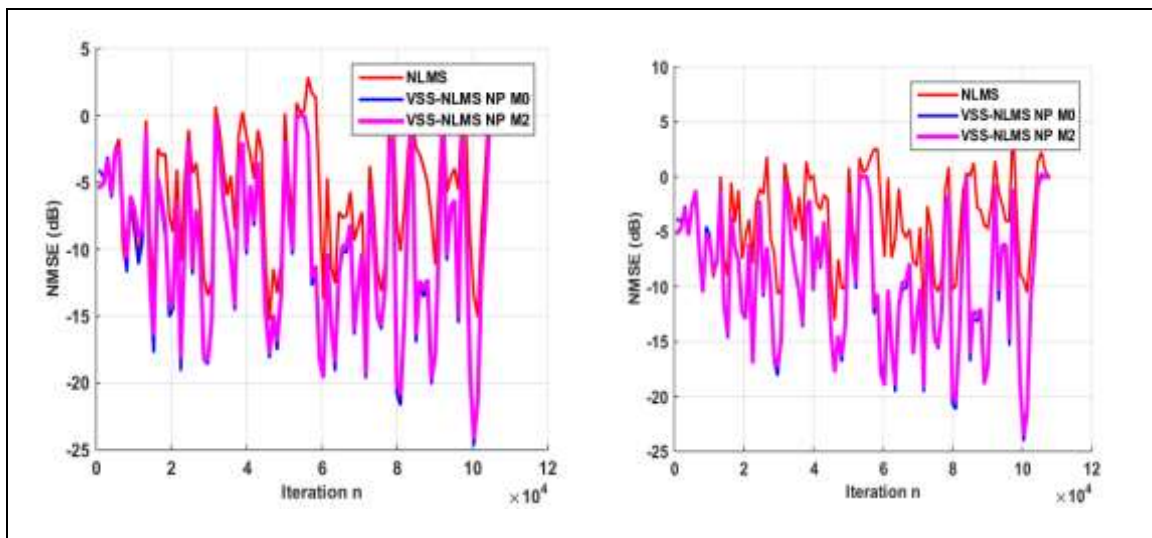
**Figure 3.21 :** signal USASi,

cas non stationnaire, SNR=30dB, L=256, Ts=30\*L

**Commentaire :** les tests révèlent que l’algorithme NPVSS-NLMS avec M2 on note une perte de poursuite et se dégrade a un mauvais comportement en poursuite pour les cas moins bruité, ainsi qu’il arrive à reconverger vers une EQM finale plus importante après la convergence et après le passage de la non stationnarité.

**Signal parole :**

- **canal stationnaire :**



**Figure 3.22 :** signal parole

cas stationnaire, SNR=15dB, L=256

**Figure 3.23 :** signal parole

cas stationnaire, SNR=15, L=512

**Commentaire :** Pour le cas parole, sachant que  $T_s > 0$  on a remarqué que le comportement de MSE normalisé de l’algorithme NPVSS-NLMS avec M2 est similaire au NPVSS-NLMS avec M0 et qui est meilleur que le NLMS classique.

### L'algorithme VSS-NLMS NP avec la méthode d'estimation M3

A l'algorithme VSS-NLMS NP décrit ci-dessus, on a intégré ces quatre équations de la méthode d'estimation de bruit M3 comme suit :

$$\hat{\sigma}_{ee}^2(n) = \lambda' \hat{\sigma}_{ee}^2(n-1) + (1 - \lambda') e(n)^2$$

$$\hat{\sigma}_x^2(n) = \lambda' \hat{\sigma}_x^2(n-1) + (1 - \lambda') \mathbf{x}(n)^2$$

$$\hat{R}_{xe}(n) = \lambda' \hat{R}_{xe}(n-1) + (1 - \lambda') \mathbf{x}(n)e(n)$$

$$\hat{\sigma}_v^2(n) = \hat{\sigma}_{ee}^2(n) - \frac{\hat{R}_{xe}^T(n) \hat{R}_{xe}(n)}{\hat{\sigma}_x^2(n) + c_0}$$

Pour obtenir les résultats ci-dessous, on a utilisé les valeurs des paramètres suivants :

$$\varepsilon = 0.000001, \lambda = 0.6, \lambda' = 0.99998, \hat{\sigma}_e^2(0) = 0.001, \hat{\sigma}_{ee}^2(0) = 0.01,$$

$$\hat{\sigma}_x^2(0) = 0.01, \delta = 20 * \sigma_x^2, c_0 = 0.001, R_{xe}(0) = 0.$$

Avec l'application de toutes les propositions montrées dans la M1, on a une sélection de résultats est donnée sur les figures ci-dessus (3.24 à 3.30) :

#### Canal stationnaire :

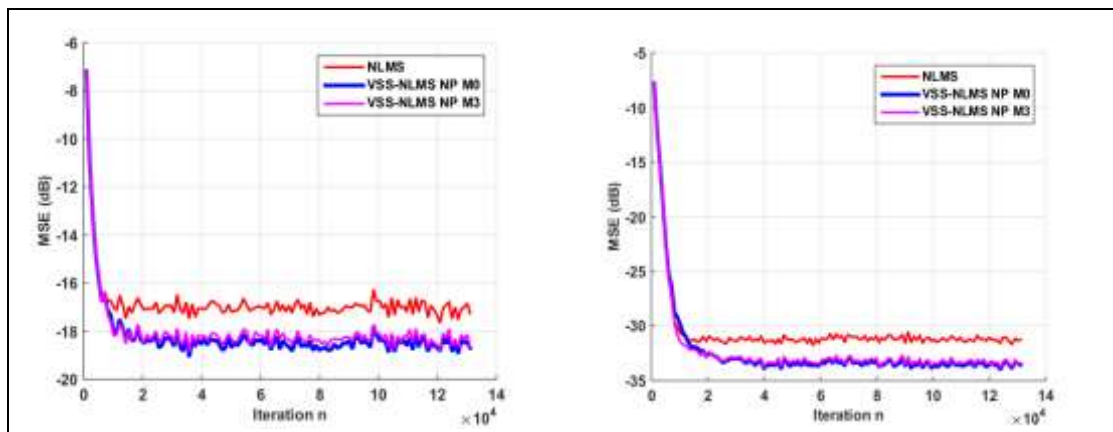


Figure 3.24 : signal USASI,

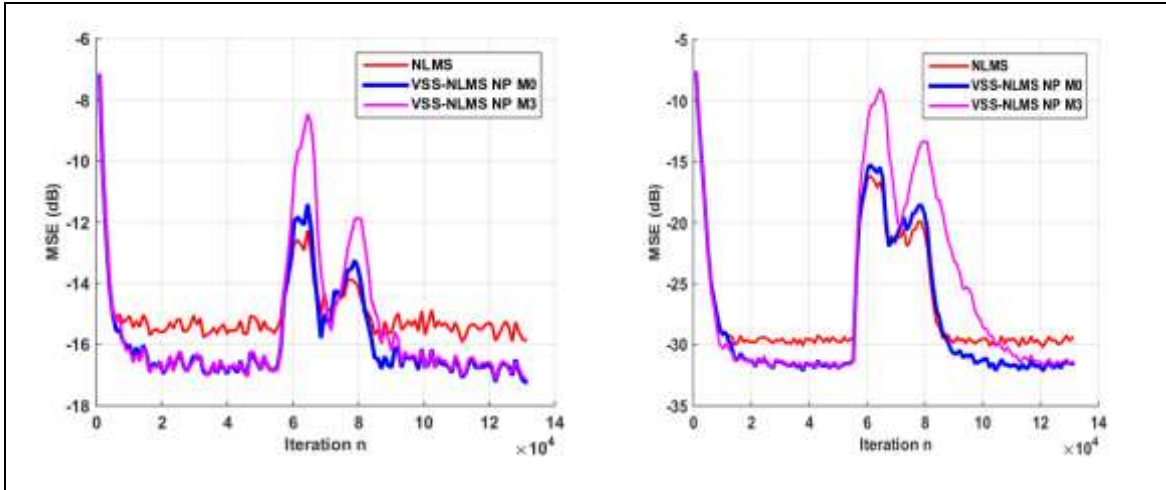
Figure 3.25 : signal USASI,

cas stationnaire, SNR=15dB, L=256, Ts=15\*L

cas stationnaire, SNR=30dB, L=256, Ts=30\*L

**Commentaire :** pour un  $T_s > 0$  la vitesse de convergence obtenue avec M3 est similaire par rapport à M0, l'EQM finale de VSS-NLMS NP avec M3 est identique avec M0 pour tous les SNR considérés.

- **Canal non stationnaire :**

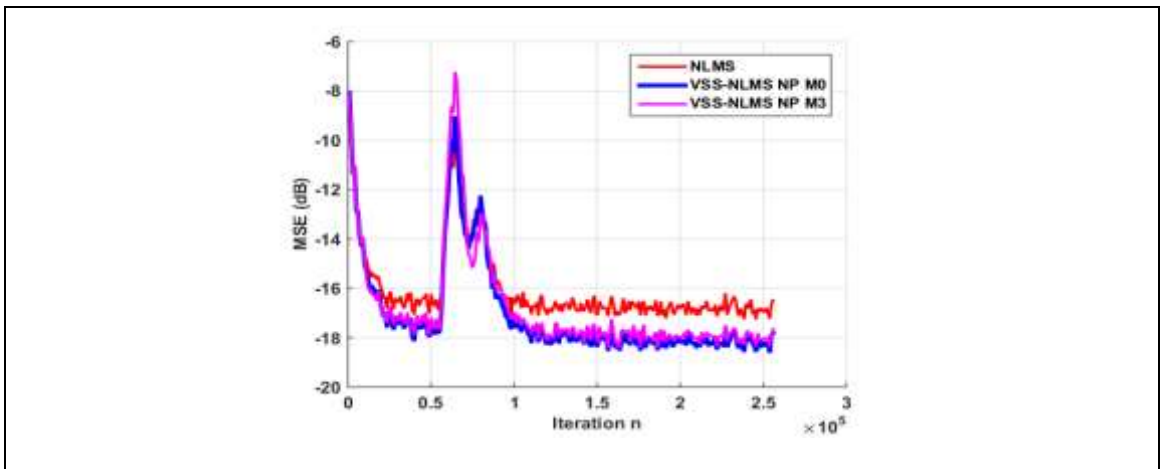


**Figure 3.26 :** signal USASI,

**Figure 3.27 :** signal USASI,

cas non stationnaire SNR=15dB, L=256,  $T_s=15*L$

cas non stationnaire, SNR=30dB, L=256,  $T_s=30*L$

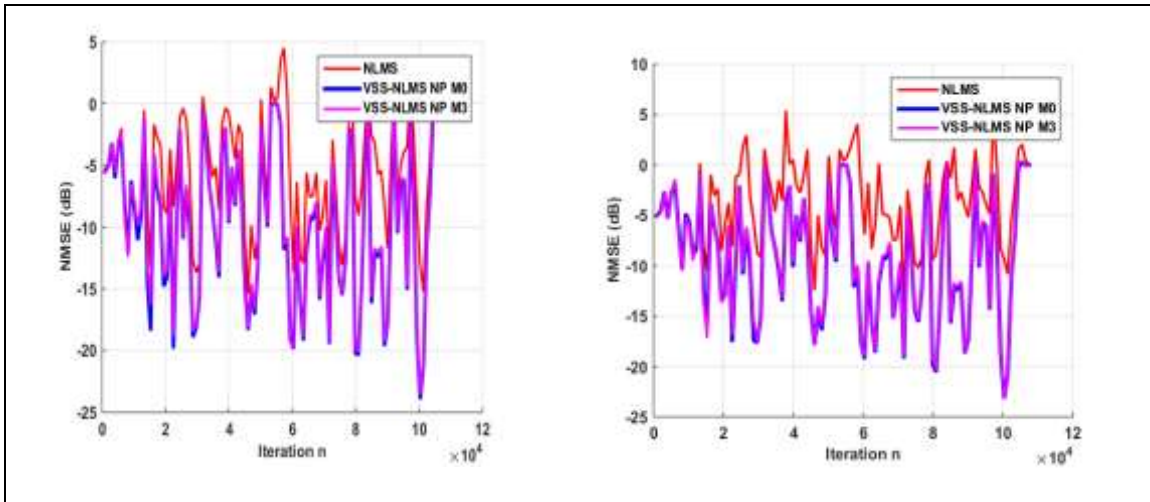


**Figure 3.28:** signal BBG1AR20, cas non stationnaire SNR=15dB, L=256,  $T_s=30*L$

**Commentaire :** En ce qui concerne les résultats révèlent que l'algorithme VSS-NLMS NP avec la méthode M3 a une perte capacité de poursuite devient plus mauvais dans le cas moins bruité, d'autre part on remarque une bonne capacité de poursuite pour le signal BBG1AR20, ainsi qu'ils arrivent à reconverger vers une EQM similaire après la convergence et après le passage du non stationnarité pour le cas bruité SNR=15dB.

**Signal parole**

- **canal stationnaire :**



**Figure 3.29** : signal parole,

cas stationnaire, SNR=15dB, L=256

**Figure 3.30** : signal parole,

cas non stationnaire SNR=15 L=256

**Commentaire** : Pour le cas parole, sachant que  $T_s > 0$  on a remarqué que le comportement de MSE normalisé de l'algorithme NPVSS-NLMS avec M3 est similaire avec NPVSS-NLMS avec M0 et qui sont plus meilleurs que NLMS traditionnel.

Enfin nous pouvons dire concerne les résultats obtenus pour l'algorithme NP-VSS-NLMS modifié avec M2 et M3 qu'il ne remplit pas tous les critères à atteindre ce qui concerne la poursuite et l'erreur finale.

#### L'algorithme VSS-NLMS NP avec la méthode d'estimation M4

A l'algorithme VSS-NLMS NP décrit ci-dessus, on a intégré ces deux équations de la méthode d'estimation de bruit M4 comme suit :

$$\hat{\sigma}_{ee}(n) = \lambda' \hat{\sigma}_{ee}(n-1) + (1 - \lambda') |e(n)|$$

$$\hat{\sigma}_v^2(n) = \hat{\sigma}_{ee}^2(n)$$

Pour obtenir les résultats suivants, on a utilisé les paramètres :

$$\varepsilon = 0.000001, \lambda = 0.7, \lambda' = 0.99998, \hat{\sigma}_e^2(0) = 0.01, \hat{\sigma}_{ee}(0) = 0.01, \delta = 20 * \sigma_x^2, \sigma_x^2 = 0.3272.$$

#### Signal stationnaire

- Canal stationnaire :

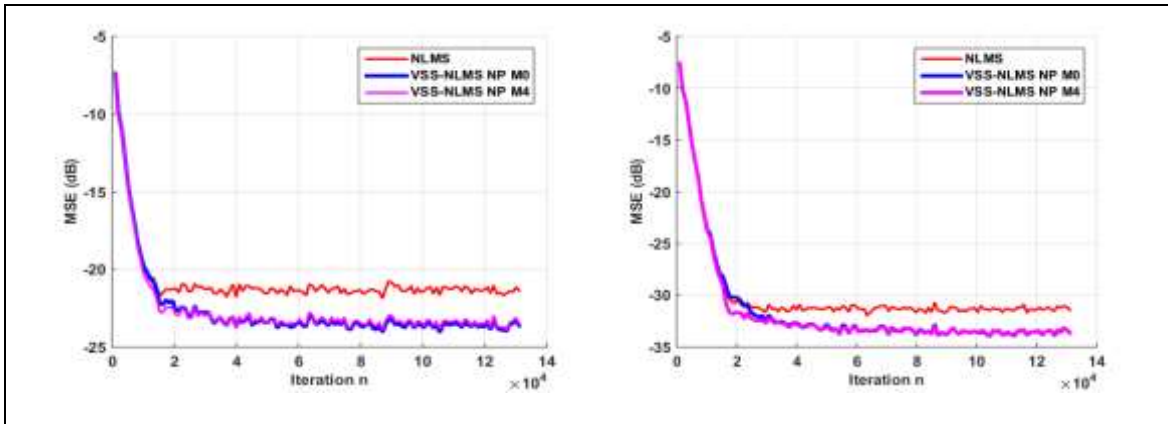


Figure 3.31 : signal USASI,

cas stationnaire, SNR=20dB, L=512 ,Ts=20\*L

Figure 3.32 : signal USASI,

cas stationnaire, SNR=30dB, L=512,Ts=30\*L

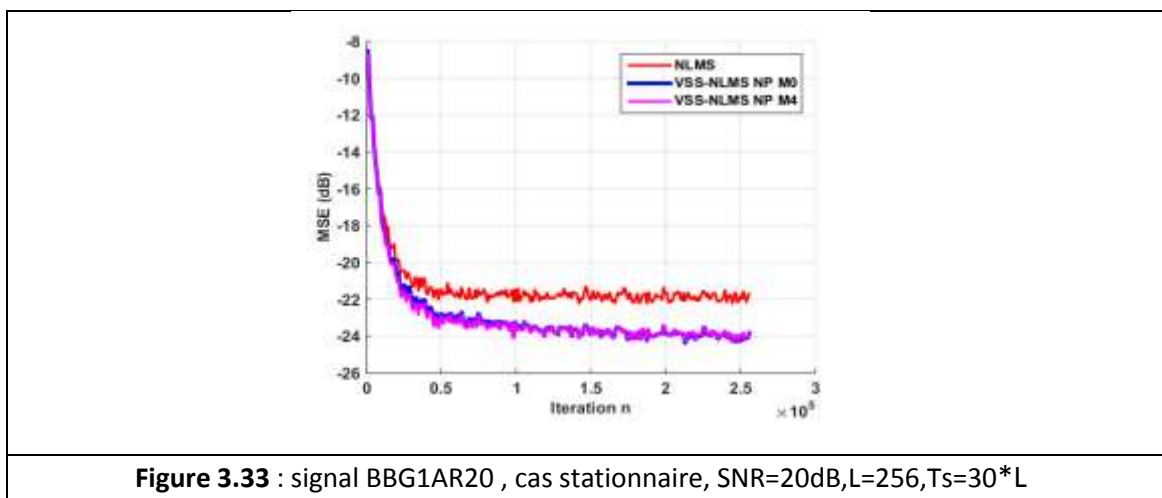


Figure 3.33 : signal BBG1AR20 , cas stationnaire, SNR=20dB,L=256,Ts=30\*L

**Commentaire :** Après la proposition de  $T_s$  nous remarquons une vitesse de convergence équivalente que de  $M_0$  dans tous les cas de SNR et avec un EQM finale similaire en régime permanent.

- **Canal non stationnaire :**

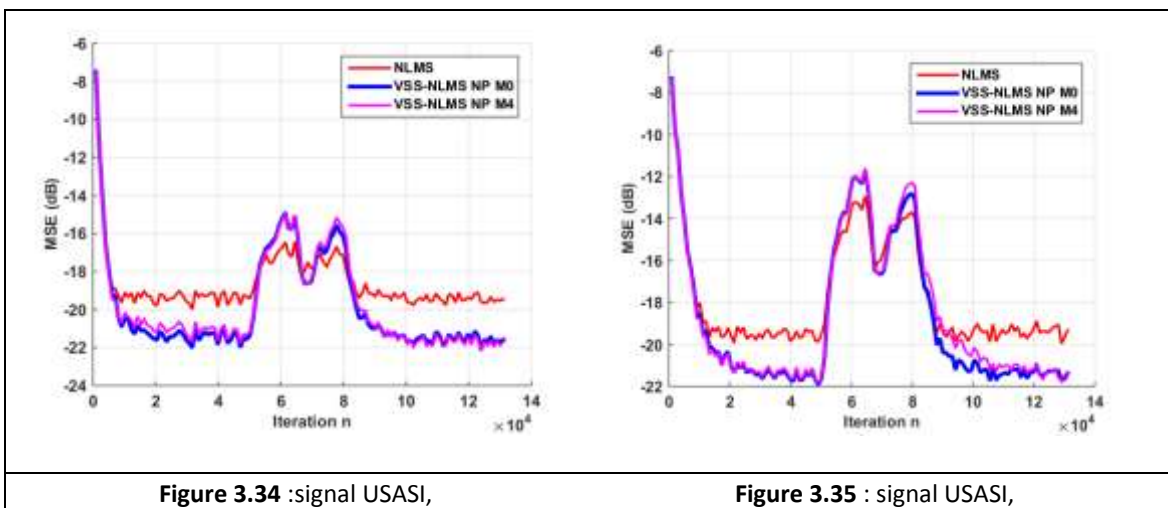


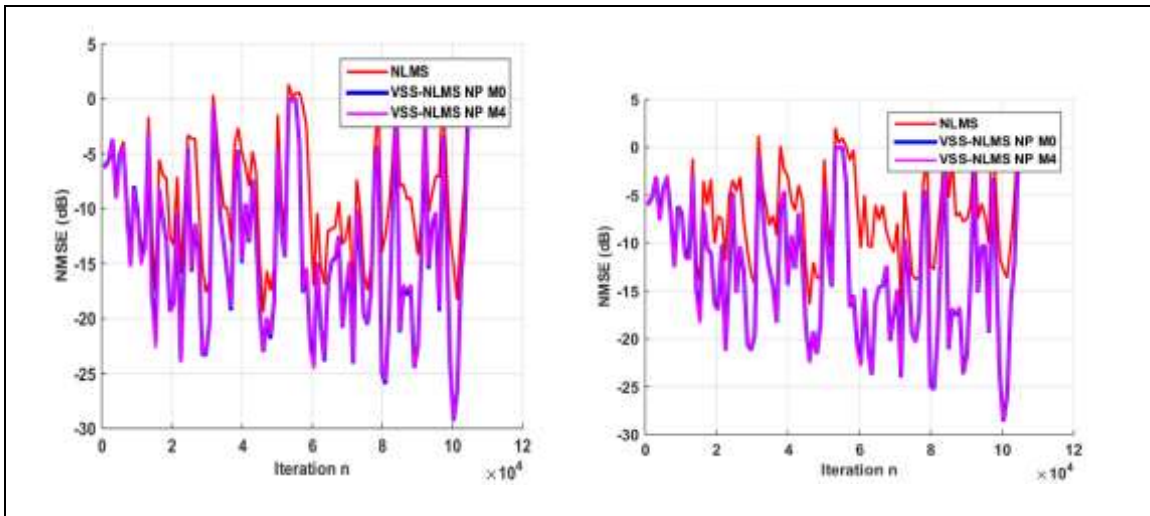
Figure 3.34 :signal USASI,

Figure 3.35 : signal USASI,

**Commentaire :** on observe une bonne capacité de poursuite pour l'algorithme VSS-NLMS NP avec la M4 , ainsi qu'il arrive à reconverger vers une EQM plus faible après la convergence et après le passage du non stationnarité dans le cas bruité SNR=20dB dans les deux types de signal (USASI et BBG1AR20).

### Signal parole

- **Canal stationnaire :**



**Figure 3.36 :** signal parole ,

**Figure 3.37 :** signal parole ,

cas stationnaire ,SNR=20dB,L=256

cas non stationnaire , SNR=20 ,L=256

**Commentaire :** Pour le cas parole, sachant que  $T_s > 0$  on a remarqué que le comportement de MSE normalisé de l'algorithme NPVSS-NLMS avec M4 est similaire avec NPVSS-NLMS avec M0 et qui sont meilleurs que NLMS traditionnel.

Les résultats obtenus dans les simulations l'algorithme VSS-NLMS NP modifié avec la méthode d'estimation de bruit M4 est équivalente que méthode théorique en termes de vitesse de convergence et capacité de poursuite avec une amélioration d'EQM finale pour cas bruité.

### 3.4.2 Algorithme VSS-NLMS 2020

#### L'algorithme VSS-NLMS 2020 avec la méthode d'estimation M1

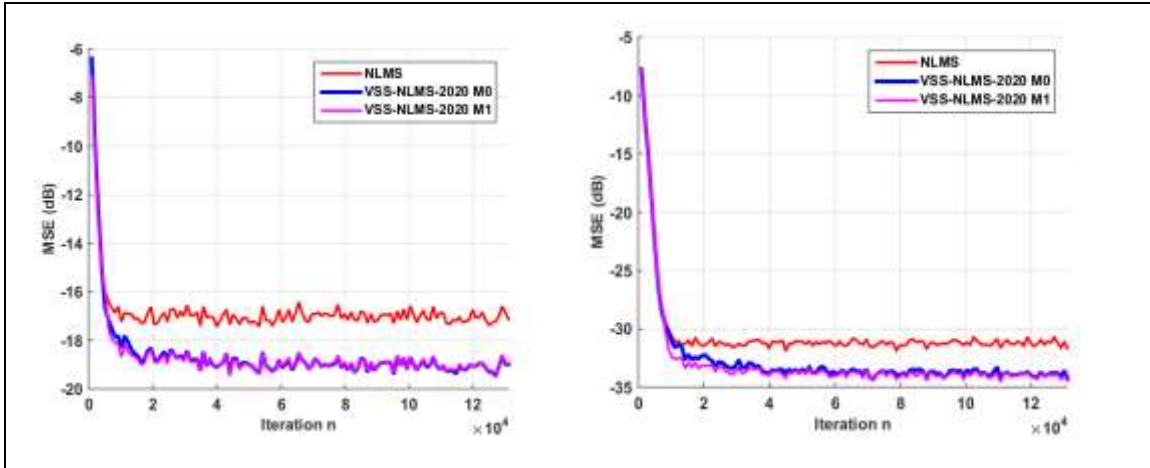
Pour obtenir les résultats suivants, on a utilisé ces valeurs des paramètres :

$$\varepsilon = 0.000001, \lambda = \lambda' = 0.9989, \hat{\sigma}_e^2(0) = 0.001, \hat{\sigma}_{a1}^2(0) = 0.01, \hat{\sigma}_{y1}^2(0) = 0.01.$$

Les mêmes séries de simulations ont été effectuées pour l'algorithme VSS-NLMS 2020. Une sélection de résultats est donnée sur les figures 3.38 à 3.44. Nous avons aussi

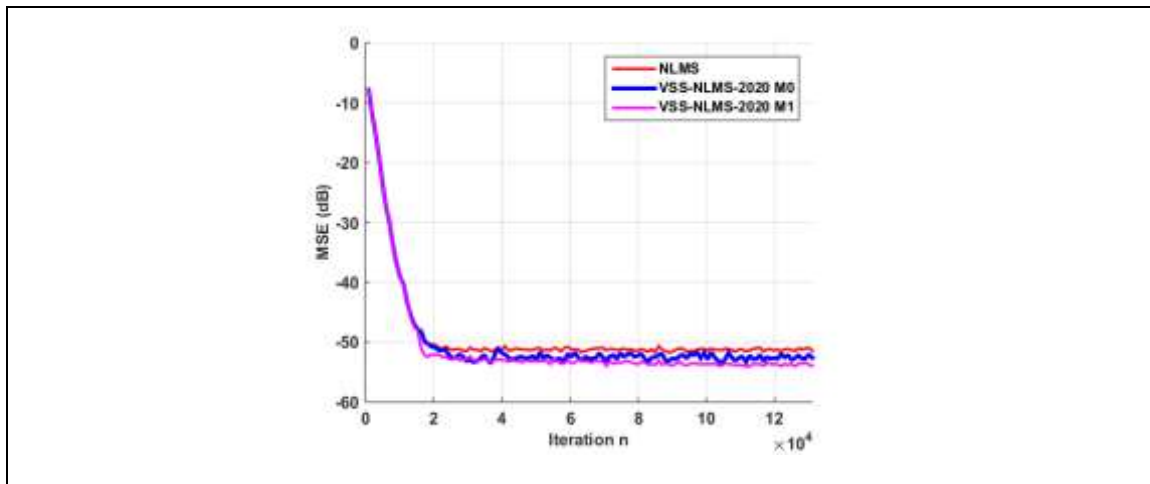
appliqué la proposition du la phase initial pour une durée de  $T_s$  pour les deux cas stationnaire et non stationnaire.

- Canal stationnaire :



**Figure 3.38:** signal USASI , cas stationnaire ,SNR=15dB,L=256,Ts=20\*L

**Figure 3.39 :**signal USASI , cas stationnaire ,SNR=30dB,L=256,Ts=30\*L



**Figure 3.40 :** signal USASI, cas stationnaire ,SNR=50dB,L=256,Ts=60\*L

**Commentaire :** on remarque une vitesse de convergence équivalente de cet algorithme VSS-NLMS 2020 avec la M1 par rapporte a M0, ainsi qu'il a un EQM finale devient moins important pour le cas moins bruite.

### Signal stationnaire

- Canal non stationnaire :



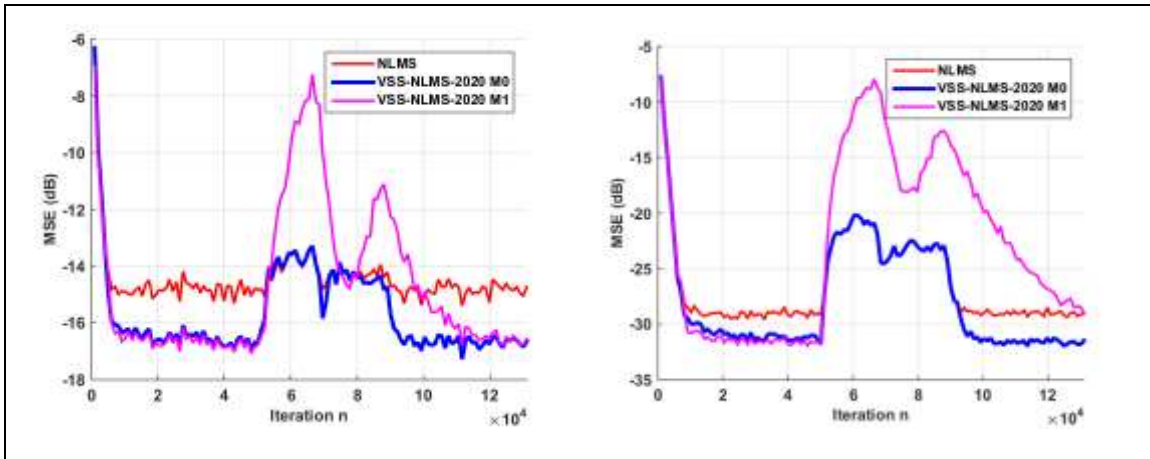


Figure 3.41 : signal USASI,

Figure 3.42 : signal USASI,

cas non stationnaire ,SNR=15dB,L=256 ,Ts=20\*L

cas non stationnaire ,SNR=30dB,L=256,Ts=30\*L

**Commentaire :** on note une incapacité de poursuite pour l’algorithme VSS-NLMS-2020 avec la méthode M1 après le passage de non stationnarité qui peut être amélioré lorsque on diminue  $\lambda$  mais va dégradé les propriétés de VSS , ainsi n’arrive pas à reconverger et reste perturber par une EQM importante jusqu'à la fin de la simulation.

### Signal parole

- canal stationnaire :

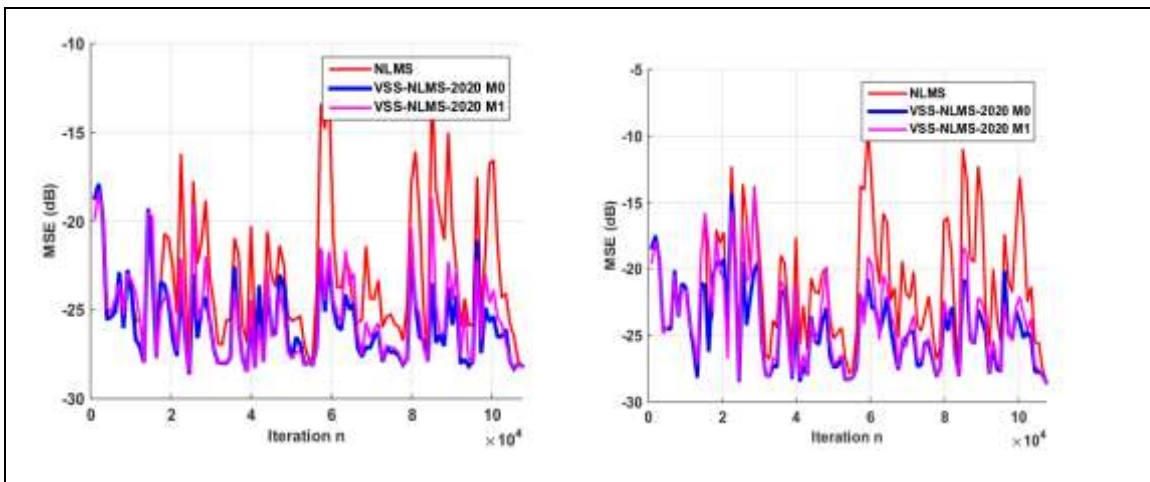


Figure 3.43 : signal parole

Figure 3.44: signal parole

cas stationnaire, SNR=15dB,L=256

cas stationnaire ,SNR=15dB,L=512

**Commentaire :** Pour le cas stationnaire, sachant que  $T_s > 0$  on a remarqué que le comportement de MSE de l’algorithme VSS-NLMS-2020 avec M1 est un peu meilleur que de VSS-NLMS PROP avec M0.

Enfin nous pouvons conclure concernant l’algorithme VSS-NLMS-2020 avec la méthode M1 et après avoir ajouté la période de convergence initiale (TS), que cet méthode n’est pas un candidat potentiel pour l’estimation de bruit avec c’est mauvaise performance

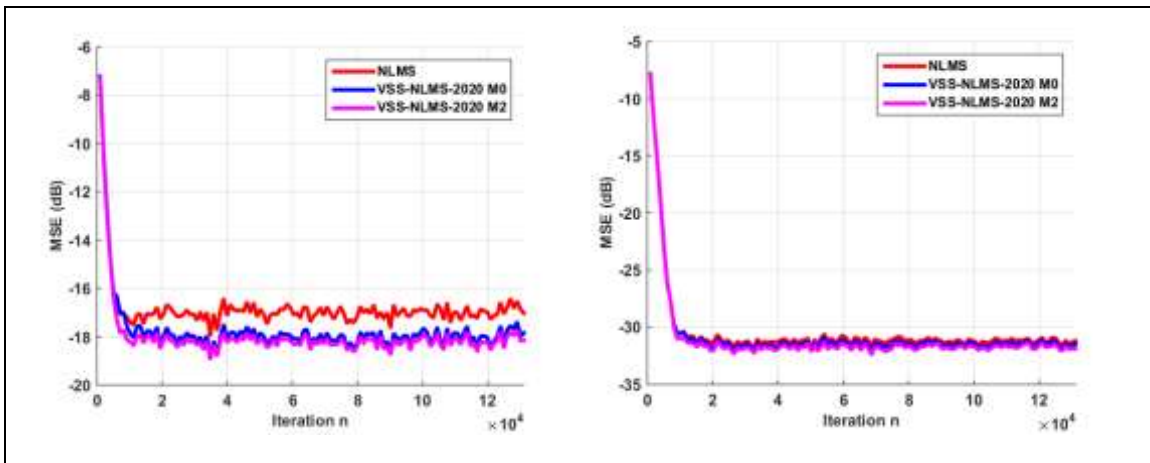
## L'algorithmme VSS-NLMS-2020 avec la méthode d'estimation M2

Pour obtenir les résultats suivants, on a utilisé les valeurs des paramètres :

$$\varepsilon = 0.000001, \delta = 20 * \sigma_x^2, \sigma_x^2 = 0.3272., \lambda = 0.83, \lambda' = 0.993, \hat{\sigma}_e^2(0) = 0.001, \hat{\sigma}_{ee}^2(0) = 0.01, \alpha = 0.1$$

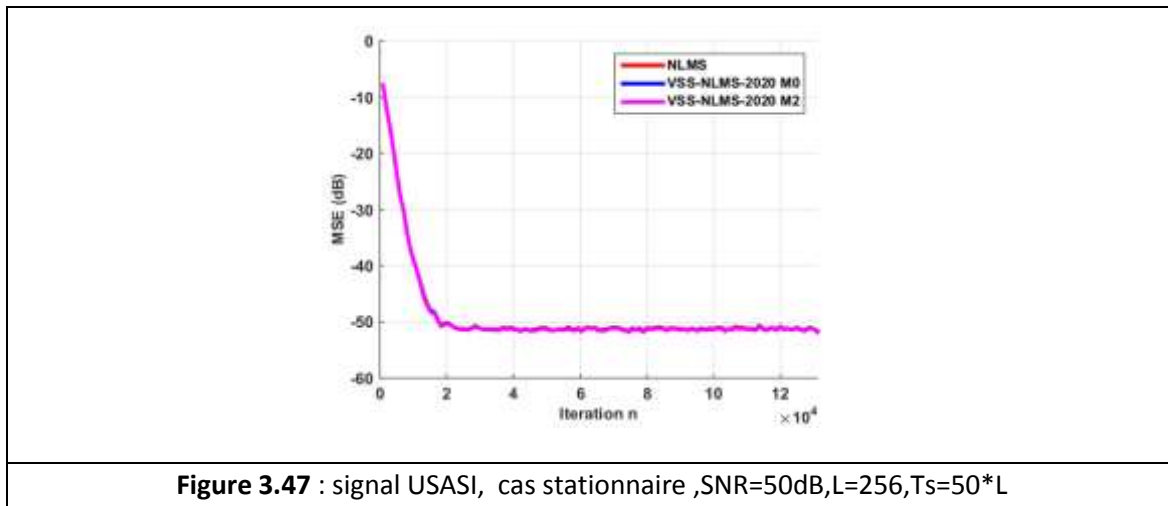
Quelques résultats de simulation de cet algorithmme avec la proposition Ts pour les deux cas stationnaire et non stationnaire sont résumés sur les figures 3.45 à 3.54

- **Canal stationnaire :**

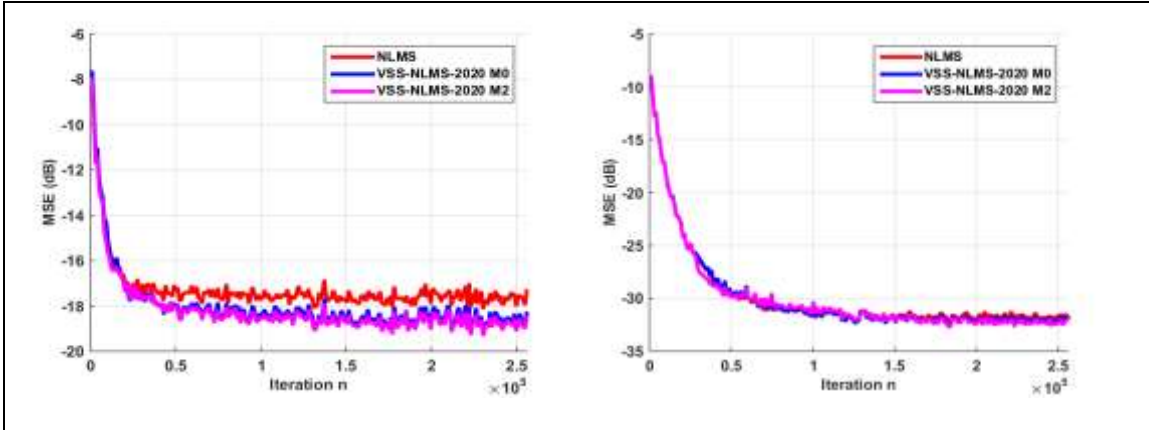


**Figure 3.45 :** signal USASI , cas stationnaire ,SNR=15dB,L=256,Ts=20\*L

**Figure 3.46 :** signal USASI , cas stationnaire ,SNR=30dB,L=256,Ts=30\*L



**Figure 3.47 :** signal USASI, cas stationnaire ,SNR=50dB,L=256,Ts=50\*L

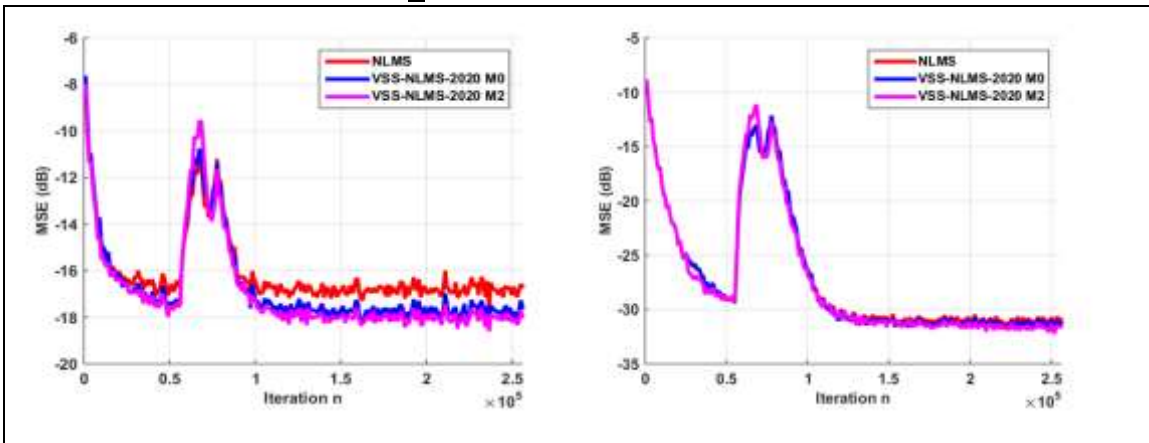


**Figure 3.48** : signal BBG1AR20, cas stationnaire ,SNR=15dB,L=256,Ts=30\*L

**Figure 3.49**:signal BBG1AR20 , cas stationnaire ,SNR=30dB,L=256,Ts=110\*L

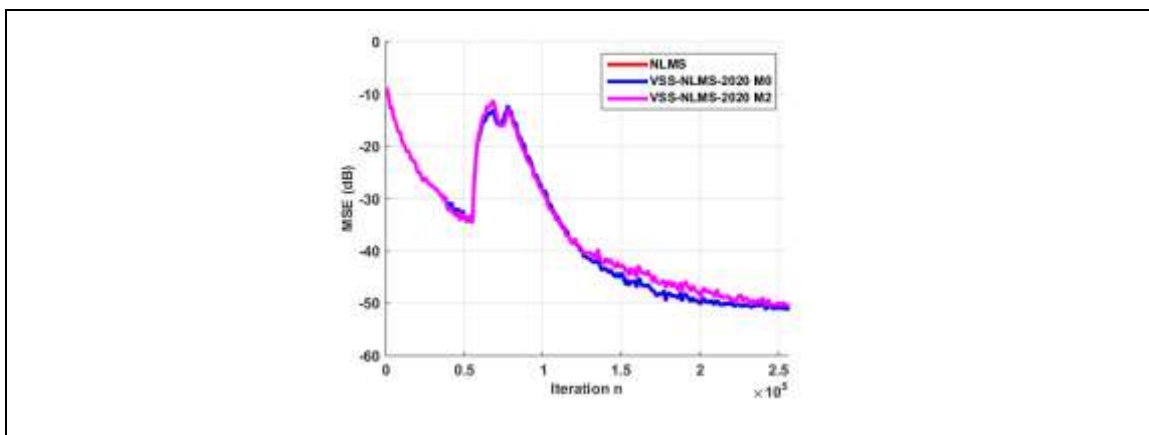
**Commentaire** : Comme il est visible sur les résultats de simulation l’algorithme VSS-NLMS 2020 avec M2 a une vitesse de convergence équivalente pour les deux type de signaux ,ainsi qu’il a un EQM finale moins importants que celle de M0 pour les deux type de signal .

**Canal non stationnaire :**



**Figure 3.50** : signal BBG1AR20 , cas non stationnaire ,SNR=15dB,L=256,Ts=30\*L

**Figure 3.51** : signal BBG1AR20 , cas non stationnaire ,SNR=30dB,L=256,Ts=100\*L

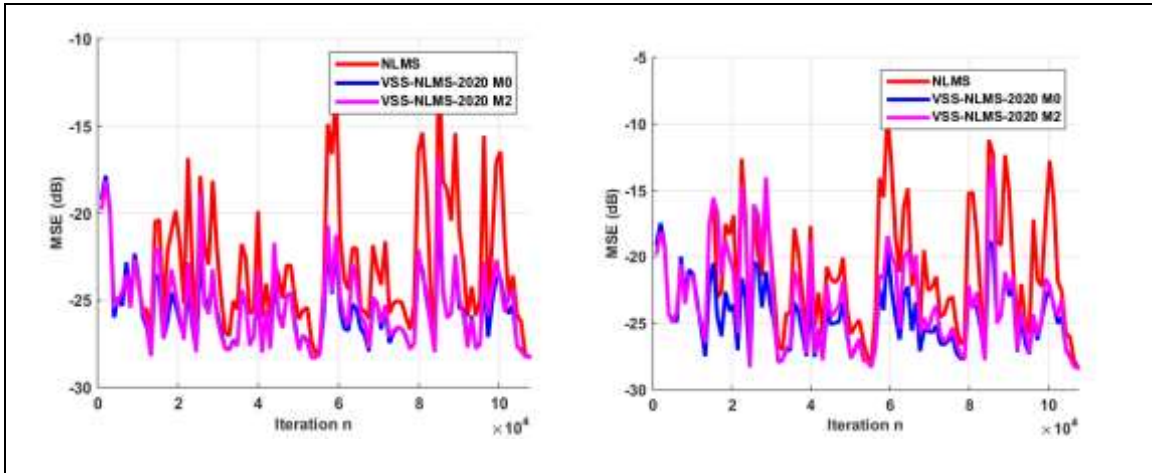


**Figure 3.52** : signal BBG1AR20, cas non stationnaire, SNR=50dB, L=256, Ts=110\*L

**Commentaire :** le cas intéressant est que la capacité de poursuite du VSS-NLMS 2020 M2 est équivalente à celle du M0 et il arrive à re-converger même rapidement après le passage de la non stationnarité, ainsi qu'il arrive à reconverger vers une EQM finale moins importante que de M0 pour un SNR 15dB et 30dB.

**Signal parole**

- **canal stationnaire :**



**Figure 3.53 :** signal parole

cas stationnaire, SNR=15dB, L=256

**Figure 3.54 :** signal parole

cas stationnaire, SNR=15dB, L=512

**Commentaire :** Pour le cas stationnaire, sachant que  $T_s > 0$  on a remarqué que le comportement de MSE de l’algorithme VSS-NLMS-2020 avec M2 est un peu meilleur que de VSS-NLMS-2020 avec M0

D’après résultats obtenus dans les simulations de l’algorithme VSS-NLMS-2020 avec la méthode M2 nous pouvons conclure ; au vu de ces bonnes performances, cette méthode peut être candidat potentiel pour l’estimation de bruit dans l’identification d’écho en termes de la rapidité de la vitesse de convergence et la poursuite et l’amélioration de l’erreur finale.

**L’algorithme VSS-NLMS 2020 avec la méthode d’estimation M3**

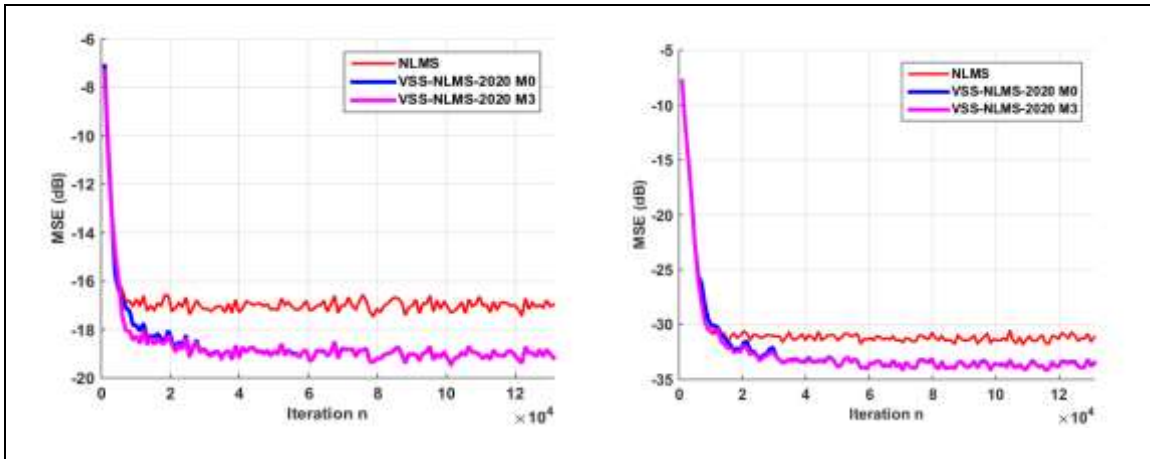
Pour obtenir les résultats suivants, on a utilisé ces valeurs des paramètres :

$$\delta = 0.000001, \lambda' = 0.9992, \lambda = 0.9961, \hat{\sigma}_{ee}^2(0) = 0.01, \hat{\sigma}_x^2(0) = 0.01$$

$$c_0 = 0.001, R_{xe}(0) = 0.$$

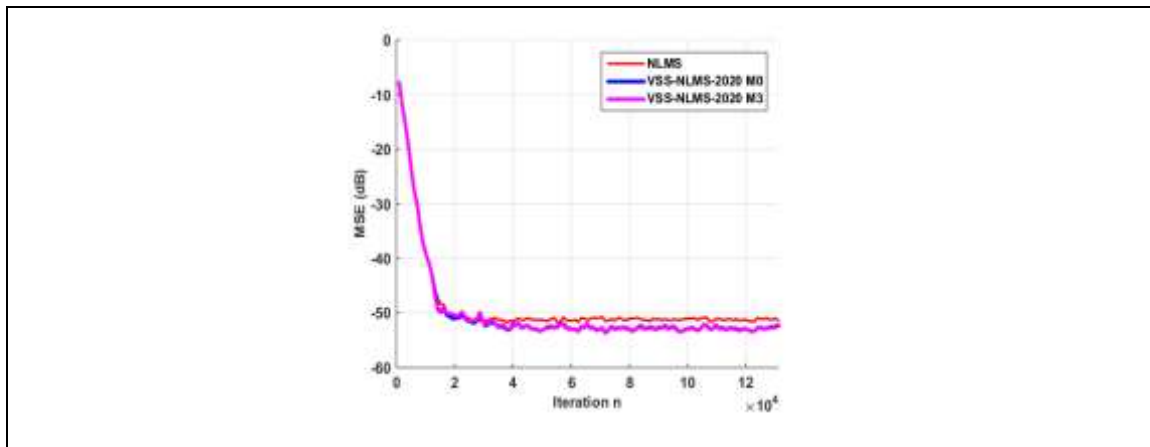
Quelques résultats de simulation de l’algorithme VSS-NLMS-2020 avec la proposition précédent pour les deux cas stationnaire et non stationnaire sont résumés sur les figures 3.55 à 3.63.

- **Canal stationnaire :**

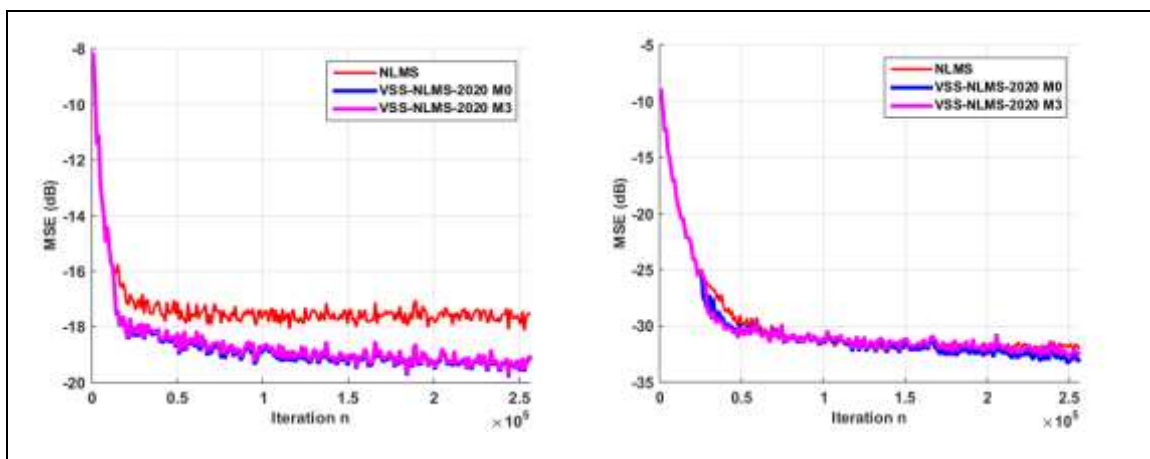


**Figure 3.55** : signal USASI ,  
cas stationnaire ,SNR=15dB,L=256,Ts=20\*L

**Figure 3.56** :signal USASI ,  
cas stationnaire ,SNR=30dB,L=256,Ts=25\*L



**Figure 3.57** : signal USASI , cas stationnaire ,SNR=50dB,L=256,Ts=50\*L



**Figure 3.58** : signal BBG1AR20,  
cas stationnaire, SNR=15dB, L=256, Ts=50\*L

**Figure 3.59**:signal BBG1AR20 ,  
cas stationnaire ,SNR=30dB,L=256,Ts=110\*L

**Commentaire** : Pour le cas stationnaire de l’algorithme VSS-NLMS-2020 avec la méthode 3 , on observe une vitesse de convergence équivalente pour les deux types de signaux ,avec un erreurs finale identique que de M0 pour les deux type de signal .

## Signal stationnaire

- Canal non stationnaire:

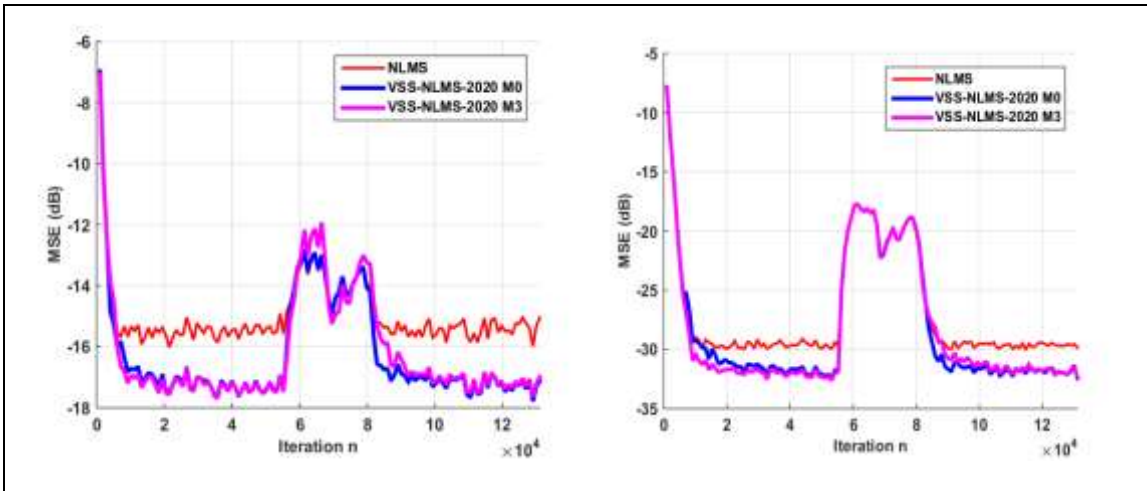


Figure 3.60 : signal USASI,

Figure 3.61 : signal USASI,

cas non stationnaire ,SNR=15dB,L=256 ,Ts=20\*L

cas non stationnaire ,SNR=30dB,L=256,Ts=30\*L

**Commentaire :** qui est intéressant de noter l'algorithme VSS-NLMS-2020 avec M3 est que sa capacité de poursuite est équivalente a celle du M0 et il arrive à re-converger vers une EQM identique après le passage de la non stationnarité du canal acoustique.

## Signal parole

- canal stationnaire:

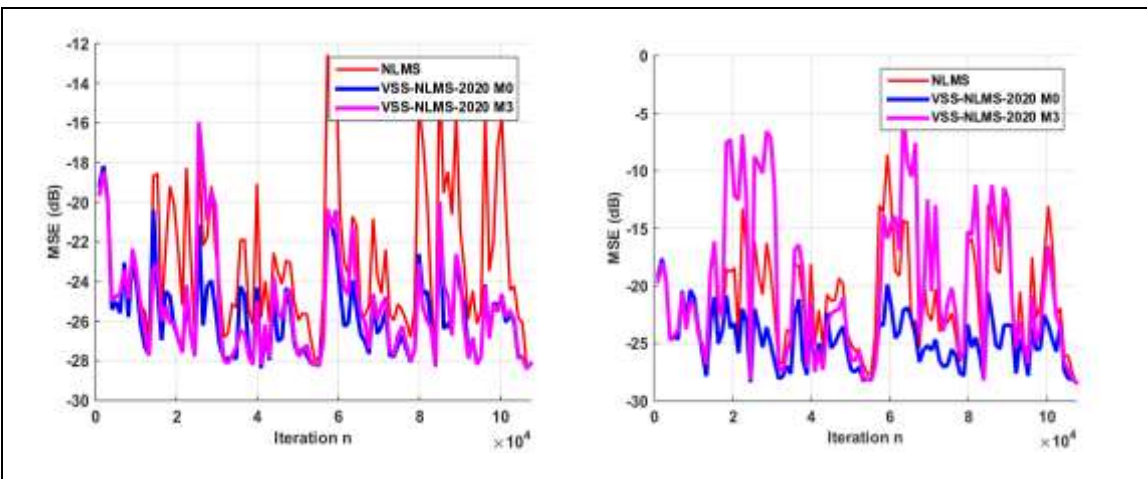


Figure 3.62 : signal parole

Figure 3.63 : signal parole

cas stationnaire, SNR=15dB,L=256

cas stationnaire ,SNR=15dB,L=512

**Commentaire :** Pour le cas stationnaire, sachant que  $T_s > 0$  on a remarqué que le comportement de MSE de l'algorithme VSS-NLMS-2020 avec M3 est meilleur que de VSS-NLMS PROP avec M0.

Par conséquent, cet algorithme est un candidat intéressant vu de ces bonnes performances, cette méthode peut être candidat potentiel pour l'estimation de bruit dans l'identification d'écho en termes de la rapidité de la vitesse de convergence et l'EQM finale.

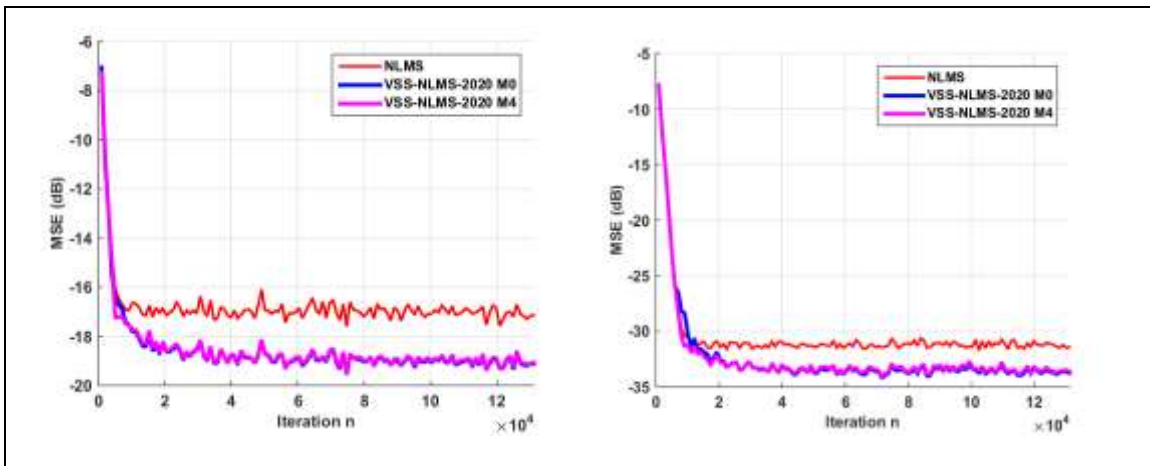
### L'algorithme VSS-NLMS 2020 avec la méthode d'estimation M4

Pour obtenir les résultats suivants, on a utilisé ces valeurs des paramètres :

$$\varepsilon = 0.000001, \quad \lambda = 0.9961, \lambda' = 0.9992, \quad \hat{\sigma}_e^2(0) = 0.001, \quad \hat{\sigma}_{ee}^2(0) = 0.01, \delta = 0.00001,$$

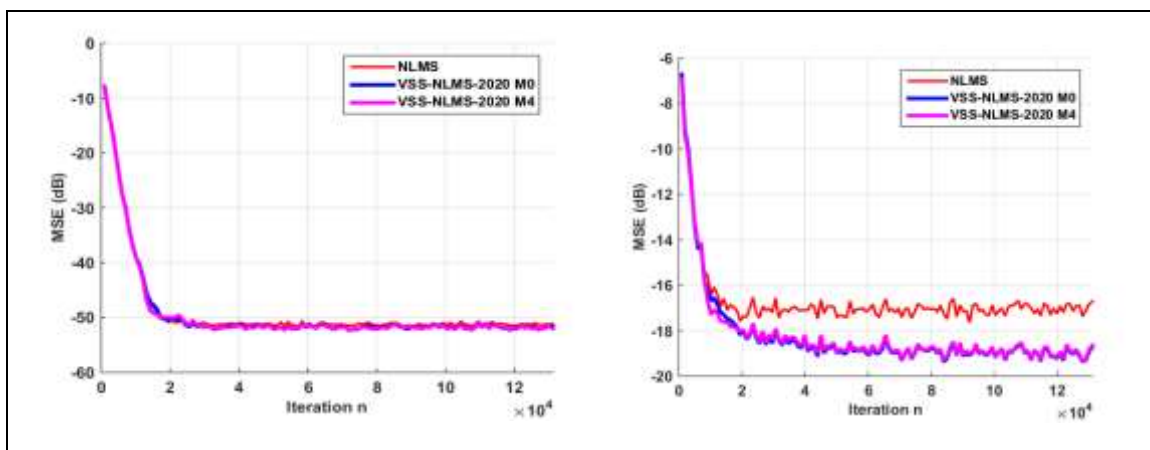
Les mêmes séries de simulations ont été effectuées avec la proposition de la dure initiale  $T_s$ . Les résultats est résumé sur les figure 3.64 à 3.72

#### Canal stationnaire :



**Figure 3.64** : signal USASI ,  
cas stationnaire ,SNR=15dB,L=256,Ts=15\*L

**Figure 3.65** :signal USASI ,  
cas stationnaire ,SNR=30dB,L=256,Ts=25\*L

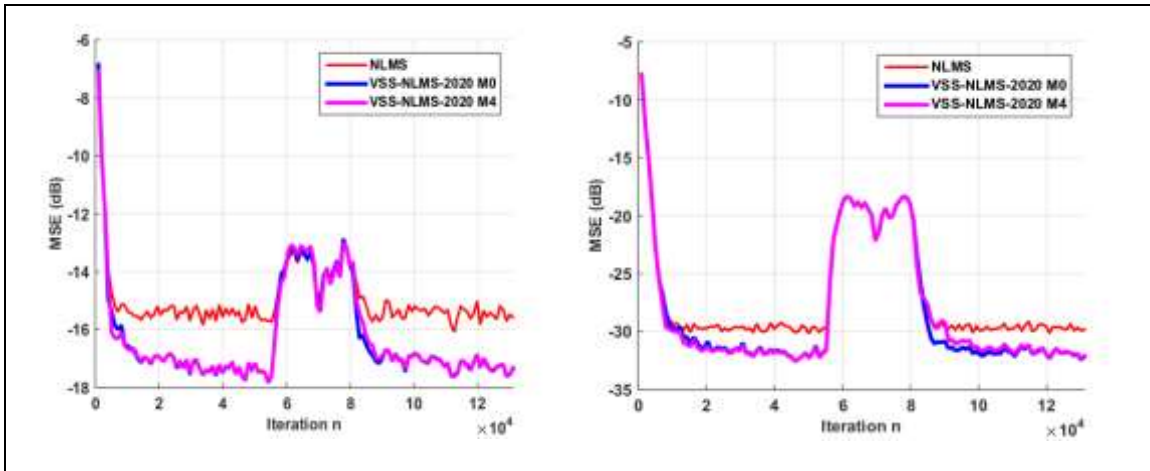


**Figure 3.66** : signal USASI ,  
cas stationnaire ,SNR=50dB,L=256,Ts=50\*L

**Figure 3.67** :signal USASI ,  
cas stationnaire ,SNR=15dB,L=512,Ts=15\*L

**Commentaire :** On voit que l’algorithme VSS-NLMS 2020 M4 réalise a des performances identique que M0 en terme de la vitesse de convergence et il fournit un EQM finale équivalent que la méthode théorique pour tous les SNR.

**Canal non stationnaire :**

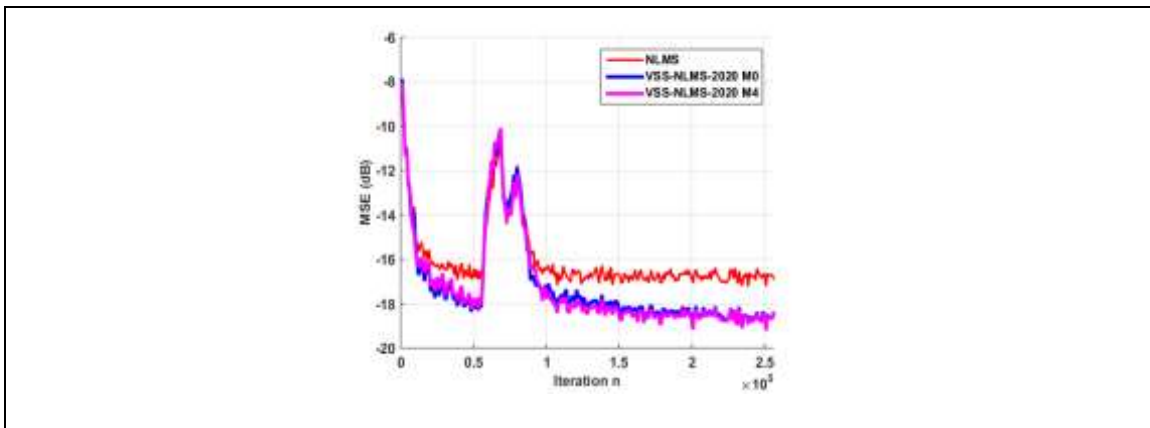


**Figure 3.68:** signal USASI,

**Figure 3.69 :** signal USASI,

cas non stationnaire ,SNR=15dB,L=256 ,Ts=20\*L

cas non stationnaire ,SNR=30dB,L=256,Ts=25\*L



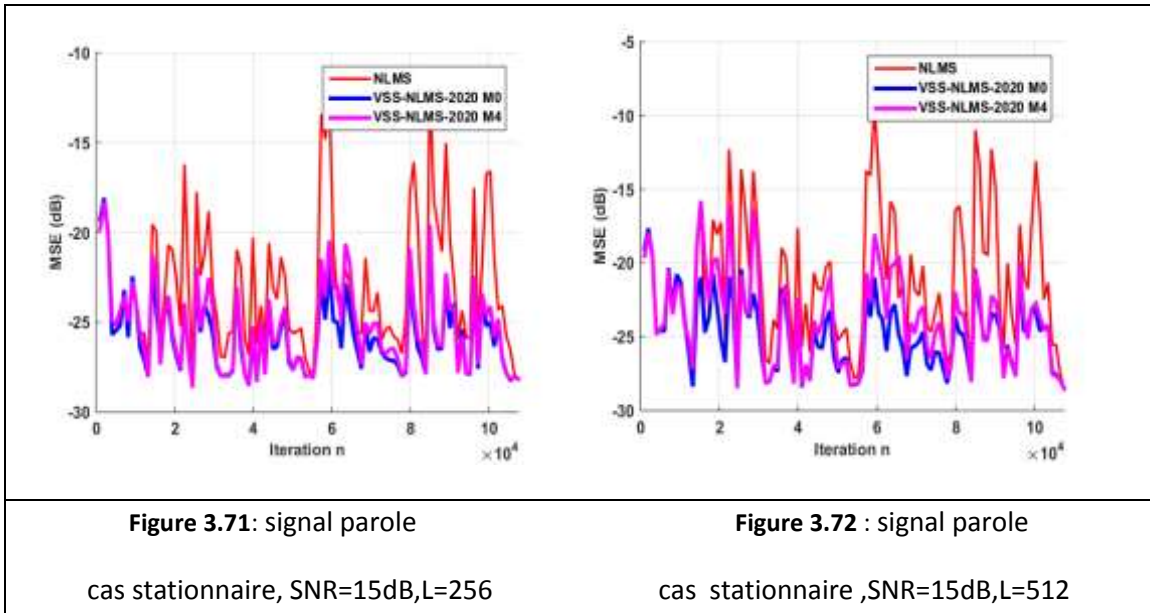
**Figure 3.70 :** signal BBG1AR20, cas non stationnaire ,SNR=15dB,L=256,Ts=30\*L

**Commentaire :** Ces résultats montrent que l’algorithme VSS-NLMS 2020 avec M4 possède un bon comportement en poursuite des non stationnarités du canal acoustique. Il arrive re-convergence vers un EQM finale identique que de celle de la M0.

**Signal parole :**

- canal stationnaire :





**Commentaire :** Pour le cas stationnaire, sachant que  $T_s > 0$  on a remarqué que le comportement de MSE de l’algorithme VSS-NLMS-2020 avec M4 est meilleur que de VSS-NLMS PROP avec M0.

On conclut après résultats obtenus avec l’algorithme VSS-NLMS-2020 avec M4 ,d’où c’est bonne performances, cette méthode peut être candidat potentiel pour l’estimation de bruit dans l’identification d’écho.

### Tableau NP-VSS-NLMS avec les 4 méthodes d’estimation

Dans cette partie nous allons faire une comparaison entre quatre méthodes d’estimations de la variance du bruit additif avec la méthode théorique M0 **Le tableau 3.1** représente les différents paramètres qui comparent l’EQM finale des 4 méthodes pour l’algorithme NP-VSS-NLMS par rapport à la valeur donnée par la formule théorique M0 dans le cas stationnaire. Pour le signal USASI l’estimateur M1 est le plus proche de M0 pour les cas SNR=15, 30 et 50 dB pour L=256. Pour le signal BBG1AR20 et signal parole l’estimateur M2 est le plus proche de M0 pour le cas bruit é SNR=15 dB et se dégrade pour SNR plus important. Pour La méthode M4 est la plus stable pour le cas bruité SNR=15 dB pour les trois types de signaux USASI, BBG1AR20 et signal parole.

signaux	L	SNR	M0	M1/M0	M2/M0	M3/M0	M4/M0
signal USASI	256	15	0.0119	1.3316	3.1386	3.1512	1.7134
		30	3.7e-04	0.6606	1.2455	4.2462	1.4374
		50	3.7e-06	0.0868	0.9837	1.4230	0.3990
	512	15	0.0122	1.6150	1.4084	1.4034	1.7168
		30	3.7e-04	0.8200	3.1455	6.0151	1.2658
		50	3.7e-06	0.1034	1.4557	0.2631	0.2363
signal BBG1AR20	256	15	0.0103	1.6788	1.0717	1.4436	1.6128
		30	3.2e-04	6.0643	0.7372	3.2655	1.9532
		50	3.2e-06	2.9358	4.5432	1.9982	2.4605
	512	15	0.0105	1.6810	1.6339	2.1490	1.6225
		30	3.2e-04	0.8564	3.5026	1.0115	2.0582
		50	3.2e-06	0.8032	2.5380	1.7117	2.0529
signal parole	256	15	0.0015	1.2106	0.9165	1.6322	1.7861
		30	4.8e-05	3.8608	1.5267	4.0038	0.9336
		50	4.8e-07	90.5989	4.7235	-6.9256	0.1177
	512	15	0.0018	1.3148	0.9499	2.3847	1.7558
		30	4.8e-05	1.9785	0.7314	2.1259	0.8400
		50	4.8e-07	86.1702	18.6391	-6.6439	0.2065

**Tableau 3.1.** Comparaisons des 4 méthode d'estimation avec M0 pour NP-VSS-NLMS

### **Tableau VSS-NLMS-2020 avec les 4 méthodes d'estimation**

Dans cette partie nous allons faire une comparaison entre quatre méthodes d'estimations de la variance du bruit additif avec la méthode théorique M0 Le **tableau 3.2** représente les différents paramètres qui comparent l'EQM finale des 4 méthodes pour l'algorithme VSS-NLMS-2020 par rapport à la valeur donnée par la

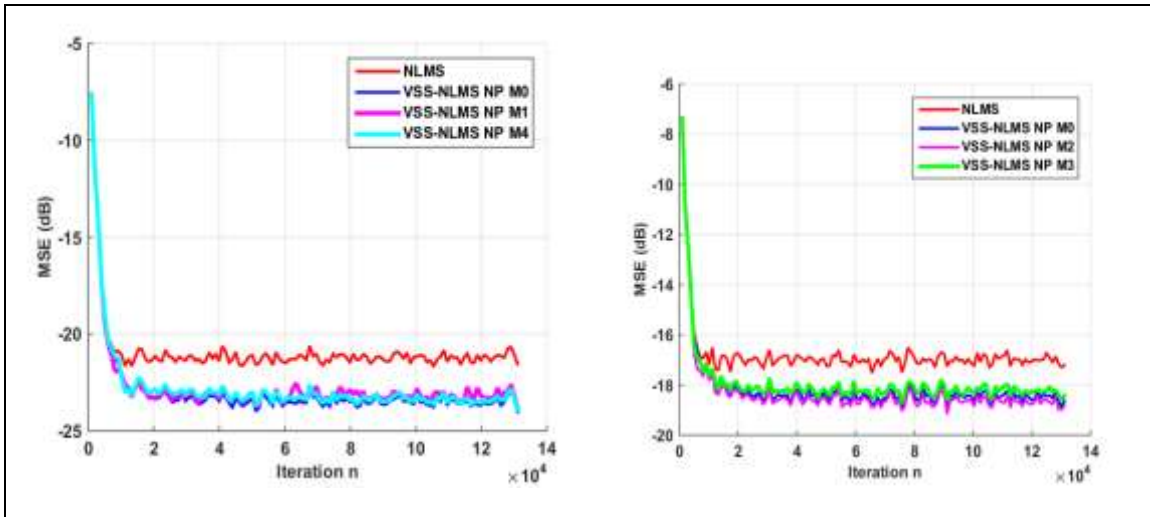
formule théorique M0 dans le cas stationnaire. Pour le signal USASI l'estimateur M3 est le plus proche de M0 pour les cas SNR=15 pour L=256. Pour le signal BBG1AR20 et signal parole l'estimateur M2 est le plus proche de M0 pour le cas bruit é SNR=15 dB pour L=256 et L=512. Pour La méthode M4 est la plus stable pour tous les SNR testés pour les deux types de signaux USASI, BBG1AR20.

signaux	L	SNR	M0	M1/M0	M2/M0	M3/M0	M4/M0
signal USASI	256	15	0.0119	1.3556	1.1520	1.0731	1.5894
		30	3.7e-04	58.7851	1.0481	1.0965	1.5919
		50	3.7e-06	0.0668	1.1043	1.0436	1.6798
	512	15	0.0119	0.6671	0.8598	1.2037	1.6897
		30	3.7e-04	0.6544	0.7883	1.2266	1.5186
		50	3.7e-06	0.0718	1.2157	1.0927	1.6604
signal BBG1AR20	256	15	0.0103	0.9984	0.9277	1.0756	1.7115
		30	3.2e-04	0.6214	1.1199	1.2453	1.9797
		50	3.2e-6	0.4817	9.1797	1.2398	1.9339
	512	15	0.0105	1.2063	1.0278	1.1991	1.7397
		30	3.2e-04	0.8907	1.5409	1.3008	1.9118
		50	3.2e-06	0.9203	3.2088	1.2500	1.9668
sigal parole	256	15	0.0015	2.1610	1.8058	1.9470	3.4465
		30	4.8e-05	1.6903	2.0595	1.6360	2.5148
		50	4.8e-07	6.5560	1.6469	3.7794	2.9135
	512	15	0.0020	3.3599	3.3139	10.3965	5.3681
		30	4.8e-05	2.9707	2.8916	-0.4602	3.2106
		50	4.8e-07	56.3289	3.3273	9.9442	5.6836

**Tableau 3.2.** Comparassions des 4 méthode d'estimation avec M0 pour VSS-NLMS-2020

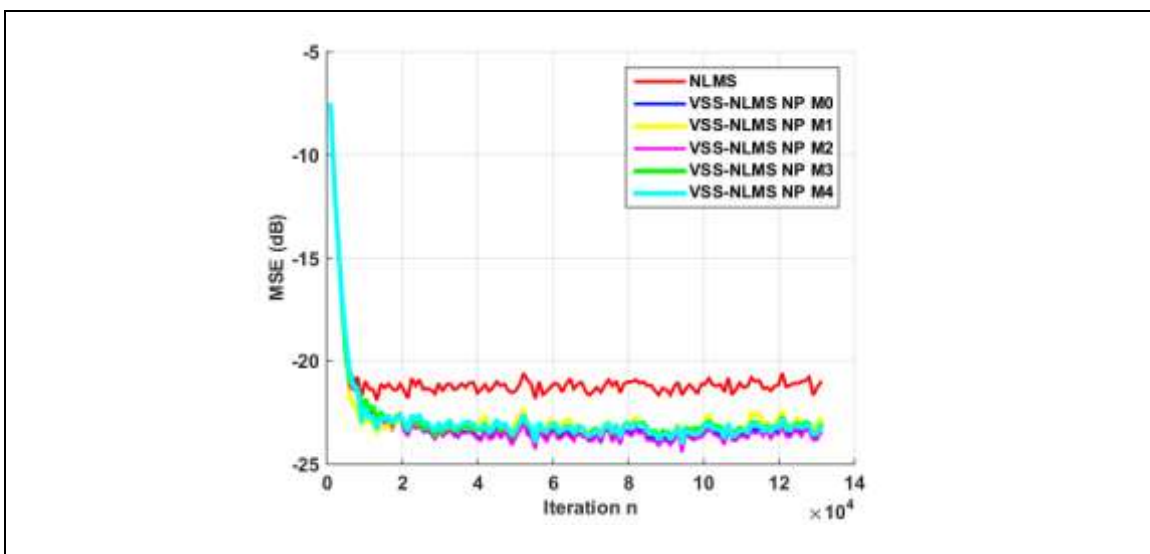
### 3.5 Comparaison entre les 4 méthodes d'estimation pour chaque VSS-NLMS (NP-VSS-NLMS/ VSS-NLMS-2020)

Dans cette partie de simulation, nous avons fixé la taille du filtre  $L=256$  et le  $SNR=15$  pour un signal USASI de type de canal stationnaire, pour faire une comparaison entre les méthodes d'estimation de variance de bruit avec chaque un d'algorithme NP-VSS-NLMS et l'algorithme VSS-NLMS-2020. Nous avons obtenu les résultats présentés sur les figures suivantes :



**Figure 3.73** : signal USASI,  
cas stationnaire ,SNR=15dB,L=256

**Figure 3.74**: signal USASI,  
cas stationnaire ,SNR=15dB,L=256

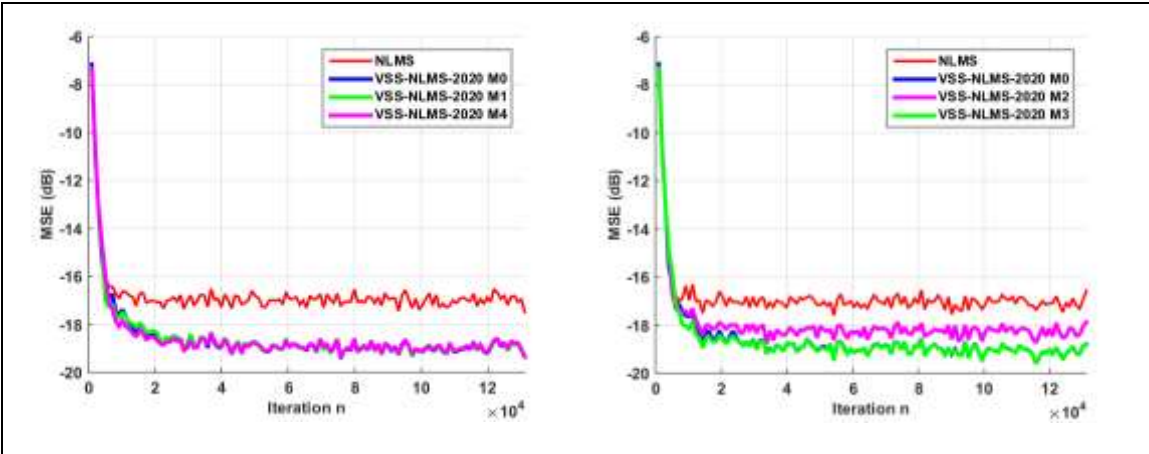


**Figure 3.75** : signal USASI, cas stationnaire ,SNR=15dB,L=256

**Commentaire** : D'après les résultats obtenus dans la figure (3.73) on remarque que les deux méthodes M1 et M4 sont équivalentes en termes de vitesse de convergence, par

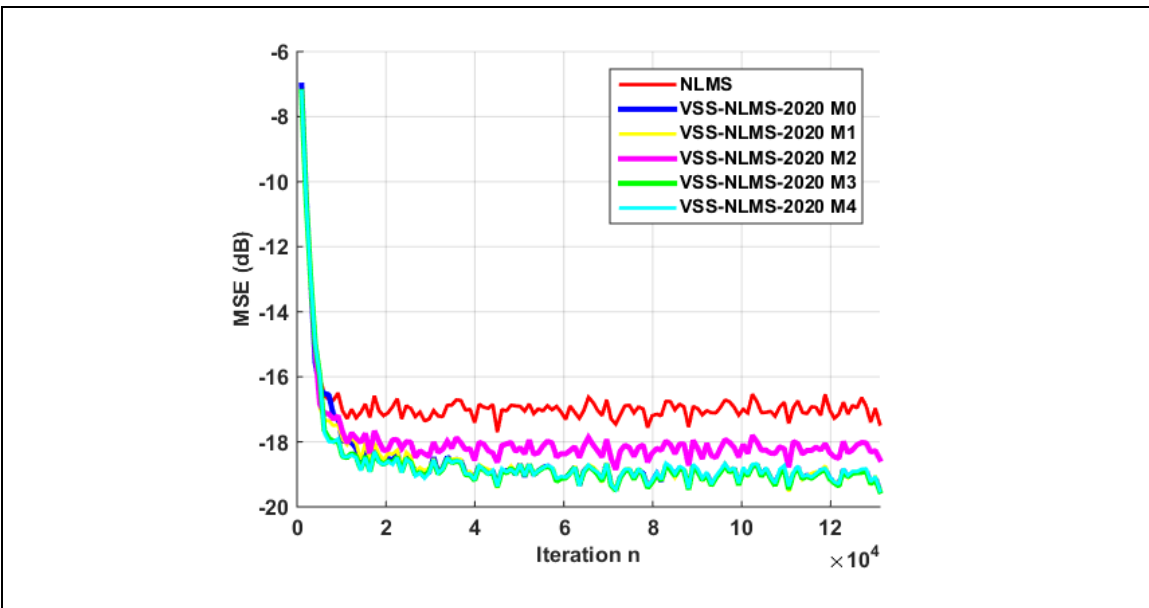
contre la meilleure performance d'erreur et la plus proche de M0 c'est la M4. Pour la figure (3.74) on observe que la M2 donne la meilleure performance que M3 par rapport à la méthode théorique

Enfin, nous pouvons conclure d'après la figure (3.75) que la meilleure méthode applique pour l'algorithme NP-VSS-NLMS c'est la M4 pour l'amélioration de la vitesse de convergence et la M2 pour un EQM finale faible.



**Figure 3.76 :** signal USASI,  
cas stationnaire ,SNR=15dB,L=256

**Figure 3.77:** signal USASI,  
cas stationnaire ,SNR=15dB,L=256



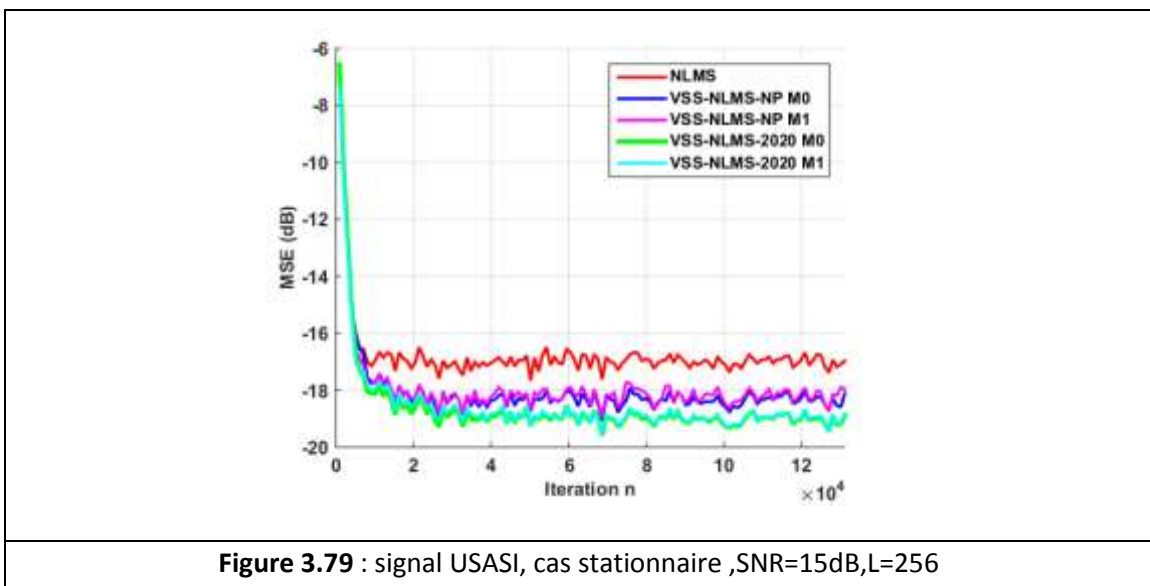
**Figure 3.78 :** signal USASI , cas stationnaire ,SNR=15dB,L=256

**Commentaire :** D'après les résultats obtenus dans la figure (3.76) on remarque que la méthode la plus proche de M0 est M4 d'où la vitesse de convergence et l'EQM finale sont identique que de méthode théorique. Pour la figure (3.77) on observe que la M3 donne la meilleure performance que M2 par rapport à la méthode théorique

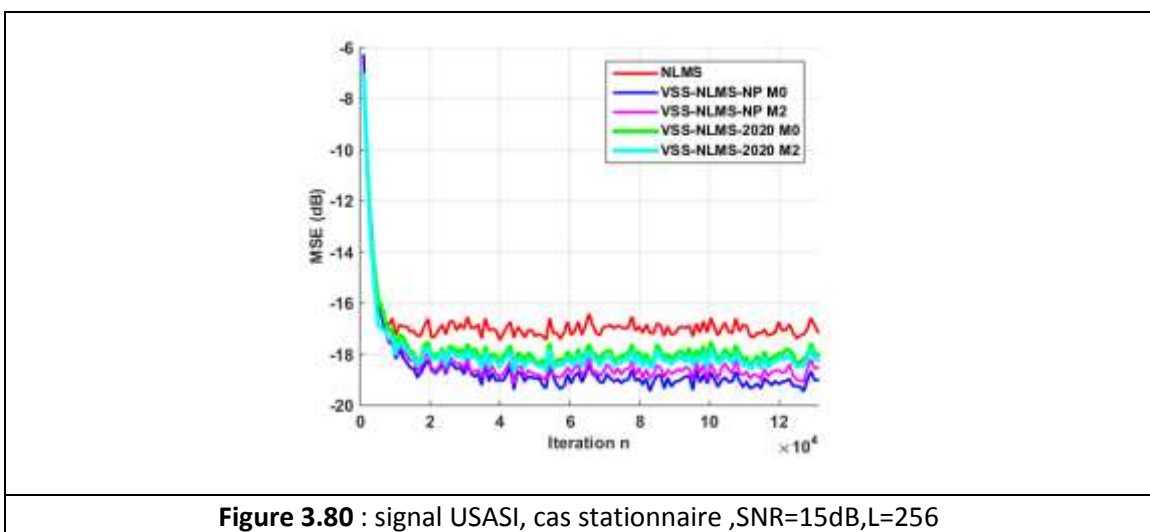
Enfin, nous pouvons conclure d'après la figure (3.78) que la meilleure méthode appliquée pour l'algorithme VSS-NLMS-2020 est celle qui donne les meilleures performances c'est la M4 et la M3.

### 3.6 Comparaison entre deux algorithmes NP-VSS-NLMS et VSS-NLMS-2020 avec chaque méthode d'estimation

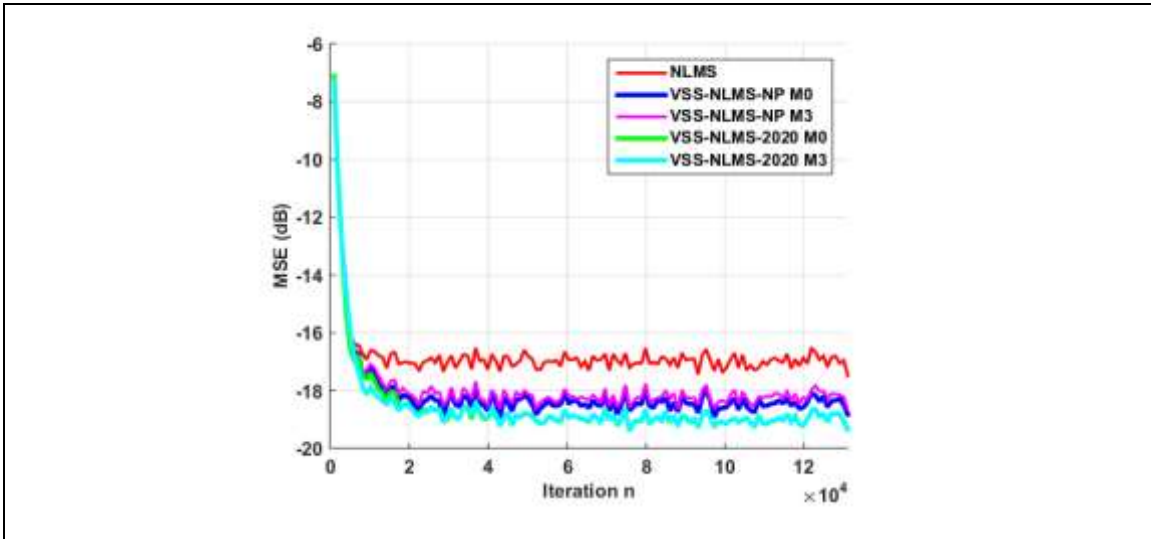
Dans cette partie de simulation, nous avons fixé la taille du filtre  $L=256$  et le  $SNR=15$  pour un signal USASI de type de canal stationnaire, pour faire une comparaison entre l'algorithme NP-VSS-NLMS et l'algorithme VSS-NLMS-2020 appliqué par chaque méthode d'estimation de variance de bruit (1 a4). Nous avons obtenu les résultats présentés sur les figures suivantes :



**Commentaire** : Nous remarquons que l'algorithme VSS-NLMS-2020 avec la M1 a une vitesse de convergence meilleur que celle d'algorithme NP-VSS-NLMS, ainsi qu'il a un EQM finale identique que de M0.

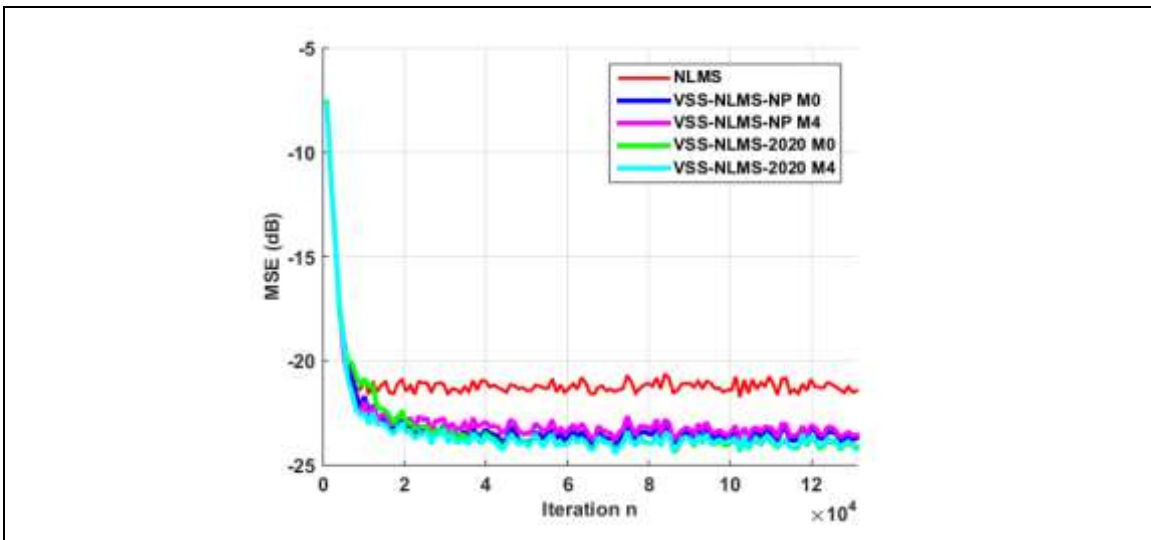


**Commentaire :** Nous observons que l’algorithme NP-VSS-NLMS avec la M2 a une vitesse de convergence meilleur que celle d’algorithme VSS-NLMS-2020, par contre l’algorithme VSS-NLMS-2020 a un EQM finale moins important et meilleur que d’algorithme NP-VSS-NLMS.



**Figure 3.81:** signal USASI, cas stationnaire ,SNR=15dB,L=256

**Commentaire :** D’après les résultats on observe que l’algorithme VSS-NLMS-2020 avec la M3 a une vitesse de convergence meilleur que celle d’algorithme NP-VSS-NLMS, ainsi qu’il a un EQM finale équivalent que de M0 et meilleur que d’algorithme NP-VSS-NLMS.



**Figure3.82:** signal USASI, cas stationnaire, SNR=15dB,L=256

**Commentaire :** Ces résultats montrent que l'algorithme VSS-NLMS-2020 avec la M4 qu'il a une amélioration de vitesse de convergence par rapport que M0 ainsi que les deux algorithmes sont similaires, d'autre part on remarque que VSS-NLMS-2020 a un l'EQM finale faible que de NP-VSS-NLMS.

Nous allons conclure d'après les résultats obtenus que l'algorithme VSS-NLMS-2020 à la meilleure performance en termes de vitesse de convergence et l'EQM finale.

### **3.7 Conclusion :**

Dans ce chapitre, nous avons présenté les résultats de simulation de l'application des deux algorithmes NP-VSS-NLMS et VSS-NLMS-2020 avec les méthodes d'estimation de la variance de bruit pour l'annulation d'écho acoustique (AEA) et qui sont implémenté sur Matlab avec les différents signaux (USASI, BBG1AR20, parole) .

A l'issu de ces résultats, de l'algorithme VSS-NLMS-2020 ont montré les performances recherchées en termes de la rapidité de la vitesse de convergence et la capacité de poursuite et l'EQM finale en régime permanent, concernant les méthodes d'estimation appliqué. Aussi, nous avons conclu que :

La méthode 1 est la meilleure méthode pour VSS-NLMS-NP

La méthode 4 est la plus stable et fiable pour les deux VSS-NLMS testés.



## Conclusion générale

---

L'écho acoustique et le bruit ambiant qui dégradent la qualité et intelligibilité de communication, et posent des problèmes très gênant et très difficiles à résoudre.

L'étude que nous avons présentée dans ce mémoire concerne l'annulation d'écho acoustique par les algorithmes de filtrage adaptatif.

Dans un premier temps nous avons étudié théoriquement quelques algorithmes à pas variables (VSS-NLMS) dans le domaine de l'annulation d'écho acoustique, ensuite nous avons implémenté les algorithmes à pas variable VSS\_NLMS avec les méthodes d'estimation de bruit sous Matlab pour bien évaluer les performances. Le pas variable VSS est utilisé pour équilibrer le conflit entre la vitesse de convergence et l'erreur en régime permanent. L'idée est d'utiliser une grande valeur du pas dans les premières itérations de la convergence de l'algorithme pour avoir une bonne vitesse de convergence, et une petite valeur du pas en régime permanent, après la convergence, pour atteindre une puissance d'erreur finale assez faible.

Parmi ces algorithmes VSS-NLMS, nous avons essayé de voir quels sont les algorithmes VSS avec les méthodes d'estimation de bruit qui ont des meilleures performances (une capacité de poursuite des non stationnarités du canal intéressante et l'EQM final, et la rapidité de la vitesse de convergence) et pour définir une meilleure méthode d'estimation, Cette dernière propriété est une performance très recherchée dans les applications d'annulation d'écho acoustique.

A l'issue de ces résultats, de l'algorithme VSS-NLMS-2020 ont montré les performances recherchées en termes de la rapidité de la vitesse de convergence et la capacité de poursuite et l'EQM finale en régime permanent, concernant les méthodes d'estimation appliqué. Aussi, nous avons conclu que :

La méthode 1 est la meilleure méthode pour VSS-NLMS-NP

La méthode 4 est la plus stable et fiable pour les deux VSS-NLMS testés.

# Bibliographie

---

- [1] FERRANE Khaled (Filtrage adaptatif pour l'annulation d'écho acoustique dans les communications mains libres) université de Bouira Mémoire de Master.2017
- [2] Ghanem C & Fedlaoui L . (Algorithmes adaptatifs à pas variables pour la poursuite d'un canal acoustique non stationnaire) université Saad Dahleb de Blida Mémoire de Master 2019
- [3] DJENDI Mohamed (étude et implémentation d'un filtre RIF adaptatif sous VHDL) université Saad dahleb de Blida Mémoire de Master 2017
- [4] BENESTY jacob (traitement des signaux numériques \_Filtrage adaptatif et analyse spectrale).  
<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.726.67&rep=rep1&type=pdf>
- [5] Jacob Benesty' A Nonparametric VSS NLMS Algorithm', iee signal processing letters, VOL. 13, NO. 10, OCTOBER 2006.
- [6] Paleologu, Jacob Benesty, and Silviu Ciochină I. Cohen et al: 'Variable Step-Size Adaptive Filters for Echo Cancellation ' Speech Processing in Modern Communication, Constantin 2010.
- [7] Paleologu, Jacob Benesty, Silviu Ciochină 'a family of variable step-size NLMS algorithms for echo cancellation', Électrotechn. et Énerg. Constantin, 55, 1, p. 49–58, Bucarest, 2010
- [8] MA Iqbal, SL Grant, in Proc.,. 'Novel variable step size NLMS algorithms for echo cancellation', 'Speech and Signal Processing (ICASSP) ,(Las Vegas,NV, 2008). pp. 241-244.
- [9] Hsu-Chang Huang and Junghsi Lee "A New Variable Step-Size NLMS Algorithm and Its Performance Analysis "IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 60, NO. 4, APRIL 2012.
- [10] Bouzertini & Serea ' Algorithme NLMS optimisé pour l'identification de réponses impulsionnelles acoustique ' Master, Univ. Blida ,2018-2019.