

**UNIVERSITÉ DE BLIDA 1**

**Faculté de Technologie**

Département d'Électronique

**THÈSE DE DOCTORAT**

Spécialité : Électronique

NOUVELLES TECHNIQUES DE RÉDUCTION DU BRUIT ACOUSTIQUE

DÉDIÉES AUX SYSTÈMES DE TÉLÉPHONIE MAINS-LIBRES

Par

**Sofiane Fisli**

Devant le jury composé de :

BENALLAL Ahmed	Prof.	Université de Blida1	Président
BOUZID Merouane	Prof.	USTHB	Examineur
MITICHE Lahcene	Prof.	Université de Djelfa	Examineur
YAKHLEF Yassine	M.C.A	Université de Boumerdes	Examineur
DJENDI Mohamed	Prof.	Université de Blida1	Rapporteur
GUESSOUM Abderrezak	Prof.	Université de Blida1	Co-Rapporteur

Blida, 2021

## RÉSUMÉ

Dans cette thèse, nous nous sommes intéressés aux algorithmes à base des métaheuristiques appliqués à la séparation de sources aveugle, l'annulation du bruit et le rehaussement de la parole. Dans un premier temps, nous présentons la méthode générale pour implémenter les algorithmes métaheuristiques aux structures directes de séparation de sources bi-capteurs, puis nous proposons un nouvel algorithme prédateur-proie dans un essaim particulaire modifié (MPPPSO). Cet algorithme est une amélioration de l'algorithme appelé prédateur-proie dans un essaim particulaire (PPPSO), il est basé principalement sur l'optimisation par essaim particulaire combiné avec le mécanisme du prédateur – proie. Les résultats de simulations présentés dans ce manuscrit viennent confirmer la supériorité de l'algorithme proposé dans le cadre de ce travail.

**Mots-clés :** Modèle du mélange bi-capteurs, Algorithme du filtrage adaptatif, Séparation de sources aveugles, Métaheuristique, Optimisation par essaim particulaire, Rapport signal sur bruit, Qualité du signal de parole.

## ملخص

في هذه الأطروحة، نحن مهتمون بخوارزميات الميتا-هيريستك لاستعمالها إلى تطبيقات فصل المصادر، الغاء الضجيج وتحسين الإشارة الصوتية. أولاً سنعرض الطريقة العامة اللازمة الاستعمال الميتا-هيريستك في فصل المصدر بتقنية الاستشعار الثنائي نحو الامام، في الجزء الثاني نقوم باقتراح خوارزمية جديدة تدعى خوارزمية اسراب العناصر مدمجة بميكانيزم المفترس والضحية المحسنة. هذه الخوارزمية هي امتداد وتحسين للخوارزمية الاصلية من اجل تجنب الحلول المسبقة وبالتالي تحسين نوعية النتيجة مما سينعكس بالإيجاب على نوعية الصوت في النهاية نتائج المحاكاة المقدمة في هذا العمل اظهرت قوة وكفاءة الخوارزمية المقترحة في فصل المصادر والغاء الضوضاء وتحسين الصوت.

**الكلمات الدالة:** فصل المصادر، تحسين الإشارة الصوتية الاستشعار الثنائي نحو الامام، الميتا-هيريستك خوارزمية اسراب العناصر، نسبة الإشارة إلى الضوضاء ، جودة إشارة الصوت.

## ABSTRACT

In this thesis, we are interested in metaheuristic-based algorithms applied to blind source separation, noise cancellation and speech enhancement. First, we present the general method to implement metaheuristic algorithms with Dual-channel forward separation structure, then we propose a modified predator-prey particle swarm optimization algorithm (MPPPSO).

This algorithm is an improvement of the called predator-prey in a particle swarm optimization algorithm (PPPSO), it is mainly based on particle swarm optimization combined with the predator-prey mechanism.

The simulation results presented in this manuscript confirm the superiority of the algorithm proposed in this work.

**Keywords:** Dual-channel convolutive mixing model, Adaptive filtering algorithm, Blind Source Separation, Metaheuristics, Particle Swarm Optimization, Signal to Noise Ratio, Speech Signal Quality.

## Remerciement

Que Dieu le Tout-Puissant soit glorifié pour m'avoir donné le courage, la patience et la persistance pour aboutir et mener à terme mon travail.

Je tiens en premier lieu à remercier le directeur de thèse, le professeur Djendi Mohamed pour avoir accepté de m'encadrer et de m'avoir permis d'évoluer pendant toute cette période dans un environnement scientifique en profitant d'une grande liberté dans l'orientation de mon travail aussi bien sûr le plan scientifique que sur le plan humain, je le remercie pour sa disponibilité et son écoute. J'exprime ma gratitude à monsieur le Professeur Guessoum Abderrazak, qui a accepté de Co encadrer ce travail. Ses remarques fructueuses et très précieux conseils m'ont permis d'accomplir ce travail.

Je souhaite également remercier toute l'équipe, qui a contribué de près ou de loin à l'aboutissement de mon travail, je pense particulièrement à toute l'équipe du Laboratoire d'Automatique et Informatique de Guelma.

Je remercie vivement l'ensemble des membres du jury, qui ont accepté de juger et de porter des critiques constructives à mon travail.

Parce qu'ils représentent tant pour moi, et en raison de leurs soutiens, je profite pour remercier mes parents et toute ma famille, pour leurs encouragements. Je tiens à dédier ce travail tout particulièrement à ma femme, sa compréhension, sa patience et ses encouragements dans les moments délicats m'ont permis de surmonter toutes les difficultés rencontrées. Je dédie ce travail à mes enfants Safa , Anis et Habiba.

## TABLE DES MATIÈRES

RÉSUMÉ.....	1
ملخص.....	2
ABSTRACT .....	3
Remerciement.....	4
LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX.....	8
TABLES .....	12
INTRODUCTION GÉNÉRALE .....	13
Chapitre 1 .....	16
ÉTAT DE L'ART SUR L'OPTIMISATION HEURISTIQUE ET .....	16
MÉTAHEURISTIQUE.....	16
1.1 Introduction .....	17
1.2 Définition de l'optimisation.....	17
1.3 Les Algorithmes de recherche.....	18
1.4 Les Algorithmes heuristiques .....	20
1.4.1 Les algorithmes de recherche locale .....	20
1.4.2 Les algorithmes heuristiques constructifs .....	20
1.5 Les algorithmes métaheuristiques.....	21
1.5.1 Les algorithmes métaheuristiques à solution unique .....	21
1.5.2 Les algorithmes à base de population .....	22
1.6 L'optimisation par essaim particulaire (PSO) .....	22
1.7 L'optimisation du loup gris (GWO) .....	26
1.8 Algorithme de chauve-souris (BA).....	29
1.9 La Métaheuristique dans le domaine du filtrage adaptatif .....	32
1.9.1 Introduction .....	32

1.9.2 Filtre de Wiener.....	32
1.9.3 Les filtres adaptatifs basés sur les méthodes de gradient stochastique ...	34
1.9.3.1 Algorithme de gradient stochastique (LMS) .....	36
1.9.3.2 Algorithme LMS normalisé (NLMS) .....	37
1.9.4 Les filtres adaptatifs basés sur les méthodes métaheuristiques .....	38
1.8 Conclusion .....	38
Chapitre 2 :.....	39
INTRODUCTION SUR LE REHAUSSEMENT DE PAROLE BASÉE SUR LA TECHNIQUE SÉPARATION DE SOURCE AVEUGLE .....	39
2.1 Introduction .....	40
2.2. Principes de la SAS dans le domaine de la réduction du bruit et le rehaussement de la parole.....	41
2.3 Les méthodes mono-capteur.....	41
2.3.1 Algorithmes de la soustraction spectrale .....	43
2.3.2 Filtrage de Wiener.....	45
2.4 Les méthodes multi-capteurs .....	47
2.4.1 Le principe de la séparation aveugle de sources .....	47
2.4.2 Mélange instantané.....	48
2.4.3 Mélange convolutif .....	49
2.5 Mélange linéaire convolutif bi-capteurs .....	52
2.6. Méthode de séparation de sources dans un mélange convolutif bi-capteurs .	54
2.6.1. Structure directe (Forward) .....	54
2.6.2. Structure récursive (Backward).....	56
2.7 État de l'art sur les algorithmes adaptatifs utilisés dans la structure Forward .	58
2.7.1 Algorithme de gradient stochastique double (2C-FLMS) .....	58
2.7.2 Algorithme de gradient stochastique normalisé double (2C-FNLMS) .....	60
2.7.3 Algorithme Forward à pas d'adaptations variables (2C-VSSF).....	61
2.8 Conclusion .....	62
Chapitre 3 .....	64

IMPLÉMENTATION DES MÉTAHEURISTIQUES DANS LE REHAUSSEMENT DE LA PAROLE .....	64
3.1 Introduction .....	65
3.2 Les métaheuristiques pour le filtrage adaptatif.....	66
3.3 Choix de la fonction objective.....	66
3.4 Implémentation du PSO dans la structure Forward bi-capteurs .....	68
3.4.1 Détecteur manuel d'activité vocale (MVAD).....	68
3.4.2 Mise à jour des filtres adaptatifs.....	70
3.5 Implémentation du PPPSO dans la structure Forward bi-capteurs .....	71
3.5.1 Formulation de l'algorithme d'optimisation prédateur proie dans un essaim particulière .....	71
3.5.3 Méthode d'implémentation du PPPSO dans la structure Forward .....	74
3.6 CONCLUSION .....	78
CHAPITRE 4 .....	79
APPLICATION DU PPPSO ET MPPPSO DANS LA STRUCTURE FORWARD ..	79
4.1 Introduction .....	80
4.2 Description de l'algorithme proposé Dual MPPPSO .....	81
4.3 Formulation de l'algorithme proposé MPPPSO .....	81
4.4 Résumé de l'algorithme bi capteurs 2C-FNLMS .....	83
4.5 Résumé de l'algorithme Forward à pas d'adaptations variables (2C-VSSF)...	84
4.6 Évaluation des résultats de simulation .....	84
4.6.1 Critère de désajustement (System Mismatch) .....	84
4.6.2 Rapport signal à bruit segmenté .....	85
4.6.3 Description des signaux d'entrée .....	85
4.6.4 Test avec un bruit blanc.....	87
4.6.4.1 Description temporelle des signaux estimés et leurs spectrogrammes .....	88
4.6.4.2 Évaluation du System Mismatch (SM) .....	90
4.6.4.3 Évaluation du rapport signal sur bruit (SNR segmenté).....	90

4.6.5 Test avec le bruit USASI.....	91
4.6.5.1 Description temporelle des signaux estimés et leurs spectrogrammes .....	92
4.6.5.2 Évaluation du System Mismatch (SM).....	92
4.6.5.3 Évaluation du rapport signal sur bruit (SNR segmenté).....	97
4.7 Conclusion .....	97
CONCLUSION GÉNÉRALE.....	99
LISTE DES SYMBOLES ET DES ABRÉVIATIONS.....	101
SYMBOLES .....	103
RÉFÉRENCES.....	105

## LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX

### Chapitre 1

Figure 1.1 Classification des algorithmes de recherche.

Figure 1.2 Volée d'oiseaux.

Figure 1.3 Mécanisme de recherche du PSO.

Figure 1.4 Organigramme de l'optimisation par essaim particulaire

Figure 1.5 Hiérarchie sociale du loup gris.

Figure 1.6 Organigramme de l'optimisation par l'algorithme du loup gris.

Figure 1.7 Mécanisme d'écholocation de la chauve-souris.

Figure 1.8 Organigramme de l'optimisation par l'algorithme de la chauve-souris.

Figure 1.9 Modèle utilisé pour l'annulation du bruit.

Figure 1.10 Système d'identification adaptative.

### Chapitre 2

Figure 2.1 Débruitage mono-capteur.

Figure 2.2 Classification des méthodes de Débruitage de la parole.

Figure 2.3 Schéma du mélange instantané

Figure 2.4 Schéma du mélange convolutif.

Figure 2.5 Modèle du mélange convolutif.

Figure 2.6 Modèle du mélange convolutif bi-capteurs.

Figure 2.7 Modèle simplifié d'un mélange convolutif bi-capteurs.

Figure 2.8 Structure directe « Forward BSS ».

Figure 2.9 Structure récursive (Backward BSS).

Figure 2.10 Structure complète d'un mélange convolutif et la structure directe.

### Chapitre 3

Figure 3.1 Schéma général d'un système de filtrage adaptatif.

Figure 3.2 Découpage du signal audio à trois taux de chevauchement différent

Figure 3.3 Organigramme général de la méthode d'implémentation des algorithmes métaheuristiques dans la structure Forward.

### Chapitre 4

Figure 4.1 Description temporelle des signaux originaux (a) : le signal de parole  $s(n)$ , (b) : le signal du bruit blanc  $b(n)$ .

Figure 4.2 Exemple des réponses impulsionnelles,  $L = 100$  (a) :  $h_{12}(n)$ , (b) :  $h_{21}(n)$ .

Figure 4.3 Description temporelle des signaux du mélange (a) : le signal  $p_1(n)$ , (b) : le signal  $p_2(n)$ .

Figure 4.4 Description temporelle et spectrogrammes de : (a) signal original, (b) Signal estimé obtenu par le Dual MPPPSO (c) signal estimé obtenu par le 2C-VSSF (d) signal estimé obtenu par le 2C-FNLMS( $\mu_{21}=1$ ) (e) signal estimé obtenu par le 2C-FNLMS( $\mu_{21}=0.5$ ) (f) signal estimé obtenu par le PPPSO.  $RSB_1 = RSB_2 = -3dB$ . La deuxième

source est un bruit blanc, la taille du filtre est  $L = 100$ .

Figure 4.5 Évaluation du système Mismatch du filtre adaptatif  $w_{21}(n)$  obtenue par Dual MPPPSO, PPPSO, 2C-FNLMS ( $\mu_{21}=0.5,1$ ) et 2C-VSSF. Le deuxième source est un bruit blanc. la taille du filtre est  $L = 100$  [En haut] et  $L = 256$  [En bas]. [À gauche :  $RSB_1 = RSB_2 = -3dB$ ], [À droite :  $RSB_1 = RSB_2 = 0dB$ ].

Figure 4.6 Évaluation du SegSNR du filtre adaptatif  $w_{21}(n)$  obtenue par Dual MPPPSO, PPPSO, 2C-FNLMS( $\mu_{21}=0.5,1$ ) et 2C-VSSF. La deuxième source est un bruit blanc. la taille du filtre est  $L = 100$  [En haut] et  $L = 256$  [En bas]. [À gauche :  $RSB_1 = RSB_2 = -3dB$ ], [A droite :  $RSB_1 = RSB_2 = 0dB$ ].

Figure 4.7 Évaluation du SegSNR du filtre adaptatif  $w_{21}(n)$  obtenue par Dual MPPPSO, PPPSO, 2C-FNLMS ( $\mu_{21}=0.5,1$ ) et 2C-VSSF. La deuxième source est un bruit blanc. la taille du filtre est  $L = 100$  [En haut] et  $L = 256$  En Bas]. [À gauche :  $RSB_1 = RSB_2 = -3dB$ ], À droite :  $RSB_1 = RSB_2 = 0dB$ ].

Figure. 4.8 Description temporelle et spectrogrammes de : (a) signal original, (b) Signal estimé obtenu par le Dual MPPPSO (c) signal estimé obtenu par le 2C-VSSF (d) ) signal estimé obtenu par le 2C-FNLMS( $\mu_{21}=1$ ) (e) signal estimé obtenu par le 2C-FNLMS( $\mu_{21}=0.5$ ) (f) signal estimé obtenu par le PPPSO.  $RSB_1 = RSB_2 = -3dB$ . La

deuxième source est un bruit USASI, la taille du filtre est  $L = 100$

Figure 4.9 Évaluation du système Mismatch du filtre adaptatif  $w_{21}(n)$  obtenue

par Dual MPPSO, PPPSO, 2C-FNLMS( $\mu_{21}=0.5,1$ ) et 2C-VSSF. Le deuxième source est un bruit USASI. La taille du filtre est  $L = 100$  [En haut] et

$L = 256$  [En bas]. [À gauche :  $RSB_1 = RSB_2 = -3dB$ ,

À droite :  $RSB_1 = RSB_2 = 0dB$ ].

Figure 4.10 Évaluation du SegSNR du filtre adaptatif  $w_{21}(n)$  obtenue par Dual MPPSO, PPPSO, 2C-FNLMS( $\mu_{21}=0.5,1$ ) et 2C-VSSF. La deuxième source est un bruit USASI. La taille du filtre est  $L = 100$  [En haut] et  $L = 256$  En

Bas]. [À gauche :  $RSB_1 = RSB_2 = -3dB$ ,

À droite :  $RSB_1 = RSB_2 = 0dB$ ].

## TABLES

## Chapitre 2

Tableau 2.1 Algorithme 2C-FLMS

Tableau 2.2 Algorithme 2C-FNLMS

Tableau 2.3 Algorithme 2C-FLMS

## Chapitre 3

Tableau 3.1 Implémentation le PSO dans la structure Forward bi-capteurs

Tableau 3.1 Données réelles de la voie principale de l'intersection étudiée.

Tableau 3.2 Implémentation du PPPSO dans la structure Forward bi-capteurs

## Chapitre 4

Tableau 4.1 Implémentation du MPPPSO dans la structure Forward bi-capteurs

Tableau 4.2 Paramètres de simulation des algorithmes Dual MPPPSO, PPPSO, 2C-NLMS et 2C-VSSF.

Tableau 4.3 Paramètres de simulation des algorithmes Dual MPPPSO, PPPSO, 2C-NLMS et 2C-VSSF.

## INTRODUCTION\_GÉNÉRALE

Parmi les grands axes de recherche dans le domaine de traitement de signal, on trouve les techniques de réduction de bruit acoustique et l'amélioration de la qualité de la parole lors des communications téléphoniques (notamment en mode mains libres) ou dans les systèmes de téléconférence où la parole est souvent altérée par des bruits acoustiques provenant de sources multiples. Par conséquent, la qualité du signal reçu se dégrade, ce phénomène est un problème majeur dans les techniques de réduction de bruit acoustique et l'amélioration de la qualité de la parole.

Le bruit acoustique est un des problèmes les plus préoccupants des chercheurs, qui désirent satisfaire le besoin d'une meilleure qualité de parole reçue par les auditeurs. Le bruit acoustique est constitué essentiellement des interférences, de bruit de l'environnement, bruit moteur, bruit des roues, etc., et/ou des bruits générés par les composants électroniques des capteurs. Pour faire face au problème du bruit acoustique, plusieurs méthodes et approches ont été proposées dans le domaine temporel, dans le domaine de transformation, dans le sous-espace en plus des approches statistiques. Dans ce travail, on s'intéresse à l'approche dans le domaine temporel et exactement à résoudre le problème du mélange convolutif, où le signal de parole est linéairement mélangé avec différents types de bruits. Ce choix est justifié du fait que les mélanges convolutifs permettent de modéliser un grand nombre de situations réelles tels que la communication numérique, la radio mobile et la téléphonie mains libres. Dans ce type de mélange les signaux observés sont constitués d'un mélange convolutif entre le signal parole et du bruit. Donc notre objectif consiste à restaurer le signal utile à partir de ces observations bruitées.

L'objectif de ce travail est la réduction du bruit acoustique et le rehaussement de la parole par les algorithmes de séparation aveugle de sources bi-capteurs, en utilisant les métaheuristiques.

Dans les travaux présentés dans ce manuscrit, nous présentons une méthode générale d'implémentation des métaheuristiques dans la structure directe de

séparation de source d'un mélange convolutif bi-capteurs, ensuite nous proposons un nouvel algorithme dans le but :

- Augmenter de la vitesse de convergence des algorithmes,
- Améliorer la qualité du signal de parole estimé.

Cette thèse est structurée en quatre chapitres encadrés par une introduction et une conclusion générales. Dans le premier chapitre, nous commençons par une courte introduction sur l'optimisation, nous passons ensuite en revue les approches heuristiques et les algorithmes métaheuristiques les plus récents.

À la fin du chapitre, nous présentons un état de l'art sur l'utilisation des métaheuristiques dans le domaine de réduction du bruit acoustique. Dans le deuxième chapitre, on présente les méthodes d'annulation de bruit acoustique multi-capteurs, précisément la séparation des sources ainsi que les types de mélange. En effet, il est introduit, dans ce chapitre, les principales méthodes utilisées pour résoudre le problème de mélange convolutif, notamment, les différentes méthodes de séparation aveugle des sources (BSS) à deux capteurs. Dans le troisième chapitre, nous présentons la technique du filtrage adaptatif à l'aide des métaheuristiques, ensuite nous allons présenter la méthode d'implémentation de l'algorithme d'optimisation par essaim particulaire (PSO) dans la structure Forward bi-capteurs. Nous allons présenter aussi le principe et la formulation de l'algorithme prédateur proie dans un essaim particulaire (PPPSO), ainsi que son implémentation dans la structure Forward bi-capteurs.

Dans le quatrième chapitre, nous proposons un nouvel algorithme (MPPPSO) développé à partir de l'algorithme prédateur proie dans un essaim particulaire, ce dernier est modifié dans le but d'améliorer la qualité de la solution obtenue en maintenant tout l'essaim en mouvement le plus longtemps possible et empêcher une convergence prématurée de l'algorithme, ce nouvel algorithme est implémenté par la suite dans la structure Forward bi-capteurs.

Nous terminons ce manuscrit par une conclusion générale sur les travaux

effectués avec des perspectives qui peuvent être développées par la suite.

**Chapitre 1**  
**ÉTAT DE L'ART SUR L'OPTIMISATION HEURISTIQUE ET**  
**MÉTAHEURISTIQUE**

# Chapitre 1

## ÉTAT DE L'ART SUR L'OPTIMISATION HEURISTIQUE ET MÉTAHEURISTIQUE

### 1.1 Introduction

L'optimisation a pour but de trouver la meilleure solution appelée aussi optimum en respectant des contraintes données d'un problème d'ingénierie ou d'autre secteur, en utilisant des méthodes de résolution exacte [1], des méthodes heuristiques ou métaheuristiques [2]. Si les méthodes exactes permettent de trouver des solutions optimales pour des problèmes de taille raisonnable, elles rencontrent des difficultés face aux applications de taille et de dimension importante. En revanche les méthodes métaheuristiques représentent une bonne alternative suite à leur capacité d'éviter les minimums locaux même dans les applications les plus complexes et cela dans la majorité des cas.

Dans ce chapitre, nous commençons par une courte introduction sur l'optimisation, nous passons ensuite en revue les approches heuristiques et les algorithmes métaheuristiques les plus récents, à la fin du chapitre, nous présentons un état de l'art sur l'utilisation des métaheuristiques dans le domaine de réduction du bruit acoustique.

### 1.2 Définition de l'optimisation

L'optimisation sert à trouver la meilleure solution pour un modèle parmi plusieurs optimums locaux et globaux, d'une manière générale on peut interpréter un problème d'optimisation dans sa forme de minimisation de la manière suivante :

$$\min_{x \in \mathbb{R}^n} f(x) \tag{1.1}$$

Où  $f$  est une fonction continue, dont l'objectif est de trouver la valeur minimale de la fonction  $f^*$  et tous les points  $x^* \in R^n$  tel que :

$$f^* = f(x^*) \leq f(x) \quad \forall x \in R^n \quad (1.2)$$

Le problème de maximisation est obtenu en minimisant la fonction  $-f(x)$ , dans cette thèse les problèmes d'optimisation sont considérés et traités comme des problèmes de minimisation.

### 1.3 Les Algorithmes de recherche

On trouve dans la littérature deux grandes classes de technique d'optimisation, les techniques déterministes ou exactes et les algorithmes issus de l'approche heuristique [3], cette alternative de classification est illustrée en détail dans figure 1.1.

Les méthodes d'optimisation exacte ont souvent des difficultés d'introduire les facteurs aléatoires d'une part, d'autre part ces méthodes nécessitent l'accès à l'information complète sur le système à optimiser or ce n'est pas toujours évident pour les systèmes réels, en plus elles nécessitent un temps de calcul très élevé lorsqu'il s'agit d'un système complexe sans la garantie d'avoir une solution de bonne qualité, cependant les heuristiques ont pris beaucoup d'attention ces dernières décennies par rapport aux méthodes exactes du fait que ces méthodes gèrent d'une manière efficace la nature aléatoire d'un système en plus de leur efficacité dans beaucoup de domaines même ceux les plus complexes en fournissant une solution optimale ou proche de l'optimale [4].

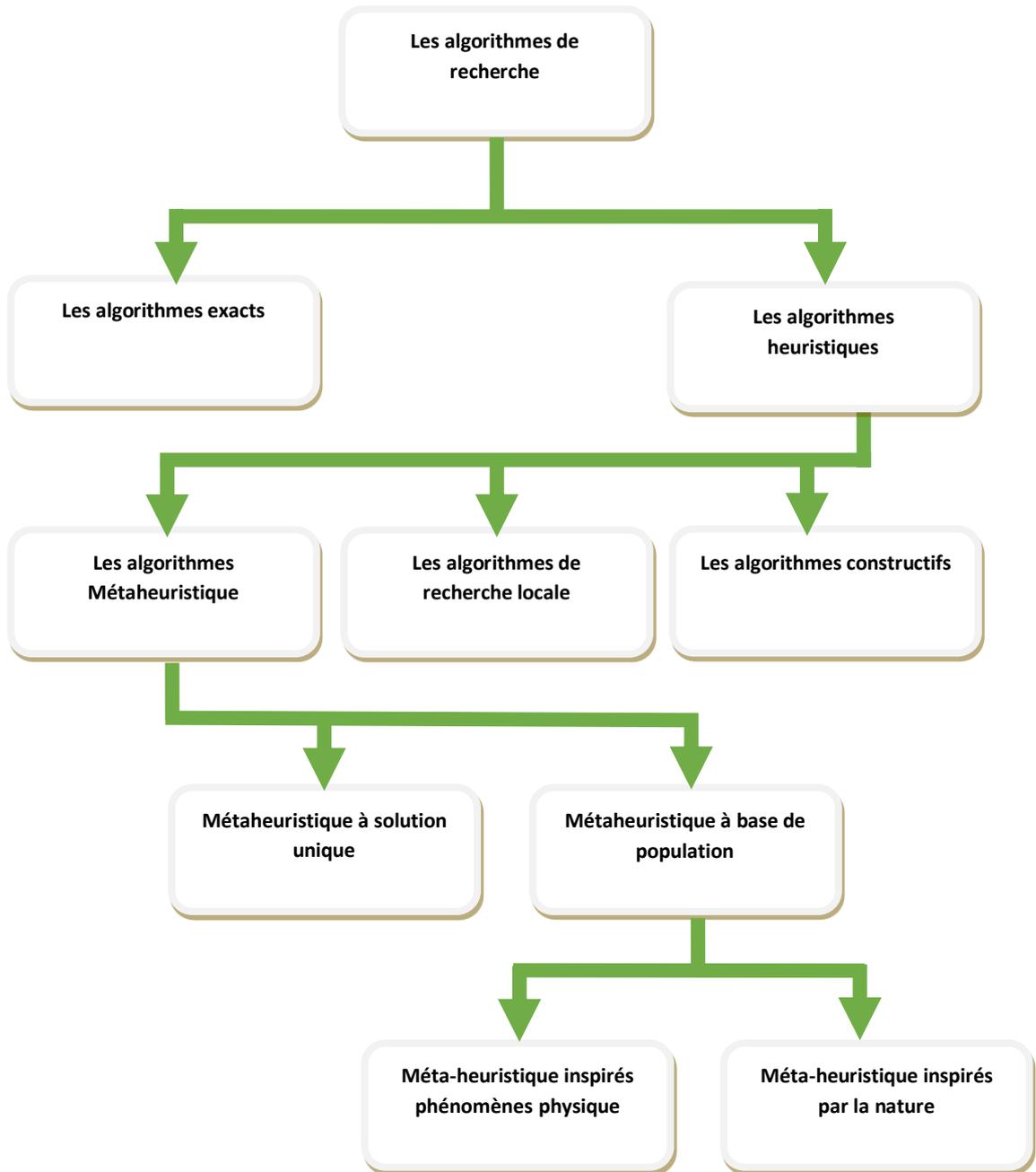


Figure 1.1 : classification des algorithmes de recherche

## 1.4 Les Algorithmes heuristiques

C'est la deuxième grande classe des algorithmes de recherche, ces algorithmes sont devenus de plus en plus populaires, leur avantage par rapport aux méthodes classiques en terme de la qualité de la solution finale d'une part et leurs rapidité de convergence d'une autre part [2] à encourager les chercheurs a les utilisées pour résoudre les problèmes d'optimisation dans plusieurs disciplines même si on sacrifie souvent l'optimum global au profit d'une solution voisine de la solution optimale et au profil d'un temps de traitement considérablement réduit, les heuristiques peuvent être classées en trois catégories les algorithmes de recherche locale, les algorithmes constructifs et les métaheuristiques.

### 1.4.1 Les algorithmes de recherche locale

Cette méthode utilise la notion de voisinage, son principe de base est d'employer une solution initiale et à partir de cette solution on cherche une autre solution voisine qui répond à l'exigence de la décroissance de la valeur de la fonction objective, le déplacement d'une solution à une autre s'appelle mouvement, ce mouvement est procédé d'une manière itérative jusqu'à ce qu'on rencontre un critère d'arrêt précédemment défini, le handicap majeur de cette méthode c'est la difficulté d'échapper au minimum local quand la recherche au voisinage de ce dernier n'arrivera pas à améliorer la valeur de la fonction objective [5] .

### 1.4.2 Les algorithmes heuristiques constructifs

Les algorithmes heuristiques constructifs consistent à construire un ensemble de solutions depuis un espace de recherche en ajoutant au bon moment à une solution initiale partiellement vide des composants d'une solution prédéfinie [6].

## 1.5 Les algorithmes métaheuristiques

Le mot Métaheuristique ou 'metaheuristic' en anglais a été utilisé pour la première fois en 1986 par F.Glover [7], il est composé de deux mots Grec 'heuristic' qui signifie 'trouver' et le mot 'Meta' qui veut dire 'niveau élevé', les algorithmes métaheuristiques sont des techniques d'optimisation stochastique de haut niveau, ces algorithmes sont des approches non déterministes qui emploient un processus itératif en reproduisant le même schéma un certain nombre de fois au cours de son exécution afin d'arriver à une approximation de la solution optimale, ils sont adaptables à un grand nombre de problèmes sans changements majeurs dans leurs algorithmes et sans la nécessité d'avoir une information complète sur le système, ces particularités leur ont permis d'être appliqués pour l'optimisation des systèmes les plus complexes et dans plusieurs domaines où les techniques classiques et heuristiques échouent.

Les métaheuristiques sont en évolution permanente, plusieurs méthodes sont proposées jusqu'à ce jour pour améliorer la résolution des problèmes les plus complexes. La différence entre tous les métaheuristiques réside dans la manière dont ils gèrent la balance entre l'exploration et l'exploitation [8], cette balance est très importante du fait qu'elle sert à identifier rapidement un espace de recherche qui contient des solutions de haute qualité et d'un autre côté d'éviter les espaces de recherche inutile ; c'est-à-dire les régions qui sont déjà explorées et/ou des espaces qui contiennent des solutions de mauvaise qualité.

Les algorithmes métaheuristiques peuvent être classés en deux grandes classes, les algorithmes à base de population et les algorithmes de trajectoire ou les métaheuristiques à solution unique [2].

### 1.5.1 Les algorithmes métaheuristiques à solution unique

Les algorithmes métaheuristiques à solution unique sont basés sur l'évolution d'une seule solution à chaque itération selon une trajectoire dans l'espace de recherche, ce type de métaheuristiques peut être différencié selon la nature de leur trajectoire dans l'espace de recherche, il y a ceux qui suivent une trajectoire de recherche unique qui correspond typiquement à une marche fermée dans le graphe de

voisinage, et ceux qui autorise des sauts dans le graphe de voisinage [9], plusieurs algorithmes sont considérés comme des métaheuristiques à solution unique, on cite la méthode du recuit simulé [10],[11] la recherche Tabou [7] [12], la méthode GRASP (Generic seaRch Algorithm for the Satisfiability Problem) [13], la recherche à voisinage variable [14], la recherche locale itérée [15],etc. En général, ce type de métaheuristiques est axé essentiellement sur l'exploitation plus tôt que l'exploration, ce qui les rend parfois inefficaces face aux problèmes de large dimension et réduit leur capacité d'éviter les minimums locaux.

### 1.5.2 Les algorithmes à base de population

Les algorithmes à base de population contrairement aux algorithmes à solution unique se basent sur l'utilisation d'une population de solutions en même temps, les membres de cette population sont remplacés par de nouveaux membres suivant une stratégie de compétition et cela a chaque itération ce qui permet une meilleure exploration de l'espace de recherche, les algorithmes à base de population deviennent de plus en plus populaires, leur plus grand avantage c'est qu'ils gèrent d'une manière efficace la balance entre l'exploitation et l'exploration de l'espace de recherche d'une part, et d'autre part leur simplicité ce qui les rend très souples a adapté à n'importe quel système même le plus complexe [8]. Ces dernières décennies ont connu une évolution rapide des algorithmes à base de population, la majorité de ces algorithmes sont inspirés soit des phénomènes physiques tel que l'algorithme de recherche gravitationnel qui se base sur la loi de la gravité soit inspiré par la nature telle que l'algorithme génétique. Dans ce qui va suivre on présente en détail trois parmi les plus récents algorithmes à base de population à savoir, l'optimisation par essaim particulaire (PSO) [16], optimisation du loup gris (GWO) [17] et l'algorithme de chauve-souris (BA) [18].

### 1.6 L'optimisation par essaim particulaire (PSO)

L'optimisation par essaim particulaire (PSO) est un algorithme Métaheuristique qui appartient à la classe des algorithmes à base de population, il a été présenté pour la première fois par Eberhart et Kennedy [16], [19] cette technique d'optimisation s'inspire fortement des relations grégaires des oiseaux, des poissons et même des

abeilles , Figure 1.2. Le PSO commence par générer un ensemble de particules d'une manière aléatoire, ces particules représentent des solutions probables, ensuite l'algorithme essaye d'améliorer la qualité de la solution toute en s'articulant sur l'amélioration de la fonction coût. Les particules changent de position à chaque itération en suivant certaine stratégie régie par des équations mathématiques simples, ces équations vont aider les particules à se communiquer entre elles d'une façon que le mouvement de chaque particule s'oriente vers sa meilleure position mémorisée et la meilleure position de l'ensemble des particules. Les équations de mise à jour de la position des particules font l'objet de plusieurs recherches ces dernières années ce qui a mené à la naissance de plusieurs variantes du PSO.



Figure 1.2 : Volée d'oiseaux

Un essaim de particules est caractérisé par un nombre de particules noté  $n_p$ , chaque particule est caractérisée par un vecteur de position  $\vec{x}$  et un vecteur de vitesse  $\vec{v}$  de la même dimension du problème à optimisé  $D$ , la première étape de l'algorithme consiste à générer aléatoirement la position et la vitesse de chaque particule dans l'espace de recherche. Puis on évalue la fonction objective en chaque particule pour trouver la meilleure solution de tout l'essaim  $G_{best}$  et sa meilleure performance  $Best\_Cost$ . Par la suite, à chaque itération, les particules adaptent leur

vitesse  $\vec{v}$ , leur position  $\vec{x}$  en tenant compte leur position et vitesse actuelle ainsi que la meilleure position de tout l'essaim suivant le mécanisme illustré dans la figure 1.3 et en utilisant les deux équations suivantes :

$$\mathbf{v}_i^{k+1} = w \mathbf{v}_i^k + c_1 r_1 (\mathbf{b}_i^k - \mathbf{x}_i^k) + c_2 r_2 (\mathbf{g}^k - \mathbf{x}_i^k) \quad (1.3)$$

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \mathbf{v}_i^{k+1} \quad (1.4)$$

Où

$\mathbf{v}_i^k, \mathbf{x}_i^k$  sont la vitesse et position de la  $i^{\text{ième}}$  particule a l'itération  $k$ ,

$\mathbf{b}_i^k$  est la meilleure position visitée par la  $i^{\text{ième}}$  particule,

$\mathbf{g}^k$  désigne meilleure position de tout l'essaim,

$r_1, r_2$  sont deux nombres aléatoires tirés uniformément dans  $[0, 1]$ ,

$w, c_1$  et  $c_2$  sont les paramètres d'inertie, cognitif et social respectivement.

L'organigramme de l'optimisation par essaim particulaire est illustré dans la figure 1.4

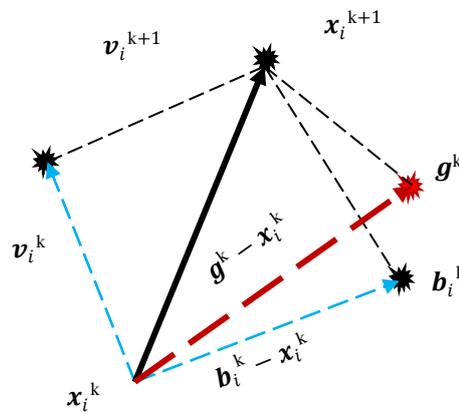


Figure 1.3 :Mécanisme de recherche du PSO

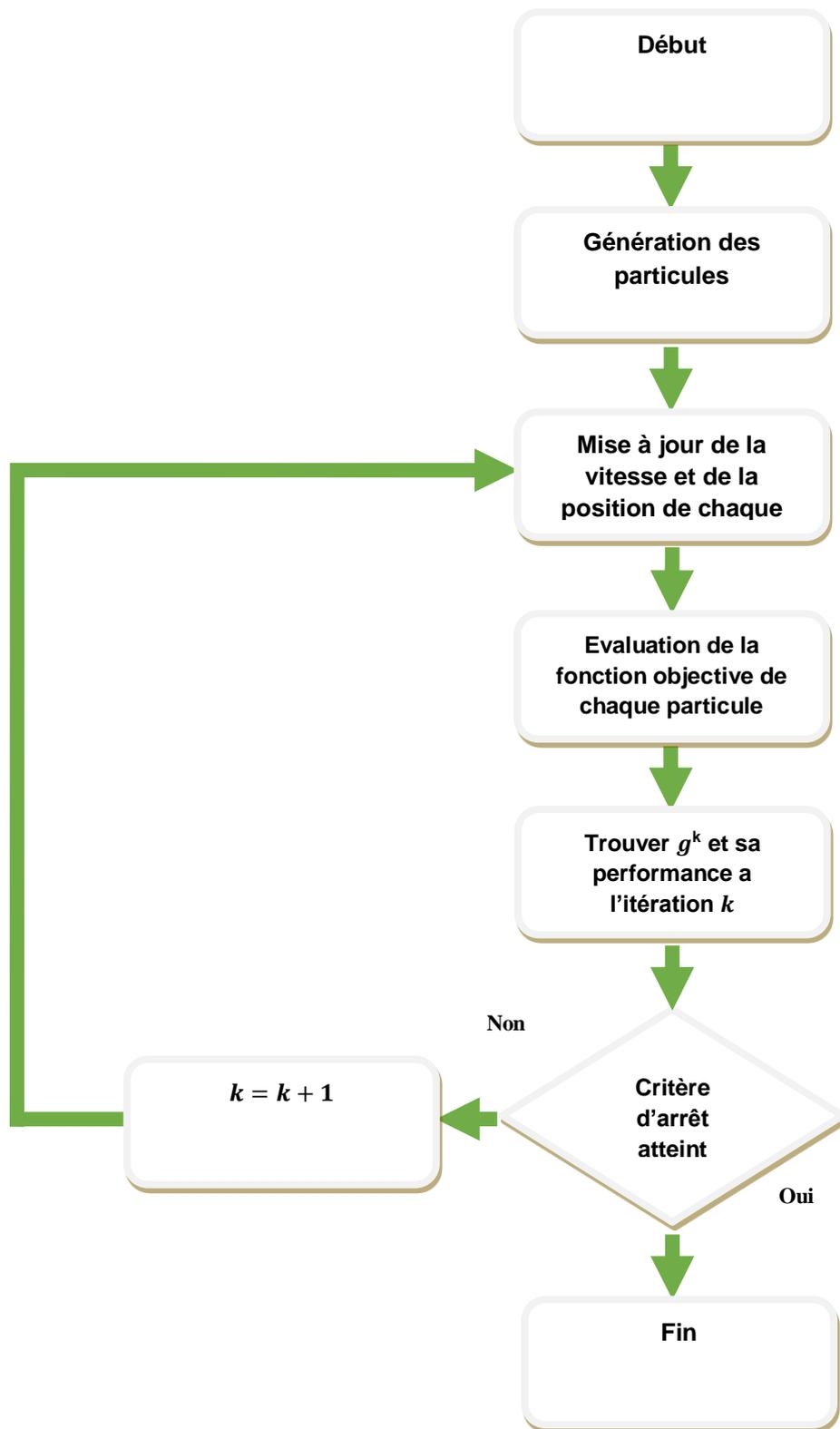


Figure 1.4 : Organigramme de l'optimisation par essaim particulaire

### 1.7 L'optimisation du loup gris (GWO)

L'optimisation du loup gris (GWO) est une technique d'optimisation Métaheuristique développée par Mirjalili [17], cette technique est inspirée du comportement des loups gris connus par leur stratégie de chasse en groupe.

Les loups gris préfèrent vivre en groupe, ils sont organisés d'une manière hiérarchique, dirigée par un chef dominant soit un mâle ou une femelle, ce chef est appelé Alpha, la plupart du temps c'est l'Alpha le responsable de toutes les décisions prises, ces décisions doivent être respectées par l'ensemble du groupe, l'Alpha est subordonné par un autre loup on l'appelle Bêta son rôle principal c'est d'aider l'Alpha dans tous les ordres, puis vient ensuite le Delta, un loup qui subordonne le Bêta, le Delta supervise le reste du groupe ou les loups dominés qu'on appelle les Oméga figure 1.5.

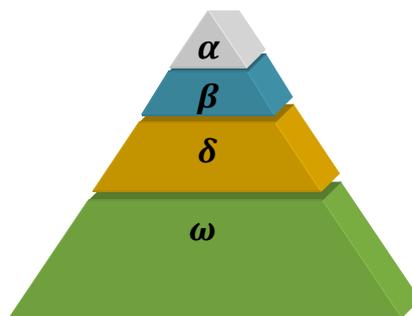


Figure 1.5 : Hiérarchie sociale du loup gris

Cette technique de chasse et la hiérarchie sociale des loups gris sont modélisées par un ensemble d'équations mathématiques afin de concevoir le GWO, le GWO a été employé et testé sur différentes fonctions testes ainsi dans de nombreuses applications réelles avec succès. Dans l'algorithme GWO, l'Alpha est considéré comme la meilleure solution. Il en résulte que, la deuxième et la troisième meilleure solution sont représentées par Bêta et Delta respectivement. Tandis que le reste du groupe, les Oméga représentent des solutions potentielles.

Le GWO peut être résumé dans les étapes suivantes :

Étape 1 : générer aléatoirement l'ensemble des loups, chaque loup est caractérisé par un vecteur position d'éléments qui représentent la dimension du problème à

optimiser, puis initialiser les paramètres  $a$ ,  $A$  and  $C$  en utilisant les équations suivantes :

$$a = 2 - k \left( \frac{2}{\max\_iter} \right) \quad (1.4)$$

$$A = 2a \cdot r_1 - a \quad (1.5)$$

$$C = 2 \cdot r_2 \quad (1.6)$$

Où

$A$ ,  $C$  sont des coefficients matriciels,

$k$ ,  $\max\_iter$  sont l'itération actuelle et le nombre maximum d'itération,

$r_1, r_2$  sont deux nombres aléatoires tirés uniformément dans  $[0, 1]$ ,

$a$  est une valeur décroissante linéairement de 2 à 0.

Étape 2 : évaluation de la fonction objective de tous les loups afin de déterminer l'Alpha  $x_\alpha$ , le Bêta  $x_\beta$  et le Delta  $x_\delta$  puis mettre à jour leur position en utilisant les équations suivantes :

$$x(k + 1) = x(k) - A \cdot D \quad (1.7)$$

$$D = |C \cdot x_p(k) - x(k)| \quad (1.8)$$

Où

$D$  est un vecteur dépendant de la location de la proie  $X_p(k)$ .

$|\cdot|$  désigne la valeur absolue.

Étape 3 : mettre à jour la position du reste des loups à travers les équations suivantes :

$$x(k + 1) = \frac{x_1 + x_2 + x_3}{3} \quad (1.9)$$

Où  $x_1, x_2$  et  $x_3$  sont calculés avec :

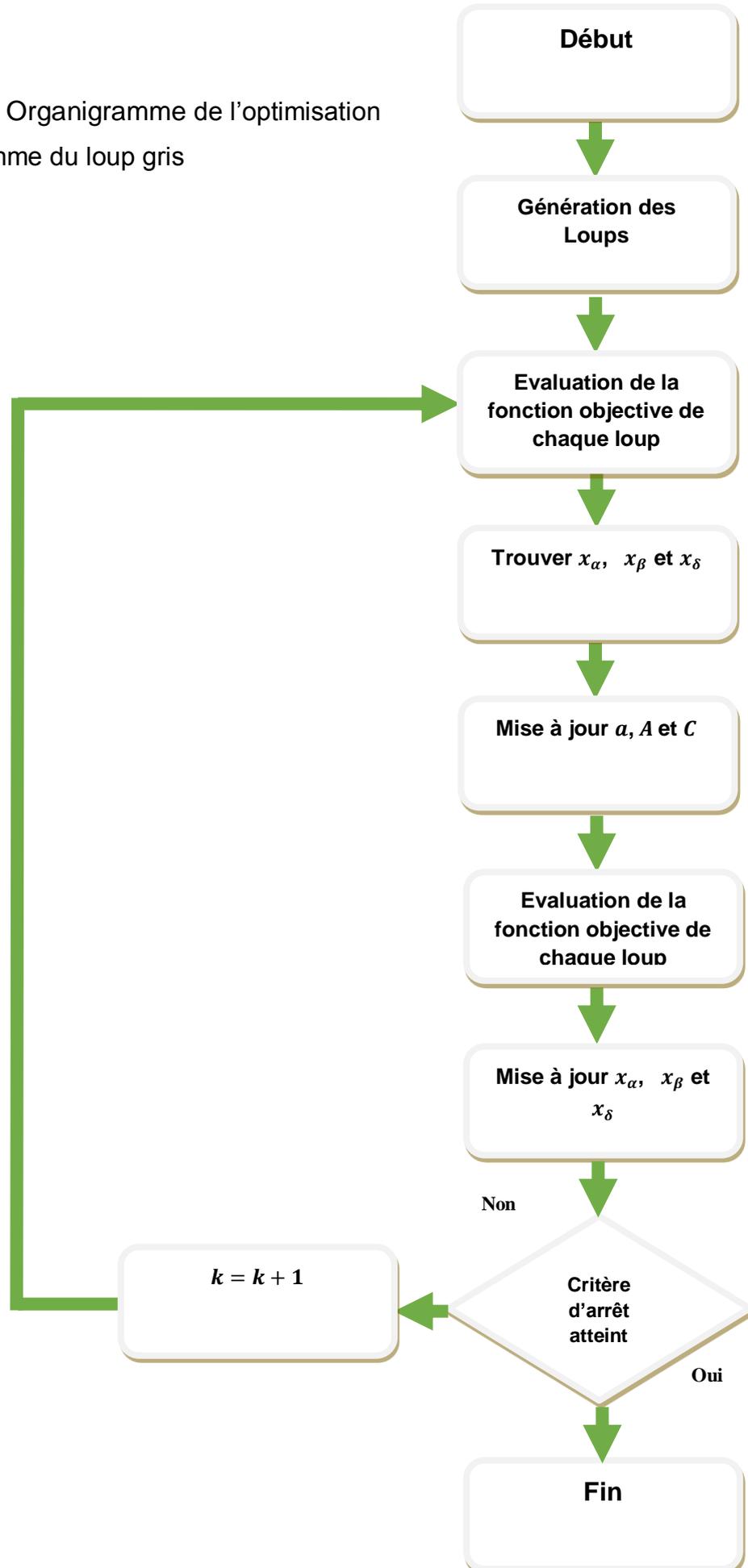
$$x_1 = x_\alpha(k) - A_1 \cdot D_\alpha \quad (1.10)$$

$$x_2 = x_\beta(k) - A_2 \cdot D_\beta \quad (1.11)$$

$$x_3 = x_\delta(k) - A_3 \cdot D_\delta \quad (1.12)$$

$D_\alpha, D_\beta$  et  $D_\delta$  peuvent être calculé avec l'équation (1.8), l'organigramme de l'optimisation par le GWO est illustré dans la figure 1.6.

Figure 1.6 : Organigramme de l'optimisation  
Par l'algorithme du loup gris



### 1.8 Algorithme de chauve-souris (BA)

L'algorithme de chauve-souris (Bat Algorithm) est considéré comme l'un des algorithmes d'intelligence d'essaim les plus prometteurs [18], le BA imite le comportement d'écholocation des chauves-souris, les chauves-souris peuvent chasser ses proies même dans l'obscurité absolue à l'aide de l'écholocation. Les chauves-souris émettent des signaux sonores dans une zone spécifique et en utilisant l'écho de ces signaux, elles parviennent à identifier la localisation et le gabarit des objets dans la zone de recherche. Dans le BA, la régulation de fréquence aide à augmenter la diversité des solutions dans un essaim de chauves-souris tout en essayant d'établir un bon équilibre de la balance entre l'exploitation et l'exploration. Le mécanisme d'écholocation de la chauve-souris est illustré dans la figure 1.7.

L'essaim de chauves-souris initiales est généré aléatoirement qui contient  $n$  individus ou chaque individu représente une solution potentielle dans un espace de recherche, tous les individus sont caractérisés par un vecteur de position, un vecteur de vitesse de dimension  $d$  à chaque itération  $k$  et une fréquence  $f_i$  qui varie aléatoirement dans l'intervalle  $[f_{min} \ f_{max}]$ .

La mise à jour de la position, la vitesse et la fréquence de chaque individu sont données par les équations suivantes :

$$f_i = f_{min} + (f_{max} - f_{min}) \delta_b \quad (1.13)$$

$$v_i^k = v_i^{k-1} + (x_i^{k-1} - x^*) f_i \quad (1.14)$$

$$x_i^k = x_i^{k-1} + v_i^k \quad (1.15)$$

Où

$f_{min}$ ,  $f_{max}$  sont les fréquences minimales et maximales,

$f_i$  est la fréquence de la  $i^{ème}$  chauve-souris,

$v_i^k$ ,  $x_i^k$  sont les vitesses et les positions du  $i^{ème}$  à l'itération  $k$ ,

$\delta_b$  est un nombre aléatoire uniformément distribué dans  $[0 \ 1]$ ,

$x^*$  est la meilleure position déjà visitée.

Il est à signaler que les fourchettes de l'intervalle où la fréquence est distribuée dépendent du problème à traiter. Après avoir mis à jour la meilleure solution, on procède par une dernière étape est la mise à jour de la position de chaque individu en ajoutant un facteur aléatoire pour assurer et améliorer la recherche locale.

$$x_i^k = x_i^k + \epsilon A_i^k \quad (1.16)$$

Où

$\epsilon$  est un nombre aléatoire distribué uniformément dans l'intervalle  $[-1, 1]$ ,

$A_i^k$  désigne l'intensité d'individu à l'itération  $k$ .

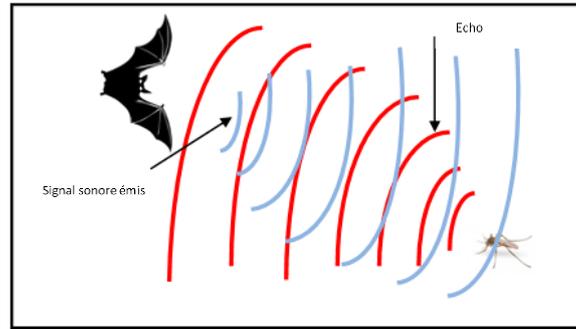


Figure 1.7 : mécanisme d'écholocalisation de la chauve-souris

Dans le BA l'intensité  $A_i$  et le taux de pulsation du signal émis  $r_i$  doivent être mis à jour à chaque itération  $k$ . Pendant l'optimisation, l'intensité diminue une fois que la chauve-souris trouve sa proie, tandis que le taux de pulsation augmente, l'intensité peut être choisie comme valeur de commodité. Lorsque l'intensité atteint la valeur minimale  $A_{min}$ , cela signifie que la chauve-souris a trouvé la proie et par conséquent elle cesse d'émettre un son. Dans le cas le plus simple, on peut choisir  $A_0 = 1$  et  $A_{min} = 0$ , ce qui signifie qu'une chauve-souris vient juste de trouver une proie ce qui entraîne la suspension d'émission de signal sonore, l'intensité et le taux de pulsation peuvent être mis à jour par :

$$A_i^{k+1} = \alpha A_i^k \quad (1.17)$$

$$r_i^{k+1} = r_i^0 (1 - \exp(-\gamma k)) \quad (1.18)$$

Où

$\alpha, \gamma$  sont deux nombres constant déterminer expérimentalement,

$r_i^k$  est le taux de pulsation de la  $i^{\text{ème}}$  chauve-souris à l'itération  $k$ .

L'organigramme de l'algorithme de la chauve-souris est illustré dans la figure 1.8.

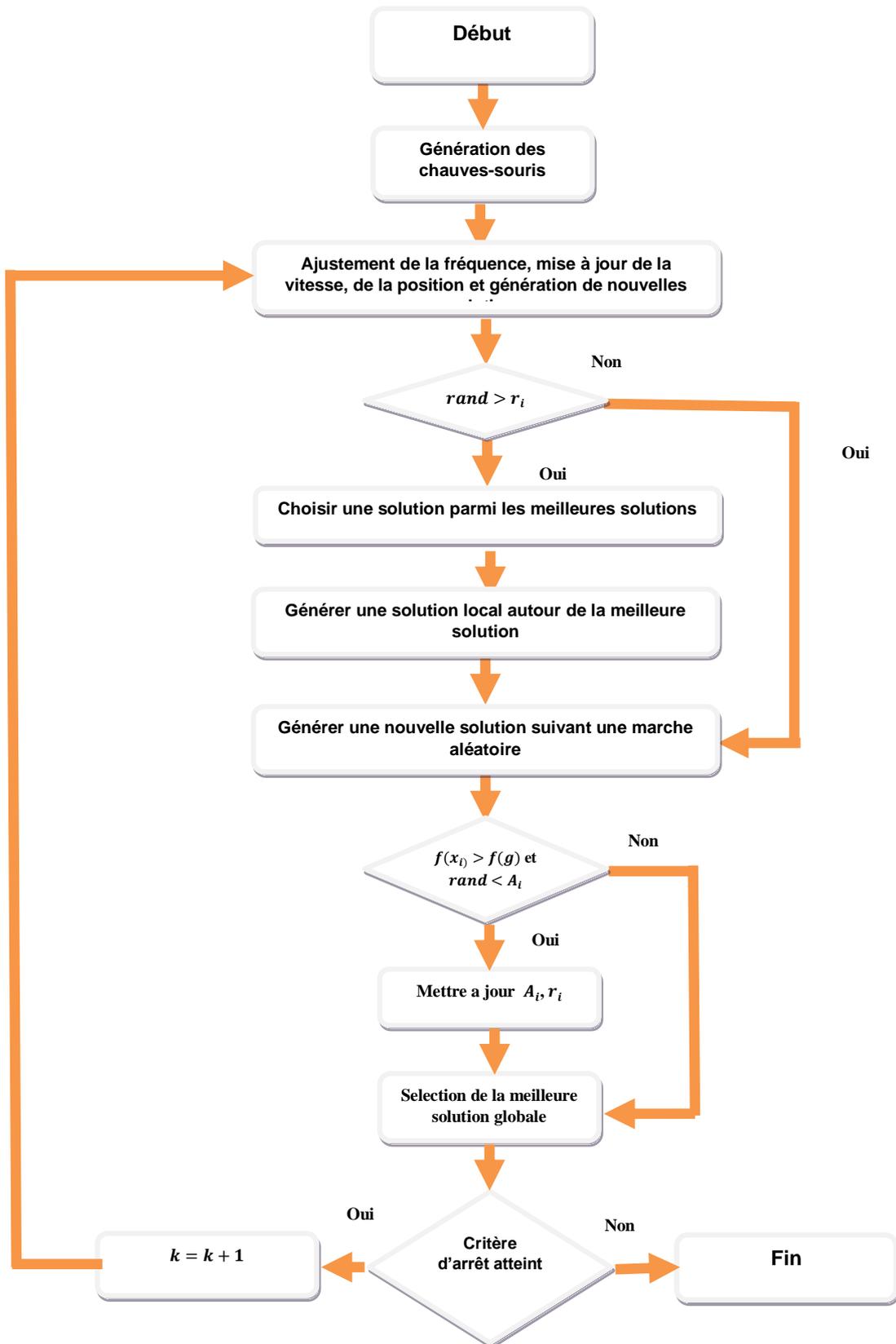


Figure 1.8 : Organigramme de l'optimisation par l'algorithme de la chauve-souris.

## 1.9 La Métaheuristique dans le domaine du filtrage adaptatif

### 1.9.1 Introduction

Les filtres adaptatifs sont des filtres dont les paramètres changent d'une manière automatique dans le but d'obtenir la meilleure qualité d'un signal dans un environnement à caractéristique variable dans le temps [20], ils sont très utilisés dans beaucoup d'applications du traitement du signal tels que les télécommunications, la reconnaissance de parole, le rehaussement de parole ,etc. Le filtrage d'un signal consiste à extraire un signal désiré d'un signal qui contient un signal désiré combiné à un signal de bruit (non désiré), cette tâche est réalisée en utilisant soit un filtre à réponse impulsionnelle finie (RIF) soit un filtre à réponse impulsionnelle infinie (IIR), la sortie du filtre peut être obtenue avec une combinaison linéaire de l'observation et de l'entrée du système, le filtre optimum peut être élaboré en minimisant l'erreur quadratique moyenne entre le signal désiré et la sortie du filtre, dans un environnement bruité avec des signaux non stationnaires le filtre adaptatif constitue le meilleur choix pour une solution meilleure, contrairement au filtre de Wiener qui est généralement employé dans des environnements stationnaires. Dans ce qui suit, on va présenter le principe de base du filtre de Wiener.

### 1.9.2 Filtre de Wiener

Le filtre de Wiener est un outil puissant et largement utilisé dans plusieurs applications de traitement numérique du signal telles que la réduction du bruit, la prédiction linéaire, l'annulation d'écho acoustique et l'identification des systèmes [21]. Son principe de base consiste à minimiser l'erreur quadratique moyenne entre le signal désiré et la sortie du filtre linéaire. La représentation du problème d'annulation du bruit acoustique par un filtre de Wiener est donnée par la figure 1.9. Où  $d(n)$  est un signal primaire qui comporte le bruit à annuler,  $x(n)$  est le signal de référence et obtenu par un capteur proche de celui qui fournit  $d(n)$ .

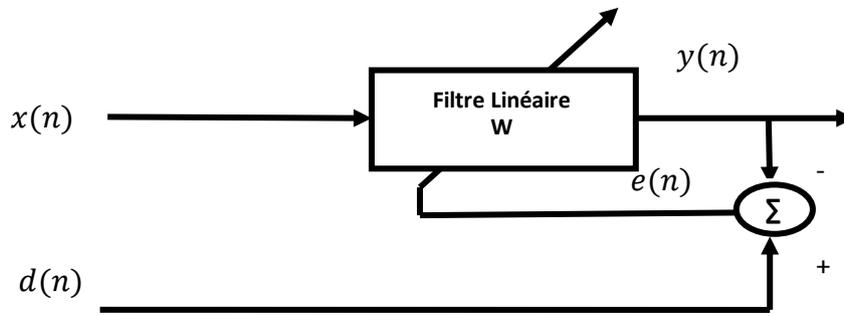


Figure 1.9 : Modèle utilisé pour l'annulation du bruit

Le principe du filtre de Wiener consiste donc à rechercher le filtre  $w(n)$  assurant l'erreur la plus faible entre la réponse  $y(n)$  et la réponse désirée  $d(n)$ .

En ce qui concerne la fonction coût, plusieurs choix sont possibles, parmi ces choix les plus utilisés, c'est l'erreur quadratique moyenne.

En utilisant une notation vectorielle de la sortie du filtre, on peut écrire :

$$y(n) = \mathbf{w}(n)^T \cdot \mathbf{x}(n) \quad (1.19)$$

Avec,

$$\mathbf{w}(n) = [w(n)_0, w(n)_1, \dots, w(n)_{L-1}]^T$$

est un vecteur de longueur  $L$  contenant les coefficients du filtre à réponse impulsionnelle finie (RIF).

$$\mathbf{x}(n) = [x(n), x(n-1), \dots, x(n-L+1)]^T$$

est le vecteur des  $L$  données d'entrée les plus récentes.

Le signal d'erreur  $e(n)$  est donné par :

$$e(n) = d(n) - \mathbf{w}(n)^T \cdot \mathbf{x}(n) \quad (1.20)$$

Le filtre de Wiener sert à trouver les paramètres du filtre  $w(n)$  de telle manière de minimiser l'erreur quadratique moyenne (EQM) :

$$J = E[e^2(n)] \quad (1.21)$$

Donc le filtre optimum  $w_{opt}$  peut être obtenu lorsque la dérivée de l'EQM par rapport aux paramètres est nulle :

$$\frac{\partial J}{\partial \mathbf{w}} = \mathbf{0}_{L \times 1} \quad (1.22)$$

$$\frac{\partial J}{\partial \mathbf{w}(n)} = 2E \left[ e(n) \frac{\partial e(n)}{\partial \mathbf{w}(n)} \right]$$

$$\frac{\partial J}{\partial \mathbf{w}(n)} = -2E[e(n)\mathbf{x}(n)] \quad (1.23)$$

Donc à l'optimum on aura :

$$E[e_{min}(n)\mathbf{x}(n)] = \mathbf{0}_{L \times 1} \quad (1.24)$$

Où  $e_{min}(n)$  est l'erreur qui minimise la fonction coût  $J$ , en se basent sur le principe d'orthogonalité [22], le critère  $J$  atteint sa valeur minimale si et seulement si l'erreur  $e(n)$  est orthogonale aux échantillons du signal d'entrée  $x(n-l)$ , et au signal de sortie  $y(n)$ .

Et à partir de l'équation (1.24) nous aboutissons après développement a :

$$E\{\mathbf{x}(n)[d(n) - \mathbf{x}(n)^T \mathbf{w}_{opt}]\} = \mathbf{0}_{L \times 1} \quad (1.25)$$

$$E[\mathbf{x}(n)\mathbf{x}(n)^T] \mathbf{w}_{opt} = E[\mathbf{x}(n)d(n)] \quad (1.26)$$

On pose :  $\mathbf{R}_{xx} = E[\mathbf{x}(n)\mathbf{x}(n)^T]$  et  $\mathbf{r}_{xd} = E[\mathbf{x}(n) d(n)]$

On obtient :

$$\mathbf{R}_{xx} \mathbf{w}_{opt} = \mathbf{r}_{xd} \quad (1.27)$$

Avec  $\mathbf{R}_{xx}$  est la matrice d'autocorrélation du signal d'entrée, elle est définie positive, de Toeplitz et symétrique et  $\mathbf{r}_{xd}$  est le vecteur d'intercorrélacion entre le signal d'entrée  $x(n)$  et le signal désiré  $d(n)$ .

L'équation (1.27) est connue sous le nom de Wiener-Hopf, cette équation permet de calculer le filtre de Wiener optimal :

$$\mathbf{w}_{opt} = \mathbf{R}_{xx}^{-1} \mathbf{r}_{xd} \quad (1.28)$$

Cette solution peut être obtenue en résolvant un système de l'équation à  $L$  inconnue ce qui nécessite un temps de calcul et un espace mémoire important, de plus la non-stationnarité des signaux ce qui est fréquent nécessite une solution adaptative tout une évitant l'inversion matricielle par un algorithme récursif [23].

### 1.9.3 Les filtres adaptatifs basés sur les méthodes de gradient stochastique

La figure 1.10 montre l'utilisation d'un filtre adaptatif pour l'identification des systèmes [24], dans cette application le filtre adaptatif s'adapte à chaque itération

afin de trouver un modèle optimal pour le système inconnu basé sur la minimisation de l'erreur quadratique moyenne avec :

$x(n)$  est le signal d'entrée du système,

$w(n)$  est le vecteur des paramètres du filtre adaptatif,

$h(n)$  est le vecteur des paramètres du filtre réel ;

$y(n)$  désigne la sortie du filtre adaptatif,

$d(n)$  est la sortie du filtre réel,

$e(n)$  est le signal d'erreur entre  $y(n)$  et  $d(n)$ .

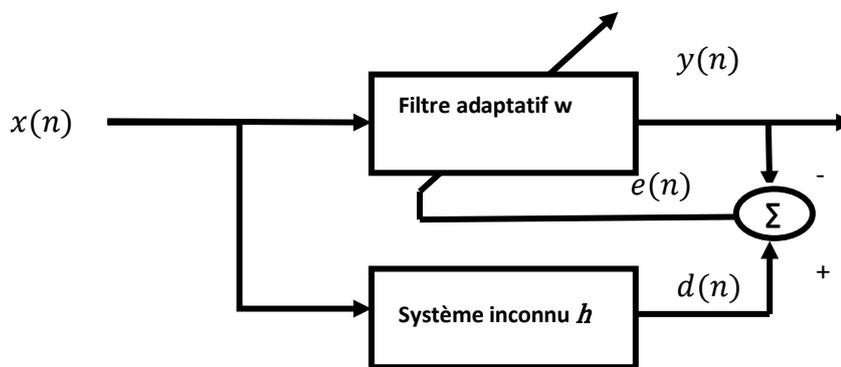


Figure 1.10 : Système d'identification adaptatif

La mise à jour des coefficients du filtre est effectuée d'une façon itérative à l'aide d'algorithme d'optimisation issue des méthodes d'optimisation locales ou globales. [3].

Dans la littérature, on distingue deux grandes classes d'algorithmes adaptatifs pour traiter en ligne le problème du filtrage adaptatif par un filtre à réponse impulsionnelle finie (RIF). Les méthodes de gradient stochastique, qui donneront l'algorithme des moindres carrés moyens (en anglais Least Mean Square ou LMS) et ses variantes, les méthodes moindres carrées et les méthodes moindres carrées récursives (en anglais Recursive Least Square ou RLS) et leur variante ou encore les variantes intermédiaires tel que la méthode des algorithmes de projection affine, les structures en treillis, etc [20].

Dans ce manuscrit, nous présentons deux algorithmes parmi les plus utilisés dans le domaine du traitement adaptatif des signaux à savoir l'algorithme des moindres

carrés moyens (LMS) et l'algorithme des moindres carrés normalisés (NLMS) qui appartient à la famille des algorithmes du gradient stochastique.

### 1.9.3.1 Algorithme des moindres carrés moyens (LMS)

L'algorithme des moindres carrés moyens est l'algorithme le plus utilisé pour ajuster les paramètres du filtre adaptatif en raison de sa simplicité [25],[26]. L'algorithme LMS est une approximation de l'algorithme du gradient déterministe où l'on utilise le signal d'erreur donné par l'équation (1.20) pour calculer le filtre optimal à chaque itération :

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu E[\mathbf{x}(n) e(n)] \quad (1.29)$$

Où  $\mu$  est un pas d'adaptation.

Sauf que dans la pratique la matrice d'autocorrélation  $\mathbf{R}_{xx} = E[\mathbf{x}(n)\mathbf{x}(n)^T]$  et le vecteur d'intercorrélation  $r_{xd} = E[\mathbf{x}(n) d(n)]$  sont inconnus, on les approchera par leur estimées instantanées des corrélations :

$$\hat{\mathbf{R}}_{xx} = \mathbf{x}(n)\mathbf{x}(n)^T \quad (1.30)$$

Et

$$\hat{\mathbf{r}}_{xd} = \mathbf{x}(n) d(n) \quad (1.31)$$

En remplaçant  $\hat{\mathbf{R}}_{xx}$  et  $\hat{\mathbf{r}}_{xd}$  dans (1.29), on obtient l'équation de mise à jour des paramètres du filtre adaptatif :

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{x}(n) e(n) \quad (1.32)$$

En résumé le LMS consiste à :

Calculer la sortie du filtre :  $\mathbf{y}(n) = \mathbf{w}(n)^T \cdot \mathbf{x}(n)$

Calculer le signal d'erreur :  $e(n) = d(n) - \mathbf{w}(n)^T \cdot \mathbf{x}(n)$

Puis mettre à jour le filtre :  $\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{x}(n) e(n)$

Le LMS ne nécessite que  $2L + 1$  multiplications et  $2L$  additions par itération ce qui le rend simple à implémenter.

Il est à noter que le pas d'adaptation joue un rôle important dans la convergence de l'algorithme et aussi son bon fonctionnement en satisfaisant la condition suivant [22], [23], [24], [25], [26] :

$$0 < \mu < \frac{2}{\lambda_{max}} \quad (1.33)$$

Où  $\lambda_{max}$  représente la valeur propre maximale de la matrice d'autocorrélation du

signal d'entrée  $\mathbf{R}_{xx}$ .

### 1.9.3.2 Algorithme LMS normalisé (NLMS)

Pour des signaux non stationnaires ou l'énergie du signal, varie avec le temps comme dans le cas des signaux acoustiques dont les variations d'énergie sont souvent brusques, l'algorithme LMS aura du mal à fonctionner correctement puisque le paramètre d'adaptation  $\mu$  est constant.

Pour remédier aux problèmes reliés aux signaux non stationnaires, une variante du LMS a été introduite par Widrow [29], le gradient normalisé (NLMS : Normalized Least-Mean-Square). Dans le NLMS, la correction des coefficients se fait en fonction de l'énergie du signal d'entrée. Ce calcul fait en sorte que le pas d'adaptation varie de façon inversement proportionnelle à l'énergie contenue dans le signal d'entrée. Ce qui rend le pas d'adaptation  $\mu$  variable et prend une nouvelle valeur à chaque itération :

$$\mu_n = \frac{\mu}{\|\mathbf{x}(n)\|^2} \quad (1.34)$$

L'équation de la mise à jour du NLMS obtenu en remplaçant le pas d'adaptation dans l'équation de mise à jour du LMS par le nouveau pas :

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu_n}{\|\mathbf{x}(n)\|^2 + \varepsilon} \mathbf{x}(n) e(n) \quad (1.35)$$

Où  $\varepsilon$  est un paramètre de régularisation permettant d'éviter la division par de petits nombres quand l'énergie du signal est petite. La convergence de cet algorithme est garantie pour un pas d'adaptation borné entre le zéro et 2.

En résumé, le NLMS consiste à :

Calculer la sortie du filtre :  $\mathbf{y}(n) = \mathbf{w}(n)^T \cdot \mathbf{x}(n)$

Calculer le signal d'erreur :  $e(n) = d(n) - \mathbf{w}(n)^T \cdot \mathbf{x}(n)$

Puis mettre à jour le filtre :  $\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu_n}{\|\mathbf{x}(n)\|^2 + \varepsilon} \mathbf{x}(n) e(n)$

L'algorithme NLMS est très simples à implémenter, il est très utilisé dans le domaine d'annulation d'écho acoustique et de réduction du bruit [30].

#### 1.9.4 Les filtres adaptatifs basés sur les méthodes métaheuristiques

Les algorithmes adaptatifs basés sur les méthodes métaheuristiques constituent un développement plutôt récent dans le domaine du filtrage adaptatif où les paramètres du filtre peuvent être adaptés en minimisant une fonction d'erreur via des algorithmes métaheuristiques.

Dans la littérature on trouve plusieurs algorithmes métaheuristiques qui ont été utilisés afin d'ajuster les paramètres du filtre adaptatif on cite l'algorithme de l'évolution différentiel (DE) qui a été utilisé dans le design d'un filtre d'ordre fractionnaire de type RIF où les méthodes conventionnelles nécessitent un calcul très complexe [31], l'algorithme de colonie des abeilles (ABC) qui a été utilisé avec succès dans le contrôle actif du bruit appliqué à l'électroencéphalogramme (EEG) [32]. Dans [33] l'auteur a employé l'algorithme génétique (GA) pour l'identification adaptative des filtres non-linéaires.

D'autres algorithmes Métaheuristique ont été employés pour le rehaussement de parole et l'annulation du bruit acoustique [34] où l'auteur utilise la structure standard d'annulation du bruit et un filtre adaptatif du type IIR combiné avec plusieurs algorithmes tels que l'essaim particulaire (PSO) et ses variantes [35],[36],[37],[38], l'algorithme de recherche gravitationnelle [39], l'hybridation entre ces deux derniers (PSOGSA) [39] et l'algorithme de chauve-souris (BA) [40].

#### 1.8 Conclusion

Nous avons présenté les différentes techniques d'optimisation. Nous avons vu que les algorithmes de recherche peuvent être classifiés en deux grandes catégories principales ; les algorithmes exacts et les algorithmes heuristiques. Les algorithmes exacts ont souvent des difficultés d'introduire les facteurs aléatoires d'une part, et lorsqu'il s'agit d'un système complexe, ils nécessitent un temps de calcul très élevé d'autre part. Les algorithmes heuristiques sont, classés en trois classes ; les algorithmes constructifs, les algorithmes de recherche locale et les algorithmes Métaheuristique.

Et contrairement aux algorithmes exacts les algorithmes heuristiques gèrent d'une manière efficace l'aspect aléatoire du système et ils permettent de trouver une solution proche de l'optimale même dans les systèmes les plus complexes.

## **Chapitre 2**

# **INTRODUCTION SUR LE REHAUSSEMENT DE PAROLE BASÉE SUR LA TECHNIQUE SÉPARATION DE SOURCE AVEUGLE**

## Chapitre 2

# INTRODUCTION SUR LE REHAUSSEMENT DE PAROLE BASÉE SUR LA TECHNIQUE SÉPARATION DE SOURCE AVEUGLE

### 2.1 Introduction

La séparation aveugle de sources (SAS) a été initiée la première fois par Bernard Ans, Jeanny Hérault et Christian Jutten en 1985 [41], [42]. Dans leurs travaux sur la modélisation neurale depuis la SAS, demeure un sujet de recherche particulièrement actif et elle fait l'objet de nombreux travaux en traitement du signal et couvre un large nombre d'applications dans plusieurs domaines tels que la communication numérique [43], la reconnaissance de forme [44] et l'analyse des données économique [45], etc. Son objectif consiste à découpler des signaux inconnus qui ont été mélangés d'une manière inconnue sans se référer à aucune information ni sur le système de mélange ni sur les signaux sources sauf les propriétés structurelles des sources et/ou du système de mélange d'où l'utilisation du mot aveugle.

Les algorithmes de séparation de sources aveugles peuvent être classés selon différents critères : mélange instantané ou mélange convolutif, batch ou adaptatif, statistiques d'ordre deux ou d'ordre supérieur [46].

Dans le cadre de cette thèse, nous considérons le cas du mélange convolutif [47] et plus particulièrement dans l'application d'annulation du bruit et le rehaussement de la parole [48], [49], [50]. En s'intéressant exactement à la séparation aveugle de sources bi-capteurs entre le signal de parole et le bruit. Nous souhaitons extraire le signal de parole (source) à partir seulement des signaux bruités (observation) recueillis par les capteurs de mesure (microphone) [51], [52].

Dans ce qui suit, nous introduisant la problématique de rehaussement de la parole et ses applications. Puis nous allons présenter les deux types de mélange instantané et convolutif, nous présentons les deux méthodes (structures) de

séparation la plus utilisée dans le cas d'un mélange convolutif. Nous exposons aussi quelques algorithmes adaptatifs qui ont été proposés dans la littérature pour la séparation de sources bi-capteurs utilisée dans la structure directe.

## 2.2. Principes de la SAS dans le domaine de la réduction du bruit et le rehaussement de la parole

Dans la réalité l'acquisition d'un signal de parole est acquise dans un environnement bruité ce qui engendre la dégradation de la clarté et l'intelligibilité du signal parole, cette dégradation influe directement sur l'efficacité des techniques de traitement de la parole tel que la compression, la reconnaissance et la synthèse de la parole, d'où la nécessité de rehausser la parole afin d'augmenter la performance de ces techniques, pour ce faire, plusieurs méthodes de rehaussement de la parole ont été développées [53], il y a ceux basés sur la structure mono-capteur et ceux qui sont basés sur la méthode multi-capteurs, dans le cadre de cette thèse nous nous restreignons au rehaussement de parole combiné avec la séparation aveugle de source.

## 2.3 Les méthodes mono-capteur

La technique de rehaussement de parole mono-capteur est utilisée dans des situations où uniquement un seul canal d'acquisition est valable. Cette situation est imposée par certains systèmes tels que les applications de téléphonie ou bien dans les situations où le signal acoustique est préenregistré. La représentation générale de la technique du débruitage de parole mono-capteur est donnée par la figure 2.1. où on considère un signal d'observation,  $y$ , composé d'un signal de parole,  $s$ , corrompu par un *bruit additif*,  $b$ . Pour chaque indice temporel discret,  $n$ , le signal d'observation bruité,  $y$ , est donné par :

$$y(n) = s(n) + b(n) \tag{2.1}$$

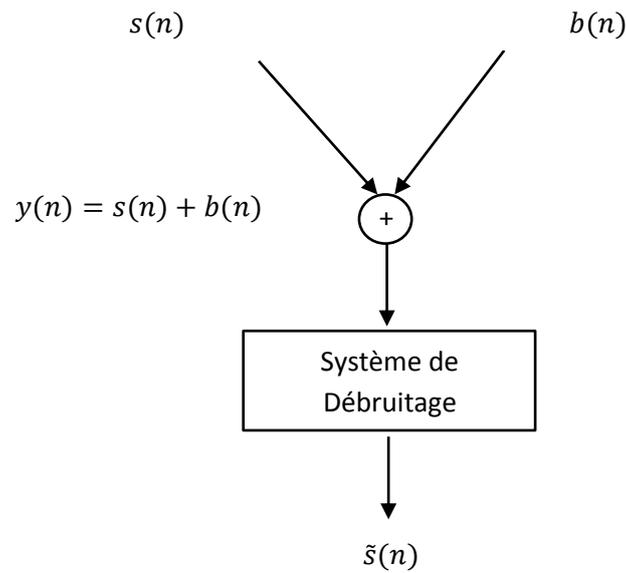


Figure 2.1 : Débruitage mono-captain.

Nous désignons par  $\tilde{s}(n)$ , le signal rehaussé en sortie qui est une estimation de  $s(n)$ .

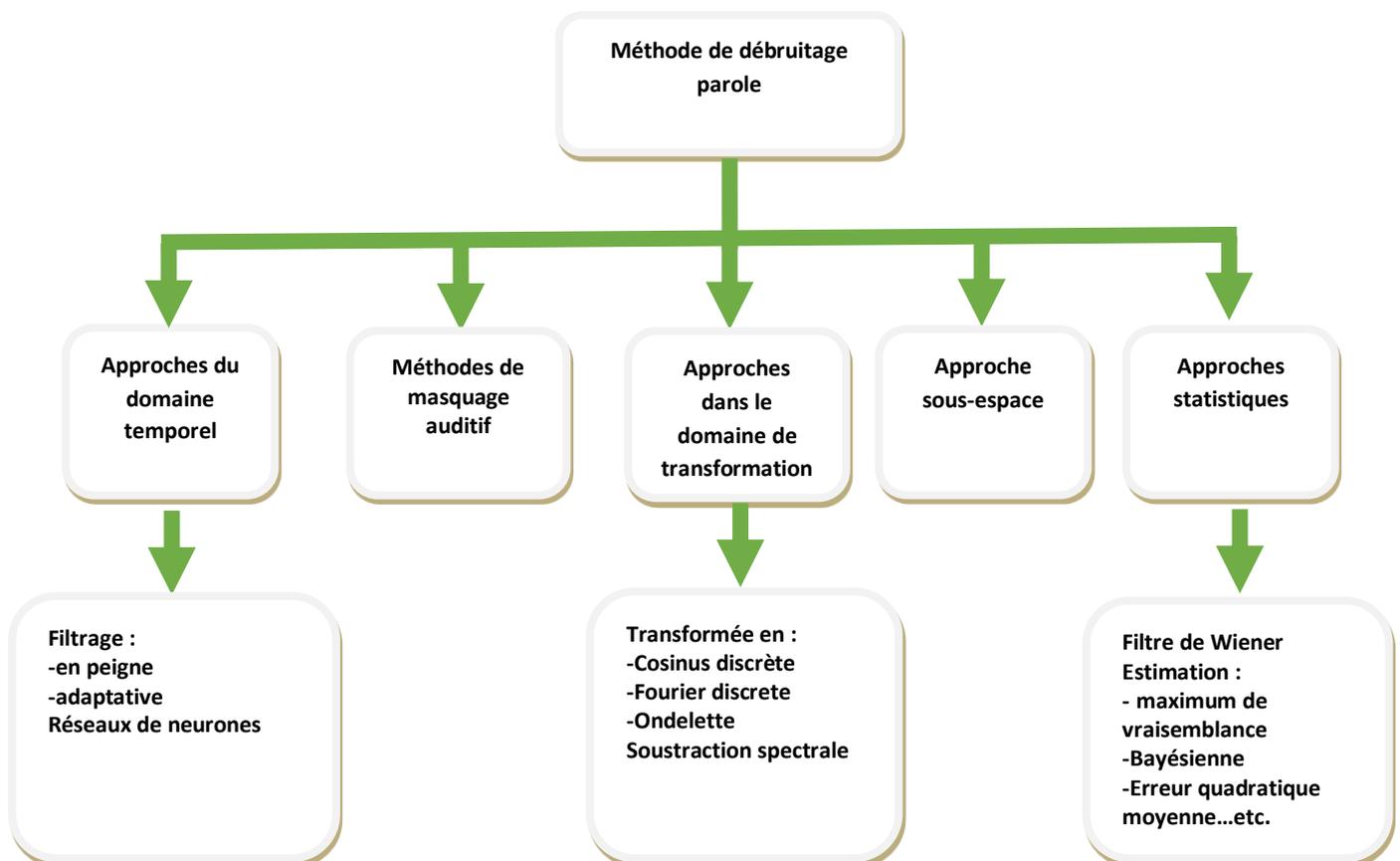


Figure 2.2 : Classification des méthodes de Débruitage de la parole

Plusieurs approches sont utilisées en littérature pour le Débruitage parole une alternative de classement de ces approches est illustrée dans la figure (2.2).

Par la suite en va passer en revue les algorithmes les plus utilisés dans la structure mono-capteur à savoir :

- Algorithmes de la soustraction spectrale
- Filtrage de Wiener

### 2.3.1 Algorithmes de la soustraction spectrale

La soustraction spectrale est la technique du débruitage mono-capteur la plus ancienne [54], elle consiste à la soustraction de l'estimé du bruit au signal dans le domaine spectral, cet estimé du bruit se fait sur plusieurs trames d'acquisition.

En utilisant Transformées de Fourier Discrètes (TFD) pour transformer les signaux du domaine temporel au domaine spectral l'équation (2.1) devient :

$$Y(v) = S(v) + B(v) \quad (2.2)$$

Et en module :

$$|Y(v)| = |s(v)| + |B(v)| \quad (2.3)$$

Où  $Y(v)$ ,  $S(v)$  et  $B(v)$  représentent les transformées de fourrier des signaux  $y(n)$ ,  $s(n)$  et  $b(n)$  respectivement,  $v$  représente la fréquence.

Donc pour effectuer cette soustraction on a besoin d'une estimé du bruit  $B(v)$  afin de pouvoir appliquer en sortie un algorithme de réduction de bruit sous l'hypothèse que ce bruit est stationnaire, non corrélé avec la trame d'analyse et que le signal de parole sur des trames d'analyse courte ( $\approx 25$  ms) peut être considéré comme stationnaire.

Ces hypothèses nous permettent d'écrire :

$$P_Y(v) = P_s(v) + P_B(v) \quad (2.4)$$

Où  $P_Y(v)$ ,  $P_s(v)$  et  $P_B(v)$  représente les densités spectrales de puissance (DSP) des signaux captés par le microphone  $y$ , le signal utile  $s$  et le bruit  $b$ .

Pour estimer la DSP du bruit plusieurs méthodes existent la plus simple consiste à considérer une trame suffisamment longue de l'ordre de 200 ms pour connaître le spectre moyen assimilé au bruit et des trames courtes de l'ordre de 8 ms du signal de parole la comparaison ensuite de ces deux aspects permet d'estimer le bruit des trames à l'entrée du signal, ensuite le travail reste à effectuer une soustraction spectrale et essayer d'atténuer le bruit, cette soustraction peut être en puissance [55],[56] ou en amplitude [57].

Dans la soustraction spectrale d'amplitude et après avoir estimé le bruit, on effectue une soustraction, l'équation (2.3) devient :

$$|\tilde{s}(v)| = |Y(v)| - |\tilde{B}(v)| \quad (2.5)$$

$|\tilde{s}(v)|$  et  $|\tilde{B}(v)|$  représentent l'estimées de l'amplitude du signal utile  $s$  et du bruit  $b$ . Et si l'estimée du bruit est donnée par la DSP on peut écrire :

$$P_{\tilde{s}}(v) = P_Y(v) - P_{\tilde{B}}(v) \quad (2.6)$$

Si nous supposons que le bruit et le signal sont indépendants, alors on peut utiliser le spectre d'énergie au lieu de l'amplitude, et dans ce cas on peut exprimer l'énergie du signal bruité selon l'équation:

$$|\tilde{s}(v)|^2 = |Y(v)|^2 - |\tilde{B}(v)|^2 \quad (2.7)$$

Ou  $P_{\tilde{s}}(v)$  et  $P_{\tilde{B}}(v)$  représentent les estimées de la densités spectrales de puissances (DSP) du signal utile  $s$  et du bruit  $b$ .

Par la suite on utilise une  $TFD^{-1}$  pour revenir au domaine temporel. La phase du signal d'entrée est gardée, du fait que notre oreille est peu sensible aux variations de la phase d'une part et une estimation de la phase du bruit est très compliquée à réaliser.

La version de base de la soustraction spectrale peut être améliorée du fait que les deuxièmes termes des équations (2.5) et (2.7) peuvent prendre une valeur négative, cette amélioration consiste soit à le rendre positif en changeant de signe ou bien en l'annulant comme dans l'équation (2.8).

$$|\tilde{X}(f)|^2 = \begin{cases} |Y(f)|^2 - |\tilde{B}(f)|^2 & \text{si } |Y(f)|^2 > |\tilde{B}(f)|^2 \\ 0 & \text{sinon} \end{cases} \quad (2.8)$$

On trouve aussi d'autres variantes de la version classique telle que la soustraction spectrale par filtrage qui consiste à utiliser un filtre, et en gardant l'estimation du bruit, on peut assimiler la soustraction spectrale à un filtrage.

La réduction du bruit par soustraction spectrale est très utilisée du fait qu'elle est très simple à implémenter, cependant elle a l'inconvénient de générer des artefacts après la réduction du bruit ainsi qu'une distorsion du signal et le bruit musical.

Dans la littérature, on trouve plusieurs améliorations pour remédier à ces inconvénients, on cite les améliorations de Berouti [58], Boll [54] et ceux basés sur les aspects psycho acoustiques pour faire face au bruit musical [59],[60],[61].

### 2.3.2 Filtrage de Wiener

Le filtrage de Wiener est parmi les approches les plus anciennes utilisées dans le débruitage de la parole [62], cette approche s'appuie sur l'addition des densités spectrales de puissance de l'équation (2.4) et consiste à trouver un filtre linéaire  $h(n)$  de telle sorte qu'il minimise l'erreur quadratique moyenne (EQM) entre le signal d'entrée et celui en sortie sous les hypothèses que les signaux  $s(n)$  et  $b(n)$  sont des signaux stationnaires et non corrélés dans la séquence d'analyse [62] :

$$\varepsilon(n) = s(n) - \tilde{s}(n) \quad (2.9)$$

donné par :

$$\tilde{s}(n) = h(n) * y(n) \quad (2.10)$$

Où (\*) représente l'opérateur de convolution et  $\tilde{s}(n)$  l'estimé du signal de parole

Si nous passons au domaine fréquentiel l'équation (2.9), devient

$$\varepsilon(v) = S(v) - H(v).Y(v) \quad (2.11)$$

L'erreur quadratique moyenne (EQM) est donnée par :

$$E[|\varepsilon(v)|^2] = E[|S(v) - H(v).Y(v)|^2] \quad (2.12)$$

Déterminer ce filtre de Wiener peut être considéré comme un problème d'optimisation, et selon le théorème de projection, il y a une solution unique pour l'équation précédente. Elle est donnée en utilisant le principe d'orthogonalité par l'équation :

$$E[\varepsilon(v).Y(v)] = 0 \quad (2.13)$$

Le filtre de Wiener est donné dans le domaine spectral en fonction des densités spectrales de puissance de la parole et du bruit comme suit :

$$H(v) = \frac{P_x(v)}{P_x(v)+P_b(v)} \quad (2.14)$$

Le filtre de Wiener estimé sous les hypothèses de la stationnarité et la non-corrélation parvient à réduire le bruit considérablement sans une significative distorsion, les densités spectrales de puissance peuvent être estimées en prenant la moyenne à travers plusieurs trames d'analyse, lorsque les hypothèses de la stationnarité et la non-corrélation ne sont pas satisfaites dans le sens où les densités spectrales de puissance du signal désiré et le bruit varient dans le temps, dans ce cas ces densités ne peuvent pas être estimées de la même manière et on est obligé à estimer le filtre de Wiener pour chaque trame séparément [63].

Nous notons que plusieurs améliorations ont été apportées à la technique de débruitage via le filtre de Wiener, dans [64] les auteurs ont proposé d'estimer la densité spectrale de puissance du signal de parole et de procéder à un calcul itératif du filtre de Wiener basée sur une modélisation LPC du signal de parole. La mise à jour de la densité spectrale de puissance est faite à chaque itération et déduite à partir des coefficients AR, d'autres proposent de réduire le bruit résiduel après l'utilisation du filtre de Wiener en particulier le bruit musical en ajoutant un second filtre considéré comme facteur de pondération à motivation psycho acoustique [65].

Dans [66] l'auteur, propose d'utiliser le filtre de Wiener dans le domaine temps fréquence, le rehaussement de parole est effectué sur la base de la prédiction linéaire du signal résiduel.

## 2.4 Les méthodes multi-capteurs

Les algorithmes de débruitage parole mono-capteur basé sur les signaux enregistrés sont rencontrés fréquemment, ils sont très efficaces généralement dans le cas où le signal parole est corrompu par un bruit stationnaire or ce n'est pas toujours le cas du fait que la parole est souvent corrompue par des bruits dont la puissance varie ou le signal bruité contient de la réverbération significative ce qui engendre la dégradation de la performance des méthodes mono-capteur classiques, et cela bien qu'il y ait plusieurs algorithmes dédiés à l'annulation des bruits non stationnaires tels que les algorithmes basés sur les détecteurs d'activités vocaux [67],[68] et le minimum statistique [69] ,[70]. Les méthodes multi-capteurs ou plusieurs microphones sont utilisés restent une bonne alternative dans le but d'annuler les bruits et faire face au problème de non-stationnarité moyennant des algorithmes adaptatifs.

Les méthodes multi-capteurs nécessitent des informations a priori sur l'environnement acoustique et la nature des sources en plus pour avoir de bonnes performances l'utilisation d'un grand nombre de microphones est nécessaire ce qui est difficile et coûteux lors de l'implémentation.

Une des solutions les plus efficaces pour alléger le cout et facilite l'implémentation tout en réduisant le nombre de microphones consiste à utiliser les algorithmes de séparation aveugle de sources. Ces derniers exploitent les informations spatiales des signaux enregistrés depuis plusieurs microphones pour arriver à les séparer sans avoir à recourir à des informations a priori ni sur le système de mélange ni sur les signaux sources.

### 2.4.1 Le principe de la séparation aveugle de sources

Le but du SAS est de reconstituer les  $N$  sources uniquement à partir des  $M$  sources observées constituées de mélanges des sources sans avoir aucune information sur le système de mélange où les sources sont inobservables avec l'hypothèse qu'elles sont indépendantes, la séparation aveugle de sources possède plusieurs degrés de difficulté selon le type du système de mélange considéré dans la littérature on trouve

différentes approches pour résoudre le problème de SAS selon le type de mélange linéaire [71] ou non linéaire [72], mélange instantané ou convolutif [73], modèle de mélange surdéterminé [74] ou sous-déterminé [75], dans le cadre de cette thèse, nous allons nous restreindre aux mélanges linéaires convolutifs invariants entre le signal de parole et le bruit.

#### 2.4.2 Mélange instantané

C'est le type de mélange le plus simple dans lequel les signaux sources arrivent en même temps sur tous les microphones quelles que soient, les distances qui les séparent, mais avec des intensités différentes (voir figure 2.3).

Dans la séparation aveugle de source multi-capteurs on dispose d'une séquence d'observation  $\mathbf{p}(n)$ , ces observations sont une combinaison linéaire de  $N$  sources  $\mathbf{s}(n)$  :

$$\mathbf{p}(n) = \mathbf{A} \cdot \mathbf{s}(n) + \boldsymbol{\eta}(n) \quad (2.15)$$

Où  $\mathbf{A}$  est une matrice  $M \times N$  et  $\boldsymbol{\eta}(n)$  est un vecteur d'un bruit additif.

Dans le cas d'un mélange instantané et linéaire, l'équation (2.15) peut être donnée sous la forme:

$$p_i(n) = \sum_{j=1}^N a_{ji} \cdot s_j(n) + \eta_i(n) \text{ avec } i \in [1, 2, 3, \dots, M] \quad (2.16.a)$$

Où  $a_{ij}$  sont des coefficients scalaires du mélange linéaire instantané.

Si on prend en considération les différentes distances entre les sources et les microphones dans lesquels les signaux sources arrivent à chaque capteur à des instants différents, donc chaque source est caractérisée par un délai entre les canaux. Ce type de mélange est appelé mélange anéchoïque [76].

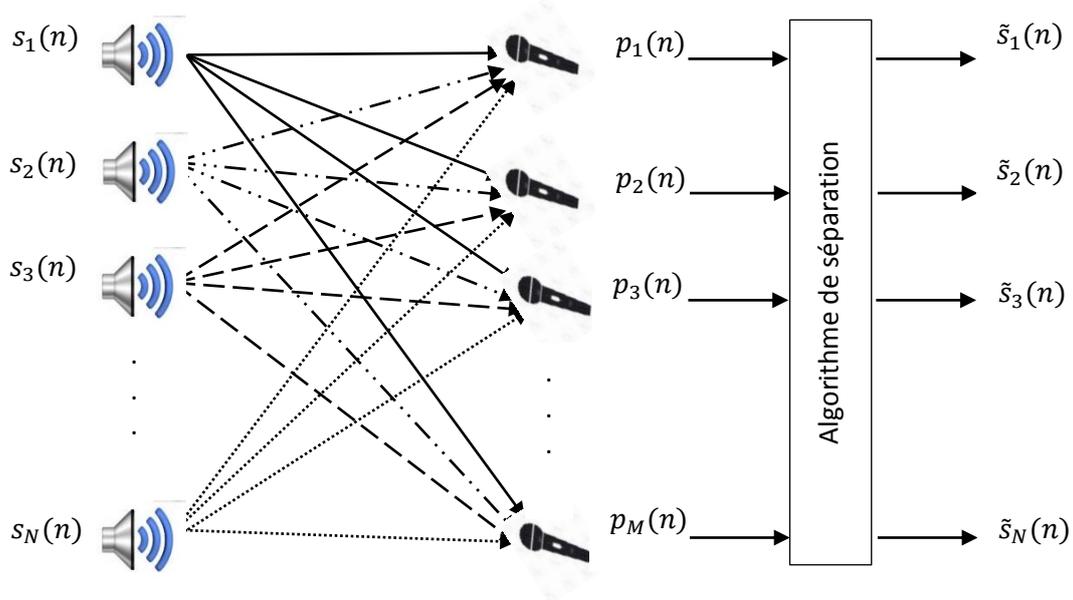


Figure 2.3 : Schéma du mélange instantané

Les relations entre les observations et les sources dans un mélange anéchoïque sont données par :

$$p_i(n) = \sum_{j=1}^N a_{ji} \cdot s_j(n - \tau_{ji}) + \eta_i(n) \quad \text{avec } i \in [1, 2, 3, \dots, M] \quad (2.16.b)$$

Où  $\tau_{ji}$  représentent les retards entre les sources et les capteurs.

### 2.4.3 Mélange convolutif

Dans le cas où les enregistrements se font dans un environnement acoustique fermé, le mélange instantané devient incomplet et ne reflète pas la réalité du fait que les sources se mélangent convolutivement où on prend en considération les délais de transmission entre les sources et les capteurs d'une part et les réflexions des signaux sur les obstacles rencontrés lors de la propagation d'autre part (figure 2.4).

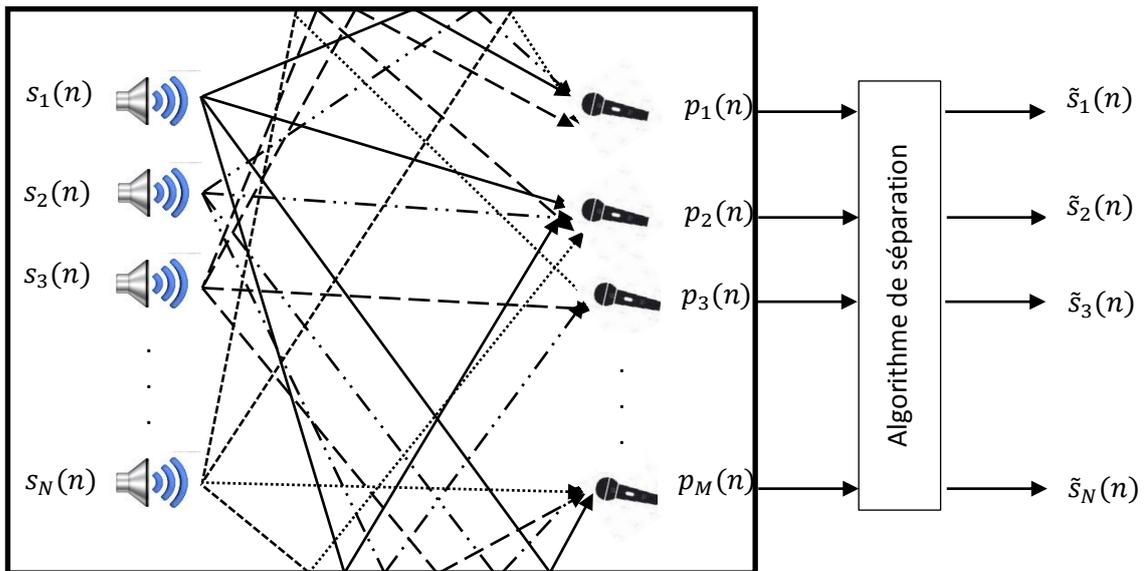


Figure 2.4 : Schéma du mélange convolutif

Pour modéliser un mélange convolutif dans le domaine temporel, l'équation (2.16) devient [49] :

$$p_i(n) = \sum_{j=1}^N \sum_{k=0}^{L-1} h_{ji}(k) \cdot s_j(n-k) + \eta_i(n) \quad \text{avec } i \in [1,2,3, \dots, M] \quad (2.17)$$

Où  $h_{ji}$  définissent les opérations de transformation réalisées par le canal de propagation respectivement sur la source  $i$  se propageant vers le capteur,  $j$ , où  $L$  est la longueur du filtre supposé à réponse impulsionnelle finie figure (2.5).

Plusieurs solutions ont été proposées pour résoudre le problème de séparation de sources de mélange convolutif dans le domaine temporel, les premiers travaux sont inspirés par les méthodes de déconvolution aveugle où l'auteur a modélisé le procédé de séparation comme un filtre à réponse impulsionnelle finie (RIF) dont il faut estimer les coefficients  $h_{ji}$  [44],[77], dans [78] les auteurs ont modélisé la procédure de séparation comme un filtre à réponse impulsionnelle infinie (RII) en supposant que l'environnement de l'enregistrement est à phase minimale, d'autres auteurs ont étudié le problème dans le domaine fréquentiel en utilisant la

transformée de Fourier à court terme [79],[80], d'autres auteurs ont traité la séparation dans le domaine temps fréquence pour transformer le produit de convolution en multiplication [73] [81],[82],[83],[84].

Dans le cadre de ce travail, nous allons traiter le problème de l'annulation du bruit acoustique via les méthodes de séparation de source bi-capteurs, c'est-à-dire le cas particulier constitué de deux sources, une source de parole perturbée avec une source de bruit, ces deux sources sont mélangées convolutivement.

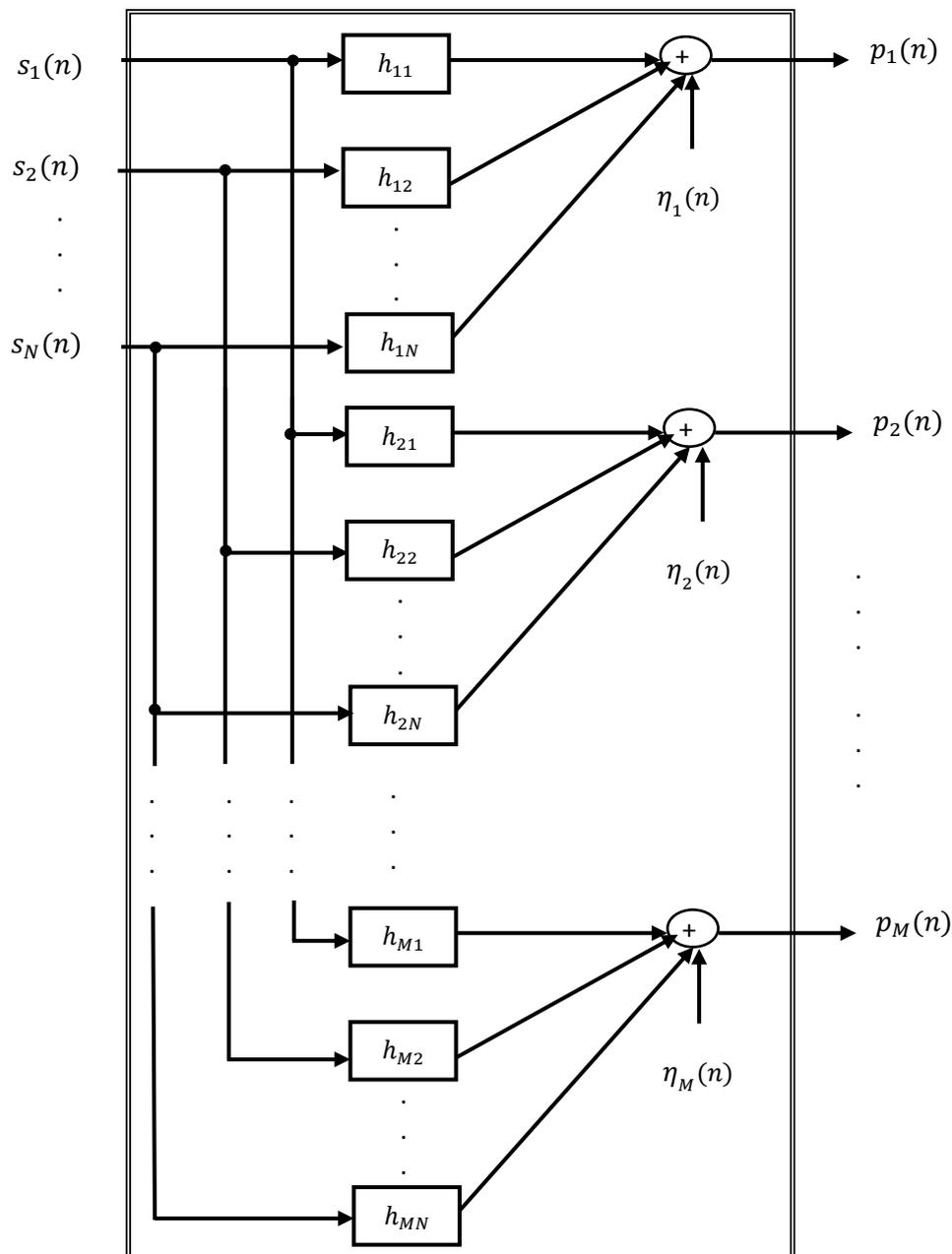


Figure 2.5 : Modèle du mélange convolutif

## 2.5 Mélange linéaire convolutif bi-capteurs

Le mélange linéaire convolutif bi-capteurs est un cas particulier du mélange linéaire convolutif multi-capteurs où on considère qu'on a deux sources et deux observations (mélange) figure 2.6.

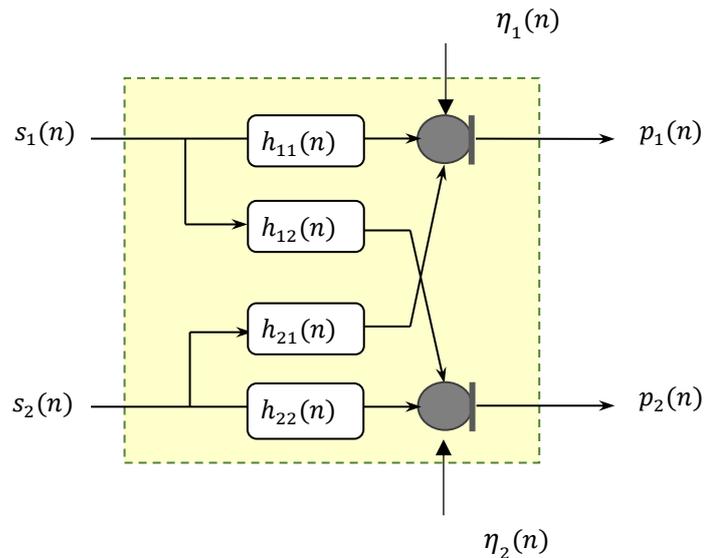


Figure 2.6 : Modèle du mélange convolutif bi-capteurs

Les deux observations sont la superposition des signaux originaux qui sont inconnus, le mélange est supposé être un mélange linéaire convolutif des signaux en pleine bande, qui dépend de la propagation des signaux dans le milieu, de la position des capteurs et des sources, et des caractéristiques du milieu. Les équations des signaux observés à la sortie de ce mélange sont obtenues à partir de l'équation (2.17), en mettant  $N = M = 2$  :

$$p_i(n) = \sum_{j=1}^2 h_{ji} \cdot s_j(n - \tau_{ji}) + \eta_i(n) \quad \text{avec } i \in [1,2] \quad (2.18)$$

Ou bien si on introduit l'opérateur de convolution :

$$p_1(n) = h_{11}(n) * s_1(n) + h_{21}(n) * s_2(n) + \eta_1(n) \quad (2.19)$$

$$p_2(n) = h_{12}(n) * s_1(n) + h_{22}(n) * s_2(n) + \eta_2(n) \quad (2.20)$$

où  $h_{11}(n)$  et  $h_{22}(n)$  représentent les réponses impulsionnelles des canaux directs de couplage,  $h_{12}(n)$  et  $h_{21}(n)$  représentent les effets du couplage croisé entre les sources et les capteurs,  $s_1(n)$  et  $s_2(n)$  représentent les signaux sources,  $p_1(n)$  et  $p_2(n)$  représentent respectivement les deux signaux observations obtenus à la sortie du mélange convolutif bi-capteurs.  $\eta_1(n)$  et  $\eta_2(n)$  sont des bruits additifs au niveau des microphones, le symbole  $(*)$  représente l'opérateur de convolution.

Le modèle de mélange linéaire convolutif bi-capteurs complet de la figure 2.5 peut être simplifié considérablement en prenant en compte les suppositions suivantes [85],[86] :

- Les deux sources représentent un signal de parole et un signal de bruit c'est-à-dire :  

$$s_1(n) = s(n), s_2(n) = b(n)$$
- En plus, les deux sources sont considérées statistiquement indépendantes,
- Les deux microphones sont placés judicieusement près des deux sources ce qui permet de dire que :  

$$h_{11}(n) = h_{22}(n) = \delta(n).$$
- La distance entre les deux microphones est petite.
- L'erreur de mesure de chaque microphone est négligeable c'est-à-dire :  

$$\eta_1(n) = \eta_2(n) = 0$$
- Les filtres sont supposés des filtres RIF

Toutes ces hypothèses [44][51],[79],[87],[88],[89],[90] sont requises et permettent de simplifier considérablement le problème de l'annulation du bruit acoustique via les méthodes de séparation aveugle de sources, le modèle de mélange linéaire convolutif bi-capteurs simplifiés est donné par la figure 2.6.

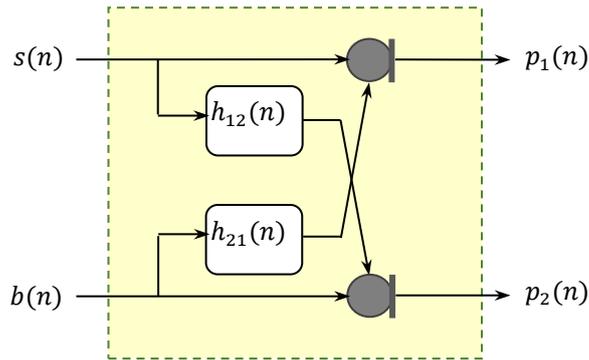


Figure 2.7 : Modèle simplifié d'un mélange convolutif bi-capteurs [89],[90].

Dans le modèle simplifié les deux observations mesurées sont données par :

$$p_1(n) = s(n) + b(n) * h_{21}(n) \quad (2.21)$$

$$p_2(n) = b(n) + s(n) * h_{12}(n) \quad (2.22)$$

## 2.6. Méthode de séparation de sources dans un mélange convolutif bi-capteurs

Deux structures ont été proposées afin d'extraire les signaux originaux  $s(n)$  et  $b(n)$  à partir des deux observations  $p_1(n)$  et  $p_2(n)$  en utilisant la technique de séparation aveugle de source : la structure directe "Forward" (FBSS), et la structure récursive "Backward" (BBSS).

### 2.6.1. Structure directe (Forward)

La structure directe (FBSS : Forward Blind Source Separation) est la structure la plus utilisée dans le domaine d'annulation du bruit acoustique elle permet d'estimer les deux signaux originaux à partir des observations mesurées au niveau des capteurs sans aucune information ni sur le mélange ni sur les signaux sources [51],[52], elle est donnée par la figure 2.8.

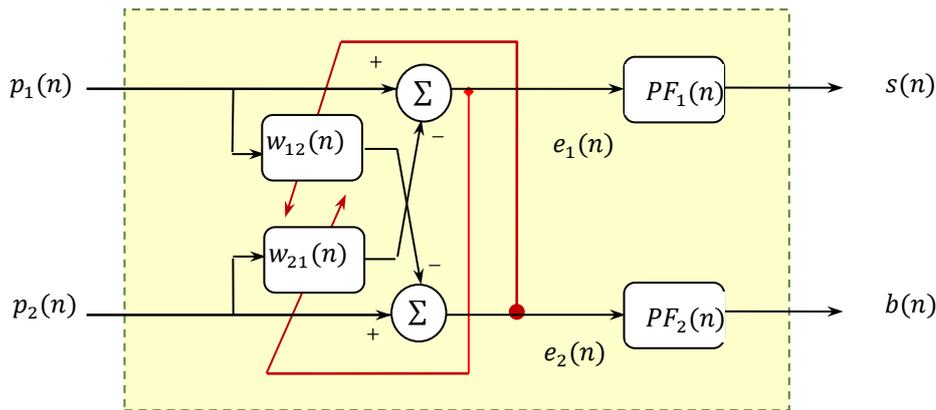


Figure 2.8 : Structure directe « Forward BSS »

La structure directe est basée sur deux filtres adaptatifs symétriques  $w_{12}(n)$  et  $w_{21}(n)$  qui servent à identifier respectivement les deux réponses impulsionnelles  $h_{12}(n)$  et  $h_{21}(n)$ , les deux signaux obtenus à la sortie de processus de séparation par la structure directe sont donnés par :

$$e_1(n) = p_1(n) - p_2(n) * w_{21}(n) \quad (2.23)$$

$$e_2(n) = p_2(n) - p_1(n) * w_{12}(n) \quad (2.24)$$

En remplaçant les deux équations (2.21) et (2.22) dans les deux équations (2.23) et (2.24), nous obtenons :

$$e_1(n) = b(n) * [h_{21}(n) - w_{21}(n)] + s(n) * [\delta(n) - h_{12}(n) * w_{21}(n)] \quad (2.25)$$

$$e_2(n) = s(n) * [h_{12}(n) - w_{12}(n)] + b(n) * [\delta(n) - h_{21}(n) * w_{12}(n)] \quad (2.26)$$

La solution théorique est obtenue en utilisant l'hypothèse d'optimalité pour les deux filtres adaptatifs  $w_{12}(n) = h_{12}(n)$  et  $w_{21}(n) = h_{21}(n)$ , [85], donc les équations (2.25) et (2.26) deviennent :

$$e_1(n) = s(n) * [\delta(n) - h_{12}(n) * w_{21}(n)] \quad (2.27)$$

$$e_2(n) = b(n) * [\delta(n) - h_{21}(n) * w_{12}(n)] \quad (2.28)$$

Les signaux de sortie donnés par les expressions (2.27) et (2.28) représentent une solution partielle du problème de séparation à cause de la présence d'une distorsion significative, la solution complète consiste à ajouter deux post-filtres  $PF_1(n)$  et  $PF_2(n)$  à la sortie de la structure directe (voir la figure 2.8), leur expression est donnée par :

$$PF_1(n) = PF_2(n) = [\delta(n) - w_{12}(n) * w_{21}(n)]^{-1} \quad (2.29)$$

Aux la sorties des deux post-filtres  $PF_1(n)$  et  $PF_2(n)$ , les deux signaux de sortie se rapprochent des signaux sources  $s(n)$  et  $b(n)$ .

Plusieurs approches automatiques en domaine fréquentiel ont été proposées par Djendi et all [85],[91] qui consiste à récupérer le signal de parole original et atténuer la distorsion qui se trouve dans les deux signaux de la sortie de la structure Forward sans l'utilisation des post-filtres.

### 2.6.2. Structure réursive (Backward)

La structure réursive (BBSS : Backward Blind Source Separation) est la deuxième architecture utilisée dans la séparation aveugle de source, la figure (2.9) représente le schéma équivalent de la structure BBSS bi-capteurs [92],[93],[94].

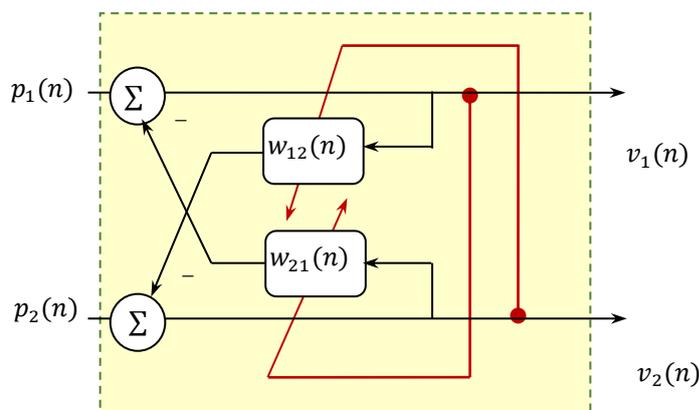


Figure 2.9 : Structure réursive (Backward BSS)

A la sortie du processus de séparation (Figure 2.9), le signal estimé du premier canal  $v_1(n)$  est obtenu par soustraction du mélange  $p_1(n)$  et la sortie du deuxième filtre  $w_{21}(n)$ , tandis que le signal estimé du deuxième canal est  $v_2(n)$  est obtenue par soustraction du mélange  $p_2(n)$  et la sortie du deuxième filtre  $w_{12}(n)$ .

Donc, les deux relations d'entrée-sortie de la structure récursive sont données par :

$$v_1(n) = p_1(n) - v_2(n) * w_{21}(n) \quad (2.30)$$

$$v_2(n) = p_2(n) - v_1(n) * w_{12}(n) \quad (2.31)$$

En remplaçant les deux équations (2.21) et (2.22) dans les deux équations (2.30) et (2.31), nous obtenons :

$$v_1(n) = [b(n) * (h_{21}(n) - w_{21}(n)) + s(n) * (\delta(n) - h_{12}(n) * w_{21}(n))] * (\delta(n) - w_{12}(n) * w_{21}(n))^{-1} \quad (2.32)$$

$$v_2(n) = [s(n) * (h_{12}(n) - w_{12}(n)) + b(n) * (\delta(n) - h_{21}(n) * w_{12}(n))] * (\delta(n) - w_{21}(n) * w_{12}(n))^{-1} \quad (2.33)$$

Comme dans le cas de la structure directe, la solution est obtenue en utilisant l'hypothèse d'optimalité pour les deux filtres adaptatifs  $w_{12}(n) = h_{12}(n)$  et  $w_{21}(n) = h_{21}(n)$ , et par conséquent les deux signaux de sorties  $v_1(n)$  et  $v_2(n)$  peuvent être donné par :

$$v_1(n) = s(n) \quad (2.34)$$

$$v_2(n) = b(n) \quad (2.35)$$

Il est intéressant de signaler que dans la structure (Backward), le signal de parole estimé  $v_1(n)$  est égale exactement au signal original  $s(n)$  sans distorsion. Par contre dans la structure directe (Forward) le signal de parole estimé  $e_1(n)$  est égale au signal de parole original  $s(n)$  modifié par une quantité,  $(\delta(n) - w_{12}(n) * w_{21}(n))$ . D'où l'obligation d'utiliser des post-filtres à la sortie de cette structure ou bien estimer les PFs par les deux structures proposées dans [85],[91].

On note aussi d'après les travaux réalisés dans [42] que la distorsion peut être considérée comme minime dans le cas où la distance qui sépare les deux capteurs est considérée comme petite et par conséquent, on peut se passer de l'utilisation des postes filtres, dans ce cas les signaux estimés peuvent être donnés par :

$$e_1(n) = s(n) \quad (2.36)$$

$$e_2(n) = b(n) \quad (2.37)$$

Dans le cadre de cette thèse, on a préféré d'utiliser la structure Forward pour plusieurs raisons on cite [94] :

- L'estimation ne dépend pas de la sortie.
- Simplicité du calcul.
- Le système est toujours causal, car on ne fait pas intervenir les différentes sorties du système.
- Stabilité de l'algorithme.

## 2.7 État de l'art sur les algorithmes adaptatifs utilisés dans la structure Forward

Pour résoudre le problème d'annulation du bruit acoustique et le rehaussement de la parole, plusieurs travaux de séparation aveugle de sources bi-capteurs ont été développés en utilisant les algorithmes du filtrage adaptatif à travers les deux structures (Forward et Backward), ces algorithmes ont pour but d'identifier les deux filtres symétriques  $w_{12}(n)$ ,  $w_{21}(n)$  et par conséquent nous arrivons à estimer le signal de parole  $s(n)$  à partir des deux observations  $p_1(n)$  et  $p_2(n)$ .

Dans la suite de ce chapitre, nous allons présenter deux algorithmes du filtrage adaptatifs appliqués à la structure directe de séparation de sources bi-capteurs, ces deux algorithmes sont utilisés dans la comparaison avec un algorithme adaptatif à base des Métaheuristiques.

### 2.7.1 Algorithme de gradient stochastique double (2C-FLMS)

Parmi les algorithmes les plus utilisés dans le domaine du filtrage adaptatif, on cite l'algorithme de gradient stochastique (LMS) [24],[29], ce dernier est parmi les

premiers algorithmes adaptatifs utilisés dans la séparation aveugle de sources [95]. Il est sert à mettre à jour les deux filtres adaptatifs symétriques  $w_{12}(n)$  et  $w_{21}(n)$  en même temps d'où l'appellation algorithme de gradient stochastique double (2C-FLMS : Two-Channel Forward LMS).

L'implémentation de cet algorithme dans la structure Forward sans l'utilisation de post filtre est très simple. Dans la figure 2.10, nous présentons le modèle du mélange convolutif et l'adaptation des deux filtres de séparation de sources Forward.

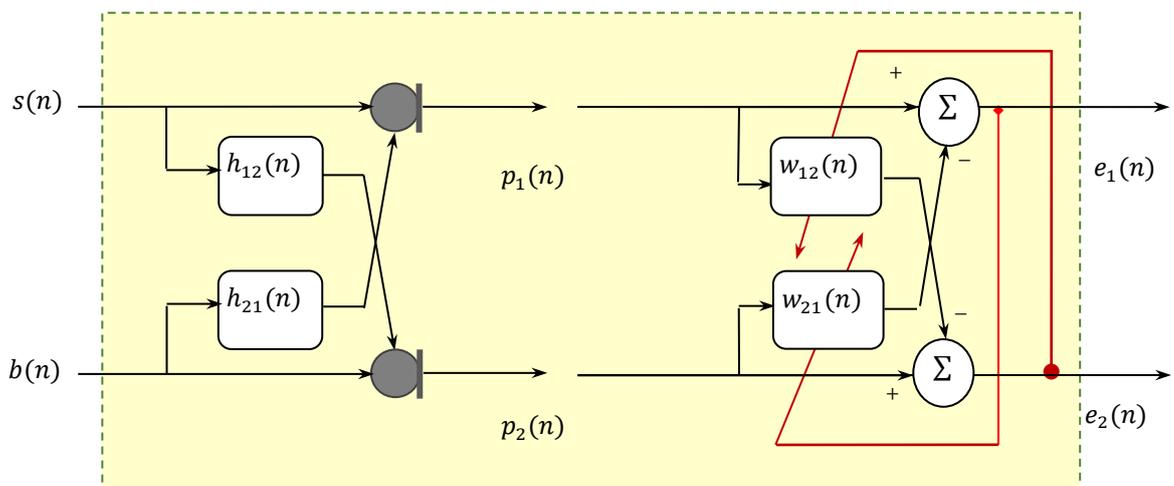


Figure 2.10 : Structure complète d'un mélange convolutif et la structure directe.

Les équations de mise à jour des coefficients des filtres adaptatifs  $w_{12}(n)$  et  $w_{21}(n)$  à l'aide de l'algorithme de gradient stochastique double (2C-FLMS : Two-Channel Forward LMS) sont données par :

$$\mathbf{w}_{12}(n) = \mathbf{w}_{12}(n-1) + \mu_{12} \mathbf{p}_1(n)e_2(n) \quad (2.38)$$

$$\mathbf{w}_{21}(n) = \mathbf{w}_{21}(n-1) + \mu_{21} \mathbf{p}_2(n)e_1(n) \quad (2.39)$$

L'algorithme 2C-FLMS est résumé dans le tableau 2.1.

Tableau 2.1 Algorithme 2C-LMS [29]

Détail de calcul	Variabes
<b>1. Initialisation</b> $w_{12}(0) = 0, w_{21}(0) = 0$	$L$ : Longueur du filtre adaptatif. $k$ : nombre d'itérations.
<b>2. for <math>n = 1:k</math> do</b> <b>Estimation des sorties</b>	$\mathbf{w}_{12}$ : Filtre adaptatif (FIR) : $\mathbf{w}_{12}(n) = [w_{12,0}(n), w_{12,1}(n), \dots, w_{12,L-1}(n)]^T$
<b>3. <math>e_1(n) = p_1(n) - \mathbf{w}_{21}^T(n-1)\mathbf{p}_2(n)</math></b> <b>4. <math>e_2(n) = p_2(n) - \mathbf{w}_{12}^T(n-1)\mathbf{p}_1(n)</math></b>	$\mathbf{w}_{21}$ : Filtre adaptatif (FIR) : $\mathbf{w}_{21}(n) = [w_{21,0}(n), w_{21,1}(n), \dots, w_{21,L-1}(n)]^T$
<b>Adaptation des coefficients des filtres</b>	$\mathbf{p}_1(n) = [p_1(n), p_1(n-1), \dots, p_1(n-L+1)]^T$ $\mathbf{p}_2(n) = [p_2(n), p_2(n-1), \dots, p_2(n-L+1)]^T$
<b>5. <math>\mathbf{w}_{12}(n) = \mathbf{w}_{12}(n-1) + \mu_{12}\mathbf{p}_1(n)e_2(n)</math></b> <b>6. <math>\mathbf{w}_{21}(n) = \mathbf{w}_{21}(n-1) + \mu_{21}\mathbf{p}_2(n)e_1(n)</math></b>	$\mu_{12}$ : le pas d'adaptation : $0 < \mu_{12} < 2/\lambda_{max}$ $\mu_{21}$ : le pas d'adaptation : $0 < \mu_{21} < 2/\lambda_{max}$
<b>7. Fin</b>	

### 2.7.2 Algorithme de gradient stochastique normalisé double (2C-FNLMS)

Le gradient stochastique normalisé (NLMS) est la variante la plus utilisée du LMS elle consiste à normaliser le pas d'adaptation avec l'énergie du signal d'entrée [24], le NLMS pour être implémenté de la même manière que le LMS dans la structure Forward, dans ce cas la mise à jour des coefficients des filtres adaptatifs en utilisant l'algorithme de gradient stochastique normalisé double (2C-NLMS : Two-Channel NLMS) est donnée par les équations [52],[94] :

$$\mathbf{w}_{12}(n) = \mathbf{w}_{12}(n-1) + \mu_{12}e_2(n) \left( \frac{\mathbf{p}_1(n)}{\xi + \|\mathbf{p}_1(n)\|^2} \right) \quad (2.40)$$

$$\mathbf{w}_{21}(n) = \mathbf{w}_{21}(n-1) + \mu_{21}e_1(n) \left( \frac{\mathbf{p}_2(n)}{\xi + \|\mathbf{p}_2(n)\|^2} \right) \quad (2.41)$$

L'algorithme 2C-NLMS est résumé sur le tableau 2.2.

Tableau 2.2 Algorithme 2C-FNLMS [52]

Détail de calcul	Variables
<b>1. Initialisation</b> $w_{12}(0) = 0, w_{21}(0) = 0$	$L$ : Longueur du filtre adaptatif. $k$ : nombre d'itérations.
<b>2. for</b> $n = 1:k$ <i>do</i> <b>Estimation des sorties</b>	$w_{12}$ : Filtre adaptatif (FIR) : $w_{12}(n) = [w_{12,0}(n), w_{12,1}(n), \dots, w_{12,L-1}(n)]^T$
<b>3.</b> $e_1(n) = p_1(n) - w_{21}^T(n-1)p_2(n)$ <b>4.</b> $e_2(n) = p_2(n) - w_{12}^T(n-1)p_1(n)$	$w_{21}$ : Filtre adaptatif (FIR) : $w_{21}(n) = [w_{21,0}(n), w_{21,1}(n), \dots, w_{21,L-1}(n)]^T$
<b>Adaptation des coefficients des filtres</b>	$p_1(n) = [p_1(n), p_1(n-1), \dots, p_1(n-L+1)]^T$ $p_2(n) = [p_2(n), p_2(n-1), \dots, p_2(n-L+1)]^T$
<b>5.</b> $w_{12}(n) = w_{12}(n-1) + \mu_{12}e_2(n) \left( \frac{p_1(n)}{\xi + \ p_1(n)\ ^2} \right)$	$\mu_{12}$ : Pas d'adaptation : $0 < \mu_{12} < 2$
<b>6.</b> $w_{21}(n) = w_{21}(n-1) + \mu_{21}e_1(n) \left( \frac{p_2(n)}{\xi + \ p_2(n)\ ^2} \right)$	$\mu_{21}$ : Pas d'adaptation : $0 < \mu_{21} < 2$
<b>7. Fin</b>	$\xi$ : Nombre positif suffisamment petit

### 2.7.3 Algorithme Forward à pas d'adaptations variables (2C-VSSF)

L'algorithme de gradient stochastique normalisé double (2C-FNLMS) a été utilisé avec succès dans l'annulation du bruit acoustique, son inconvénient majeur reste le conflit entre la rapidité de la convergence et le faible désajustement. Plusieurs d'algorithmes ont été proposés pour traiter ce compromis, ces alternatives sont essentiellement basées sur l'algorithme de gradient stochastique normalisé. Dans cette thèse, on va passer en revue l'algorithme Forward à pas d'adaptations variables (2C-VSSF : Two-Channel Variable Step-Sizes Forward) [94], qui d'après les résultats théoriques obtenus en l'employant dans une structure de séparation directe à montrer son efficacité en matière de convergence et faible désajustement, ce qui influe positivement sur la qualité du signal rehaussé.

Le 2C-VSSF consiste à utiliser des pas d'adaptation variables  $\mu_{12}(n)$ ,  $\mu_{21}(n)$ , ces pas d'adaptation sont contrôlés à leur tour par le critère de décorrélation entre les deux signaux estimés à la sortie de la structure Forward  $e_1(n)$ ,  $e_2(n)$ , Les équations mathématiques de l'algorithme 2C-VSSF sont résumées dans le tableau 2.3 :

Tableau 2.3 Algorithme 2C-VSSF [94]

Détail du calcul	Variables
<p>1. <b>Initialisation</b>  <math>w_{12}(0) = 0, w_{21}(0) = 0, g_{12}(0) = 0, g_{21}(0) = 0</math></p> <p>2. <b>for</b> <math>n = 1:k</math> <b>do</b>  <b>Estimation des sorties</b></p> <p>3. <math>e_1(n) = p_1(n) - \mathbf{w}_{21}^T(n-1)\mathbf{p}_2(n)</math></p> <p>4. <math>e_2(n) = p_2(n) - \mathbf{w}_{12}^T(n-1)\mathbf{p}_1(n)</math></p> <p><b>Mise à jour des coefficients des filtres</b></p> $\mathbf{w}_{12}(n) = \mathbf{w}_{12}(n-1) + \mu_{12}(n)e_2(n)\left(\frac{\mathbf{p}_1(n)}{\xi + \ \mathbf{p}_1(n)\ ^2}\right)$ <p>5. <math>\mathbf{w}_{21}(n) = \mathbf{w}_{21}(n-1) + \mu_{21}(n)e_1(n)\left(\frac{\mathbf{p}_2(n)}{\xi + \ \mathbf{p}_2(n)\ ^2}\right)</math></p> <p><b>Mise à jour des pas d'adaptation</b></p> <p>6. <math>\mu_{12}(n) = \mu_{12,max} \frac{\ \mathbf{g}_{12}(n-1)\ ^2}{\ \mathbf{g}_{12}(n-1)\ ^2 + \delta}</math></p> <p>7. <math>\mu_{21}(n) = \mu_{21,max} \frac{\ \mathbf{g}_{21}(n-1)\ ^2}{\ \mathbf{g}_{21}(n-1)\ ^2 + \delta}</math></p> <p>8. <math>\mathbf{g}_{12}(n) = \alpha_1 \mathbf{g}_{12}(n-1) + (1 - \alpha_1) \frac{e_2(n)\mathbf{e}_1(n-m)}{\ \mathbf{p}_1(n)\ ^2 + \gamma}</math></p> <p>9. <math>\mathbf{g}_{21}(n) = \alpha_2 \mathbf{g}_{21}(n-1) + (1 - \alpha_2) \frac{e_1(n)\mathbf{e}_2(n-m)}{\ \mathbf{p}_2(n)\ ^2 + \gamma}</math></p> <p>10. <b>FIN</b></p>	<p><math>L</math> : Longueur du filtre adaptatif.  <math>k</math> : nombre d'itérations</p> <p><math>\mathbf{w}_{12}</math> : Filtre adaptatif (FIR) :  <math>\mathbf{w}_{12}(n) = [w_{12,0}(n), w_{12,1}(n), \dots, w_{12,L-1}(n)]^T</math></p> <p><math>\mathbf{w}_{21}</math> : Filtre adaptatif (FIR) :  <math>\mathbf{w}_{21}(n) = [w_{21,0}(n), w_{21,1}(n), \dots, w_{21,L-1}(n)]^T</math></p> <p><math>\mathbf{p}_1(n) = [p_1(n), p_1(n-1), \dots, p_1(n-L+1)]^T</math></p> <p><math>\mathbf{p}_2(n) = [p_2(n), p_2(n-1), \dots, p_2(n-L+1)]^T</math></p> <p><math>\mu_{12}(n)</math> : Premier pas d'adaptation :  <math>0 &lt; \mu_{12}(n) &lt; \mu_{12,max} &lt; 2</math></p> <p><math>\mu_{21}(n)</math> : Deuxième pas d'adaptation :  <math>0 &lt; \mu_{21}(n) &lt; \mu_{21,max} &lt; 2</math></p> <p><math>\alpha_1, \alpha_2</math> : Nombre positif inclut entre 0 et 1.  <math>\xi, \gamma</math> : Nombre positif constant.  <math>\delta</math> : Nombre positif constant pour contrôler les variations de <math>\mu_{12}(n)</math> et <math>\mu_{21}(n)</math>.</p> <p><math>m</math> : Indice de retard, <math>m = 0, 1, 2, \dots, L-1</math>.</p> <p><math>\mathbf{g}_{12}(n) = [g_{12,0}(n), g_{12,1}(n), \dots, g_{12,L-1}(n)]^T</math></p> <p><math>\mathbf{g}_{21}(n) = [g_{21,0}(n), g_{21,1}(n), \dots, g_{21,L-1}(n)]^T</math></p>

## 2.8 Conclusion

La séparation aveugle de sources est une technique qui consiste à découpler des signaux inconnus qui ont été mélangés d'une manière inconnue sans se référer à aucune information ni sur le système de mélange ni sur les signaux sources sauf les propriétés structurelles des sources et/ou du système de mélange. Récemment, cette technique a reçu beaucoup d'attention dans le domaine de traitement de signal et particulièrement dans les applications d'annulation du bruit acoustique et le rehaussement de la parole.

Dans ce chapitre, nous avons présenté les approches, mono-capteur et multi-capteurs, on a vu aussi les différents types de mélange convolutif et instantané, les deux structures principalement utilisées dans la séparation de sources appliquent pour de débruitage parole, la structure directe et récursive, nous avons présenté l'avantage et l'inconvénient de chaque structure.

Nous avons passé en revue quelques algorithmes utilisés dans la structure directe à savoir l'algorithme de gradient stochastique double (2C-FLMS), l'algorithme de gradient stochastique normalisé double (2C-FNLMS) et l'algorithme direct à pas d'adaptations variables (2C-VSSF)

Dans ce travail, nous allons utiliser la structure directe afin d'extraire les signaux originaux en utilisant des algorithmes basés sur les Métaheuristiques.

**Chapitre 3**  
**IMPLÉMENTATION DES MÉTAHEURISTIQUES DANS LE**  
**REHAUSSEMENT DE LA PAROLE**

## Chapitre 3

# IMPLÉMENTATION DES MÉTAHEURISTIQUES DANS LE REHAUSSEMENT DE LA PAROLE

### 3.1 Introduction

L'annulation du bruit acoustique sert à améliorer la qualité d'un signal de parole corrompu par différents types de bruit ou d'interférences. Différentes approches adaptatives ont été proposées pour résoudre le problème d'annulation de bruit acoustique (ANC) en utilisant des techniques basées sur les gradients stochastiques. Parmi les algorithmes du gradient qui sont largement utilisés, on cite les algorithmes des moindres carrés moyens (LMS) [29] et les moindres carrés moyens normalisés (NLMS) [52]. Cependant, ces algorithmes sont bien connus pour être des techniques d'optimisation locale et l'optimum global est rarement atteint. Pour résoudre le problème du minimum local dans l'ANC, plusieurs modifications et améliorations ont été proposées, telles que les algorithmes NLMS à pas variable (VSS-NLMS) [94], les algorithmes NLMS rapides en sous-bande [96], les filtres transversaux rapides [97] et les algorithmes de décorrélation normalisés.

D'autres algorithmes d'optimisation tels que les algorithmes à base des métaheuristiques sont de plus en plus utilisés afin de faire à face aux problèmes liés au minimum local, ils sont faciles à mettre en œuvre et réduisent considérablement le risque d'être coincés sur des minimums locaux [9], diverses approches Métaheuristiques ont été proposées pour résoudre le problème du ANC en utilisant un filtre adaptatif à réponse impulsionnelle infinie (IIR). Dans [34] l'auteur, utilise la structure standard d'annulation du bruit et un filtre adaptatif du type IIR combiné avec plusieurs algorithmes tels que l'essaim particulaire (PSO) et ses variantes [35],[36],[37],[38], l'algorithme de recherche gravitationnelle [39],

l'hybridation entre ces deux derniers (PSOGSA) [39] et l'algorithme de chauve-souris (BA) [40].

Dans la suite de ce chapitre, nous allons présenter la technique du filtrage adaptatif à l'aide des métaheuristiques, ensuite nous allons présenter la méthode d'implémentation du PSO dans la structure Forward bi-capteurs. Nous allons présenter aussi le principe et la formulation de l'algorithme prédateur proie dans un essaim particulière (PPPSO : Predator Prey Particule Swarm Optimization), et ainsi comment l'implémenter dans la structure Forward bi-capteurs. Puis nous terminerons ce chapitre par une conclusion.

### 3.2 Les métaheuristiques pour le filtrage adaptatif

Généralement, pour résoudre un problème d'optimisation, on utilise la totalité des données d'entrée. Par contre dans les problèmes typiques de filtrage adaptatif en ligne (Figure 3.1), l'ensemble des données d'entrée n'est pas toujours disponible ou il est d'une taille énorme pour être traité d'une manière efficace telle que le problème de rehaussement de la parole ou l'annulation d'écho acoustique [98]. Par conséquent, les données d'entrée doivent être traitées et évaluées par blocs. Dans ce cas, à chaque fois, la fonction coût ne fournira qu'une estimation de l'erreur réelle. La qualité de cette estimation dépend de la nature statistique des données d'entrée qu'on peut l'améliorer en calculant la moyenne des erreurs d'estimation sur une fenêtre de données d'entrée précédentes. Dans le filtrage adaptatif, le carré de l'erreur moyenne (MSE) entre la sortie du système inconnu et la sortie du filtre adaptatif représente la fonction coût typique et sera donc utilisé pour l'évaluation des performances de chaque agent de l'algorithme en temps réel. Dans le cas du rehaussement de la parole, L'erreur quadratique moyenne minimale (MSE) dans chaque trame est considérée comme la fonction de coût qui doit être optimisé par l'algorithme [34].

### 3.3 Choix de la fonction objective

La fonction objective en optimisation est définie comme la fonction qu'on souhaite maximiser ou minimiser pour choisir la meilleure solution à un problème donné parmi un ensemble de solutions possibles.

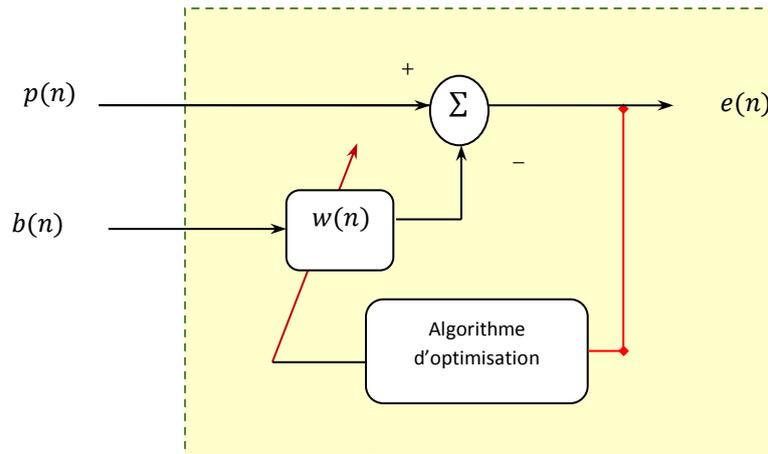


Figure 3.1 : Schéma général d'un système de filtrage adaptatif

L'évaluation de la fonction objective décrit la qualité de chaque solution. Cela nous permet de classer toutes les solutions de l'espace de recherche afin de choisir la meilleure solution, en d'autres termes la fonction objective guide la recherche vers les bonnes solutions dans l'espace de recherche.

Dans les techniques de programmation non linéaire [2], la fonction objective est décrite comme une équation à obtenir sous des contraintes données et avec des variables qui doivent être minimisées ou maximisées, si le problème de l'optimisation consiste à minimiser une fonction donnée, alors la fonction objective est appelée fonction coût.

La fonction objective peut être de nature continue/ non-continue, différentielle/ non-différentielle et linéaire/non linéaire, les variables d'un algorithme d'optimisation peuvent être réelles, complexes ou entières [1].

Dans le domaine de rehaussement de la parole, le signal bruité est segmenté en plusieurs trames, la méthode d'optimisation de la fonction coût est définie par l'erreur moyenne entre les observations et l'estimation du signal bruit dans chacune des trames, l'expression de la fonction objective  $J_i$  peut-être donnée par :

$$J_i = \frac{1}{N} \sum_{n=0}^{N-1} (p(n) - b(n) * w(n))^2 \quad (3.1)$$

Où  $J_i$  est la fonction objective de la trame,  $N$  est la longueur de la trame.

Après quelques itérations, l'algorithme d'optimisation nous fournit la meilleure solution quand la fonction objective atteint sa valeur minimale, dans ce cas, on peut dire que le filtre adaptatif  $w(n)$  à identifier le filtre réel, le signal débruité  $e(n)$  est estimé pour chaque trame  $i$  par :

$$e_i(n) = p_i(n) - b_i(n) * w_i(n) \quad (3.2)$$

Par la suite, on va introduire la méthode d'implémentation des algorithmes métaheuristiques dans la structure Forward bi-capteurs en particulier le PSO et son extension prédateurs proies dans un essaim particulière (PPSO : Predator Prey Particule Swarm Optimization).

### 3.4 Implémentation du PSO dans la structure Forward bi-capteurs

D'une manière générale pour résoudre un problème d'optimisation, en utilisant les métaheuristiques on a besoin d'évaluer la fonction objective à chaque itération par l'utilisation d'un ensemble de données d'entrée. Dans la structure Forward le signal observé qui représente les données d'entrée en temps réel du filtre adaptatif n'est pas entièrement disponible pour cette raison la meilleure façon consiste à évaluer la fonction objective pour chaque trame disponible à chaque itération, de plus pour mieux contrôler la mise à jour des coefficients du filtre adaptatif on va utiliser un détecteur manuel d'activité vocal ou manual voice activity detection (MVAD) [99], cette adaptation manuelle va nous permettre d'évaluer la fonction objective uniquement dans les périodes du silence (bruit seul) pour le cas du filtre  $w_{21}(n)$ , tandis que le filtre  $w_{12}(n)$  est mis à jour uniquement pendant les périodes de la présence du signal de parole [93]. Dans la sous-section suivante, on va passer en revue la méthode d'utilisation d'un MVAD pour le rehaussement de parole.

#### 3.4.1 Détecteur manuel d'activité vocale (MVAD)

La détection d'activité vocale est un processus qui consiste, à partir d'un signal de parole, à différencier les portions qui contiennent de la voix à celles qui n'en contiennent pas. Le détecteur d'activité vocal (VAD : Voice Activity Detector) fournit en sortie un signal binaire '0' qui signifie l'absence de parole et '1' qui signifie présence de voix.

Le VAD est utilisé dans de nombreux systèmes de communications. Parmi eux, citons la téléphonie mobile, les systèmes de communications sans fils et la reconnaissance de la parole [100], [101].

Le VAD découpe le signal parole d'entrée en trames. Une trame se définit comme une portion de signal de longueur fixe et de l'ordre de quelques millisecondes (habituellement de l'ordre de 10 à 50 ms). Le VAD va décider pour chaque trame si elle est active (présence de parole) ou inactive (absence de parole).

Le signal de parole est découpé en trame avec ou sans chevauchement (overlapping), par exemple un découpage à 0 % correspond à un découpage sans chevauchement où le signal est décomposé en trames de longueur égale.

Le chevauchement influe sur la précision du VAD et sur la complexité de calcul. En effet, un chevauchement à 50 % va demander plus de calcul qu'un chevauchement de 25 % (voir figure 3.2).

Plusieurs techniques sont utilisées pour extraire les paramètres avec lesquelles un VAD peut décider si la sortie vaut '0' ou '1' (respectivement inactivité ou activité vocale). Dans la littérature, on trouve des algorithmes dans lesquels, à partir d'une portion de signal audio, on extrait un ensemble de paramètres. Ces paramètres peuvent être calculés dans le domaine temporel (nombre de passages par zéro, niveau d'énergie, coefficients d'autocorrélation, etc.) ou à partir du domaine spectral (DFT, analyse de forme spectrale, DOA, ondelettes, etc.), dans la plupart des cas, on utilise des algorithmes basés sur des règles de seuils. Ces seuils peuvent être fixes ou adaptables selon le niveau de bruit.

Dans ce manuscrit on va utiliser un détecteur d'activité manuel (MVAD : Manual Activity Detector) combinée avec la structure directe bi-capteurs pour estimer le signal de parole original à partir de deux observations bruitées.

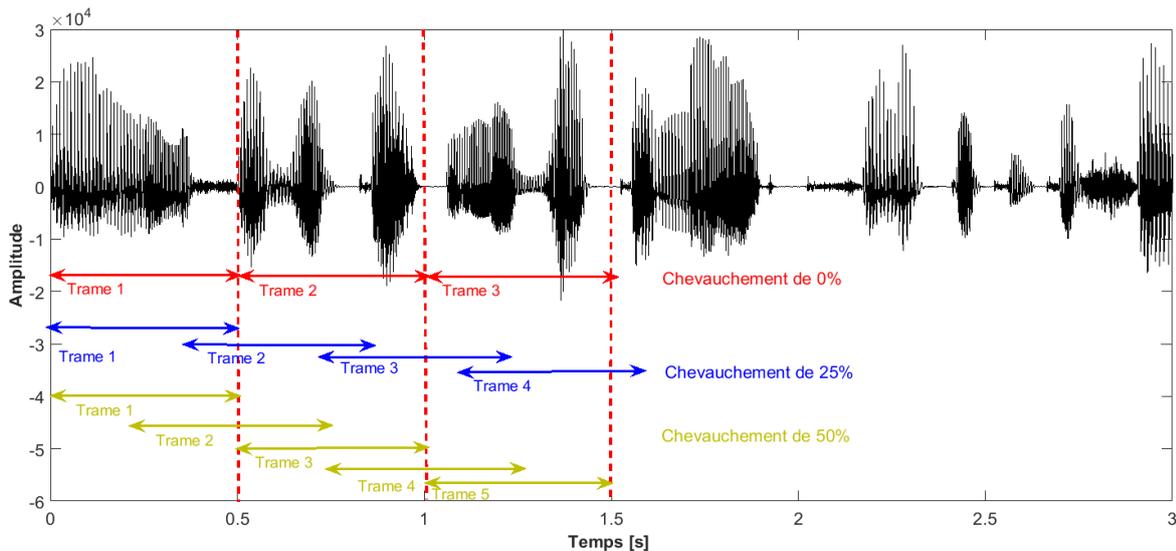


Figure 3.2 Découpage du signal audio à trois taux de chevauchement différents

### 3.4.2 Mise à jour des filtres adaptatifs

Avec l'utilisation d'un MVAD, les signaux observés sont découpés en trames, chaque trame est traitée selon la décision du MVAD. Les coefficients du filtre adaptatif sont mis à jour par l'algorithme d'essaim particulaire (PSO), où chaque agent de l'essaim représente les coefficients du filtre adaptatif  $w_{21}(n)$ .

Après quelques itérations et lorsque la valeur de la fonction coût  $J$  devient minime, alors  $w_{21}(n)$  nous fournira la meilleure solution.

L'implémentation du PSO dans la structure Forward peut être réalisée à l'aide des étapes suivantes :

Étape 1 : Générer aléatoirement les agents de l'essaim, chaque agent représente une solution probable du filtre  $\hat{h}(n)$ .

Étape 2 : Le signal d'entrée du système  $p_1(n)$  et  $p_2(n)$  sont découpés en trame en utilisant des fenêtres de 25 ms.

Étape 3 : Évaluer la fonction objective pour chaque agent, la fonction objective est définie comme l'erreur moyenne entre le signal désiré et la sortie.

Étape 4 : Trouver la solution globale caractérisée par l'agent qui a la meilleure performance pour l'itération  $k$  ;

Étape 5 : Mise à jour de la position et la vitesse de chaque agent :

$$\mathbf{v}_i^{k+1}(f) = w \mathbf{v}_i^k(f) + c_1 r_1 (\mathbf{b}_i^k(f) - \mathbf{x}_i^k(f)) + c_2 r_2 (\mathbf{y}_i^k(f) - \mathbf{x}_i^k(f)) \quad (3.3)$$

$$\mathbf{x}_i^{k+1}(f) = \mathbf{x}_i^k(f) + \mathbf{v}_i^{k+1}(f) \quad (3.4)$$

Où  $f$  est l'index de la trame.

Étape 6 : Répéter les étapes de 2 à 6 jusqu'à la rencontre du critère d'arrêt.

Toutes les formules mathématiques nécessaires pour implémenter le PSO dans la structure Forward bi-capteurs sont représentées dans le tableau 3.1.

### 3.5 Implémentation du PPPSO dans la structure Forward bi-capteurs

#### 3.5.1 Formulation de l'algorithme d'optimisation prédateur proie dans un essaim particulière

Bien que le PSO reste toujours parmi les meilleurs algorithmes d'optimisation, il présente des difficultés lorsqu'il est utilisé dans des problèmes de grande taille où le nombre de variables à optimiser est considérablement grand tel que la taille des filtres adaptatifs dans l'application d'annulation du bruit acoustique, dans ce cas le PSO souffre d'une convergence prématurée aux optimums locaux [102].

Pour éviter cette convergence prématurée, plusieurs chercheurs ont essayé une stratégie inspirée du système prédateur proie, pour développer un nouvel algorithme le prédateur proie dans un essaim particulière (PPPSO : Predator Prey Particle Swarm Optimization) [103].

Tableau 3.1 : Implémentation le PSO dans la structure Forward bi-capteurs

Détails de calcul	Variables
<ol style="list-style-type: none"> <li>1. Initialisation <math>w_{21}(0) = 0, L, Lf, N, MaxIt</math>.</li> <li>2. <i>for</i> <math>f = 1:Lf</math>: <b>Ldo</b></li> <li>3. <math>\mathbf{g} = \mathbf{w}_{21}(f - 1), c_1, c_2, w, P</math>.</li> <li>4. Initialisation de la population avec des positions et des vitesses aléatoires <math>\mathbf{x}_i</math> dans l'espace de recherche (<math>\mathbf{x}_i</math> représente une solution probable) ;</li> <li>5. <i>for</i> <math>i = 1:N</math> <b>do</b></li> <li>6. <i>if</i> <math>f(\mathbf{x}_i) \leq f(\mathbf{b}_i)</math></li> <li>7. <i>set</i> <math>\mathbf{b}_i = \mathbf{x}_i</math></li> <li>8. <i>endif</i></li> <li>9. <i>if</i> <math>f(\mathbf{x}_i) \leq f(\mathbf{g})</math></li> <li>10. <i>set</i> <math>\mathbf{g} = \mathbf{x}_i</math></li> <li>11. <i>endif</i></li> <li>12. <b>while</b> <math>iter \leq MaxIt</math> <b>do</b></li> <li>13. <i>for</i> <math>i = 1:N</math> <b>do</b></li> <li>14. Mise à jour de la vitesse de la <math>i^{\text{ème}}</math> particule.</li> <li>15. Mise à jour de la position de la <math>i^{\text{ème}}</math> particule <i>if</i> <math>f(\mathbf{x}_i) \leq f(\mathbf{b}_i)</math></li> <li>16. <i>set</i> <math>\mathbf{b}_i = \mathbf{x}_i</math></li> <li>17. <i>endif</i></li> <li>18. <i>if</i> <math>f(\mathbf{x}_i) \leq f(\mathbf{g})</math></li> <li>19. <i>set</i> <math>\mathbf{g} = \mathbf{x}_i</math></li> <li>20. <i>endif</i></li> <li>21. <i>endfor</i></li> <li>22. <math>w = w_{max} - \frac{w_{max} - w_{min}}{MaxIt} iter,</math> <math>c_1 = c_1 - \frac{2}{MaxIt}, c_2 = c_2 + \frac{2}{MaxIt}</math></li> <li>23. <math>iter = iter + 1</math></li> <li>24. <i>end while</i></li> <li>25. <i>set</i> <math>\mathbf{w}_{21}(f) = \mathbf{g}</math></li> <li>26. <math>\mathbf{e}_1(f) = \mathbf{p}_1(f) - \mathbf{p}_2^T(f) \mathbf{w}_{21}(f - 1)</math></li> <li>27. <i>end for</i></li> </ol>	<p><math>\mathbf{w}_{21}</math> : Le filtre adaptatif.  <math>L</math> : La longueur de la séquence d'entrée  <math>Lf</math> : la longueur de la trame.  <math>N</math> : Le nombre de particules.  <math>MaxIt</math> : Le nombre maximal d'itérations.  <math>\mathbf{g}</math> : La meilleure position globale  <math>c_1</math> : Paramètre de cognitivité  <math>c_2</math> : Paramètre social  <math>w</math> : Poids d'inertie.  <math>w_{max}, w_{min}</math> : La valeur maximale et minimale de l'inertie des  <math>\mathbf{x}_i</math> : Position de la <math>i^{\text{ème}}</math> particule  <math>\mathbf{v}_i</math> : Vitesse de la <math>i^{\text{ème}}</math> particule  <math>\mathbf{b}_i</math> : La meilleure position de la <math>i^{\text{ème}}</math> particule  <math>f(\cdot)</math> : La fonction objective où :  <math display="block">f(\mathbf{x}) = \frac{1}{Lf} \sum_{n=0}^{Lf} [\mathbf{p}_1(f) - \mathbf{p}_2^T(f) \mathbf{x}]^2</math> <math>\mathbf{p}_1(f), \mathbf{p}_2(f)</math> : Vecteur qui représente la <math>f^{\text{ème}}</math> trame des observations.  <math>\mathbf{e}_1(f)</math> : Vecteur qui représente la <math>f^{\text{ème}}</math> trame du signal débruité.</p>

### 3.5.2 Principe du prédateur-proie dans un essaim particulaire

Le prédateur-proie dans un essaim particulaire (PPPSO) a été présenté la première fois en 2003 par Silva et al [103], cette extension du PSO consiste à maintenir

l'équilibre entre l'exploration et à l'exploitation de l'essaim en introduisant une deuxième population des particules, qu'ils appellent "les prédateurs", ces derniers ont une dynamique différente de la première population qu'on appelle "les proies". Les prédateurs convergent aux particules qui présentent une meilleure performance dans l'ensemble de l'essaim, tandis que les proies sont repoussées ou chassées par les prédateurs, ce qui aide à maintenir une certaine diversité dans la population et assurer un bon équilibre entre l'exploration et à l'exploitation et par conséquent on réduit le risque de convergence prématurée aux optimums locaux, le mécanisme d'évasion des particules lors de la convergence de l'essaim autour d'un optimum local constitue le point fort du PPPSO.

Le concept prédateur-proie est inspiré de la chasse des troupeaux des animaux par un ou plusieurs prédateurs. Une fois, la chasse commence les proies se dispersent pour chercher d'autres endroits sûrs [103], le but du prédateur est de poursuivre les individus qui ont les meilleures performances dans l'espace de recherche. Les équations de mise à jour de la position des prédateurs et proies restent inchangées :

$$\mathbf{x}_{di}^{k+1} = \mathbf{x}_{di}^k + \mathbf{v}_{di}^{k+1} \quad (3.5)$$

$$\mathbf{x}_{yj}^{k+1} = \mathbf{x}_{yj}^k + \mathbf{v}_{yj}^{k+1} \quad (3.6)$$

Où

$\mathbf{v}_{yj}^k$ ,  $\mathbf{x}_{yj}^k$  sont la vitesse et la position du  $j^{\text{ième}}$  proies (prey) à l'itération  $k$ ,

$\mathbf{v}_{di}^k$ ,  $\mathbf{x}_{di}^k$  sont la vitesse et la position du  $i^{\text{ième}}$  prédateur (predator) à l'itération  $k$ ,

Tandis que les équations de mise à jour des vitesses deviennent [104] :

$$\mathbf{v}_{yj}^{k+1} = w_y \mathbf{v}_{yj}^k + c_1 r_1 (\mathbf{b}_{yi}^k - \mathbf{x}_{yj}^k) + c_2 r_2 (\mathbf{g}^k - \mathbf{x}_{yj}^k) - P \text{sign}(\mathbf{x}_{di}^k - \mathbf{x}_{yj}^k) \cdot \exp[-b|\mathbf{x}_{di}^k - \mathbf{x}_{yj}^k|] \quad (3.7)$$

$$\mathbf{v}_{di}^{k+1} = w_d \mathbf{v}_{di}^k + c_1 r_1 (\mathbf{b}_{di}^k - \mathbf{x}_{di}^k) + c_2 r_2 (\mathbf{g}^k - \mathbf{x}_{di}^k) \quad (3.8)$$

Où

$\mathbf{b}_{di}^k$ ,  $\mathbf{b}_{yj}^k$  sont les meilleures positions visitées par le  $i^{\text{ième}}$  et le  $j^{\text{ième}}$  prédateur et

proie respectivement à l'itération  $k$ ,  
 $g$  est la meilleure position de tout l'essaim ou meilleure position globale,

$w_d$  et  $w_y$  sont les paramètres d'inerties des prédateurs et proies respectivement, ils sont donnés par :

$$w_y = w_{max} - \frac{w_{max} - w_{min}}{maxit} iter \quad (3.9)$$

$$w_d = 0.2 \exp\left(-10 \frac{iter}{maxit}\right) + 0.4 \quad (3.10)$$

$w_{max}$  et  $w_{min}$  représentent la valeur maximale et minimale de l'inertie des proies (0.9 et 0.2).

$iter$  et  $maxit$  représentent l'itération actuelle et le maximum nombre d'itérations respectivement.

L'indice  $I$  représente l'indice de la proie la plus proche et qu'on peut calculer avec :

$$I = \{k | \min_k (|x_{dk} - x_{yj}|)\} \quad (3.11)$$

$P$  est une variable binaire représente la "peur", qui est la probabilité qu'une particule pour échapper à un prédateur ou non en changeant sa vitesse,  $a$  et  $b$  sont des paramètres qui détermine la manière dont les proies s'échappent des prédateurs :

$$a = x_{span}, b = \frac{100}{x_{span}} \quad (3.12)$$

Où  $x_{span}$  représente étendue d'une variable

### 3.5.3 Méthode d'implémentation du PPPSO dans la structure Forward

L'algorithme PPPSO a été utilisé avec succès dans plusieurs applications. Il se caractérise par sa vitesse de convergence et la bonne qualité de la solution obtenue [105],[106],[107],[108]. L'implémentation du PPPSO dans la structure Forward bi-capteurs est assurée par les mêmes étapes utilisées pour le PSO à l'exception des

équations de mise à jour des positions et de vitesses des particules, en effet, toutes les équations de l'algorithme PPPSO sont résumées dans le tableau 3.2.

Tableau 3.2 : Implémentation du PPPSO dans la structure Forward bi-capteurs

Détails de calcul	Variables
<ol style="list-style-type: none"> <li>1. Initialisation <math>w_{21}(0) = 0, L, Lf, N_d, N_y, MaxIt</math>.</li> <li>2. <i>for</i> <math>f = 1 : Lf : L</math> <b>do</b></li> <li>3. <math>\mathbf{g} = \mathbf{w}_{21}(f - 1), c_1, c_2, w_d, w_p, P</math>.</li> <li>4. Initialisation de la population des prédateurs avec des positions et des vitesses aléatoires <math>\mathbf{x}_{di}, \mathbf{v}_{di}</math> dans l'espace de recherche (<math>\mathbf{x}_{di}</math> représente une solution probable) ;</li> <li>5. <i>for</i> <math>i = 1 : N_d</math> <b>do</b></li> <li>6. <i>if</i> <math>f(\mathbf{x}_{di}) \leq f(\mathbf{b}_{di})</math></li> <li>7. <i>set</i> <math>\mathbf{b}_{di} = \mathbf{x}_{di}</math></li> <li>8. <i>endif</i></li> <li>9. <i>if</i> <math>f(\mathbf{x}_{di}) \leq f(\mathbf{g})</math></li> <li>10. <i>set</i> <math>\mathbf{g} = \mathbf{x}_{di}</math></li> <li>11. <i>endif</i></li> <li>12. Initialisation de la population des proies avec des positions et des vitesses aléatoires <math>\mathbf{x}_{yj}, \mathbf{v}_{yj}</math> dans l'espace de recherche (<math>\mathbf{x}_{yj}</math> représente une solution probable) ;</li> <li>13. <i>for</i> <math>i = 1 : N_y</math> <b>do</b></li> <li>14. <i>if</i> <math>f(\mathbf{x}_{yi}) \leq f(\mathbf{b}_{yi})</math></li> <li>15. <i>set</i> <math>\mathbf{b}_{yi} = \mathbf{x}_{yi}</math></li> <li>16. <i>endif</i></li> <li>17. <i>if</i> <math>f(\mathbf{x}_{yi}) \leq f(\mathbf{g})</math></li> <li>18. <i>set</i> <math>\mathbf{b}_{yi} = \mathbf{g}</math></li> <li>19. <i>endif</i></li> <li>20. <b>while</b> <math>iter \leq MaxIt</math> <b>do</b></li> <li>21. <i>for</i> <math>i = 1 : N_d</math> <b>do</b></li> <li>22. Mise à jour de la vitesse du <math>i^{\text{ème}}</math> prédateur.</li> <li>23. Mise à jour de la position du <math>i^{\text{ème}}</math> prédateur.</li> <li>24. <i>if</i> <math>f(\mathbf{x}_{di}) \leq f(\mathbf{b}_{di})</math></li> <li>25. <i>set</i> <math>\mathbf{b}_{di} = \mathbf{x}_{di}</math></li> <li>26. <i>endif</i></li> <li>27. <i>if</i> <math>f(\mathbf{x}_{di}) \leq f(\mathbf{g})</math></li> </ol>	<p><math>\mathbf{w}_{21}</math> : Le filtre adaptatif.  <math>L</math> : La longueur de la séquence d'entrée  <math>Lf</math> : la longueur de la trame.  <math>N_d</math> : Le nombre des prédateurs.  <math>N_y</math> : Le nombre des proies.  <math>MaxIt</math> : Le nombre maximal d'itérations.  <math>\mathbf{g}</math> : La meilleure position globale  <math>c_1</math> : Paramètre de cognitivité  <math>c_2</math> : Paramètre social  <math>w_d</math> : Poids d'inertie du prédateur.  <math>w_y</math> : Poids d'inertie de la proie  <math>w_{max}, w_{min}</math> : La valeur maximale et minimale de l'inertie des  <math>P</math> : Variable binaire  <math>\mathbf{x}_{di}</math> : Position du <math>i^{\text{ème}}</math> prédateur  <math>\mathbf{v}_{di}</math> : Vitesse du <math>i^{\text{ème}}</math> prédateur  <math>\mathbf{b}_{di}</math> : La meilleure position du <math>i^{\text{ème}}</math> prédateur  <math>\mathbf{x}_{yj}</math> : Position de la <math>j^{\text{ème}}</math> proie  <math>\mathbf{v}_{yj}</math> : Vitesse de la <math>j^{\text{ème}}</math> proie  <math>\mathbf{b}_{yj}</math> : La meilleure position de la <math>j^{\text{th}}</math> proie  <math>f(\cdot)</math> : La fonction objective où :</p>

```

28. set  $\mathbf{g} = \mathbf{x}_{di}$ 
29. endif
30. for  $j = 1:N_y$  do
31. Mise à jour de la vitesse de la  $j^{\text{ème}}$  proie.
32. Mise à jour de la position de la  $j^{\text{ème}}$  proie.
33. if  $f(\mathbf{x}_{yj}) \leq f(\mathbf{b}_{yj})$ 
34. set  $\mathbf{b}_{yj} = \mathbf{x}_{yj}$ 
35. endif
36. if  $f(\mathbf{x}_{yj}) \leq f(\mathbf{g})$ 
37. set  $\mathbf{g} = \mathbf{x}_{yj}$ 
38. endif
39. endfor
40. endfor
41.  $w_y = w_{max} - \frac{w_{max} - w_{min}}{MaxIt} iter, w_d =$ 
     $0.2 \exp(-10 \frac{iter}{MaxIt}) + 0.4$ 
42.  $c_1 = c_1 - \frac{2}{MaxIt}, c_2 = c_2 + \frac{2}{MaxIt}$ 
43. if  $|f(\mathbf{g}(iter)) - f(\mathbf{g}(iter - 1))| \leq k_{th}$ 
44. Réinitialisation de la position des proies
45. endif
46.  $iter = iter + 1$ 
47. end while
48. set  $\mathbf{w}_{21}(f) = \mathbf{g}$ 
49.  $\mathbf{e}_1(f) = \mathbf{p}_1(f) - \mathbf{p}_2^T(f) \mathbf{w}_{21}(f - 1)$ 
50. end for

```

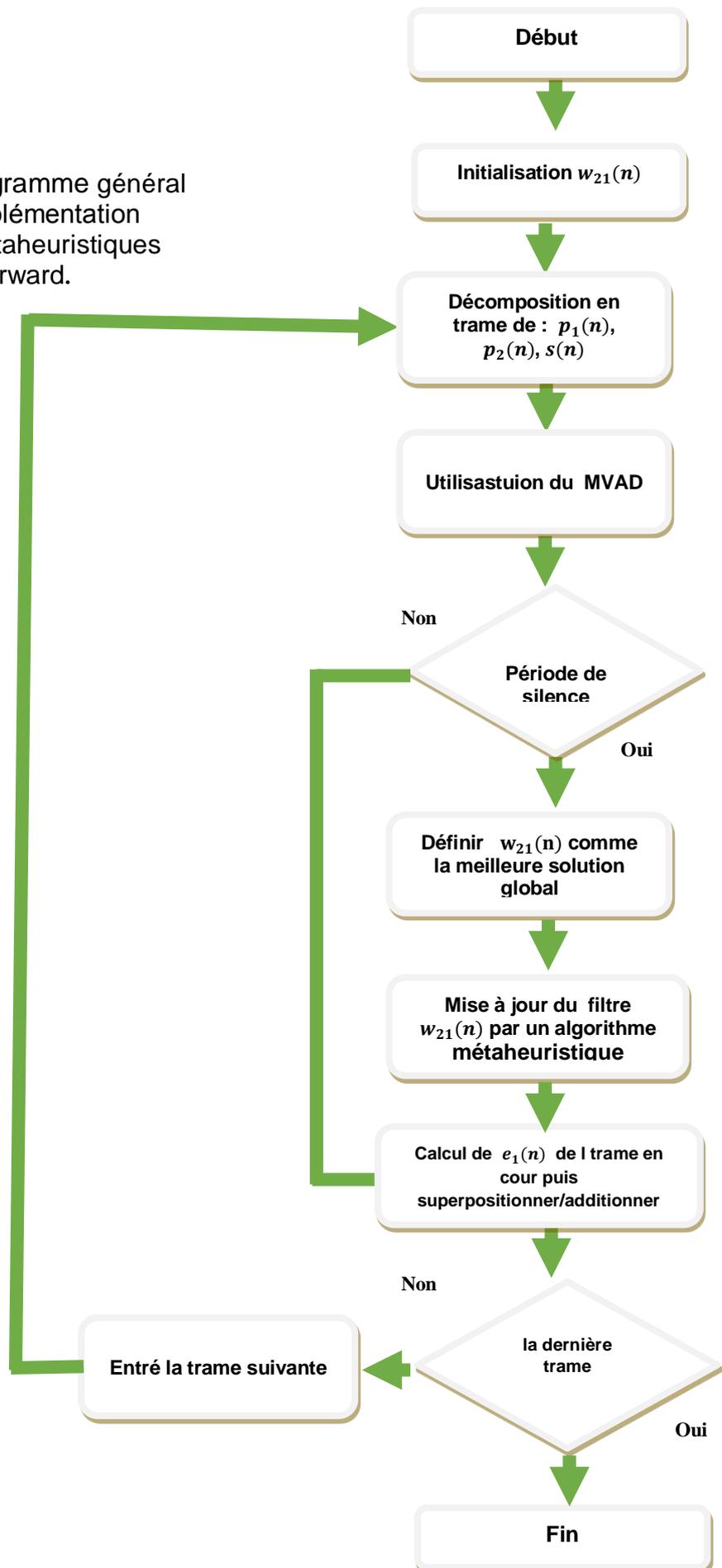
$$f(\mathbf{x}) = \frac{1}{L_f} \sum_{n=0}^{L_f} [\mathbf{p}_1(f) - \mathbf{p}_2^T(f) \mathbf{x}]^2$$

$\mathbf{p}_1(f), \mathbf{p}_2(f)$ : Vecteur qui représente la  $f^{\text{ème}}$  trame des observations.

$\mathbf{e}_1(f)$ : Vecteur qui représente la  $f^{\text{ème}}$  trame du signal débruité.

En général l'implémentation d'un algorithme métaheuristique peut être réalisée de la même manière que le PSO en suivant les mêmes étapes, la figure 3.3 montre l'organigramme général de la méthode d'implémentation des algorithmes métaheuristiques dans la structure Forward bi-capteurs.

Figure 3.3 : Organigramme général de la méthode d'implémentation des algorithmes métaheuristiques dans la structure Forward.



### 3.6 CONCLUSION

L'annulation du bruit acoustique peut être réalisée à l'aide des filtres adaptatifs. Différentes techniques adaptatives basées essentiellement sur les algorithmes du gradient. Cependant, ces algorithmes sont bien connus pour leur convergence prématurée vers des optimums locaux ce qui influe considérablement sur la qualité de la solution obtenue. Nous avons vu que d'autres algorithmes à base des métaheuristiques constituent une bonne alternative pour résoudre le problème des minimums locaux, nous avons présenté la méthode et les étapes nécessaires pour l'implémentation du PSO dans la structure Forward bi-capteurs.

Nous avons présenté une extension très intéressante du PSO qui consiste à combiner le concept du prédateur proie et le PSO standard pour obtenir le PPPSO. Nous avons montré aussi la technique implémentation du PPPSO dans la structure Forward bi-capteurs.

**CHAPITRE 4**  
**APPLICATION DU PPPSO ET MPPPSO DANS LA**  
**STRUCTURE FORWARD**

## CHAPITRE 4

# APPLICATION DU PPPSO ET DU MPPPSO DANS LA STRUCTURE FORWARD

### 4.1 Introduction

L'annulation du bruit et la séparation aveugle de sources dans les systèmes de télécommunication numérique peuvent être réalisées à l'aide des filtres adaptatifs ou plusieurs algorithmes tels que LMS et le NLMS jouent un rôle important dans ces applications et restent les plus utilisés dans ce domaine à cause de leur simplicité d'implémentation et leur robustesse.

Parmi les paramètres qui influent sur le taux de convergence des algorithmes LMS et NLMS dans l'application d'annulation du bruit, on cite la longueur du filtre à adapter et la corrélation du signal d'entrée du filtre. Pour améliorer le taux de convergence par rapport aux algorithmes classiques, l'adaptation par des algorithmes métaheuristiques s'avère être une alternative intéressante. Dans ce chapitre, nous nous intéresserons à la structure Forward bi-capteurs de séparation de sources pour l'annulation du bruit acoustique et le rehaussement de la parole.

Considérant le problème du mélange linéaire convolutif bi-capteurs entre le signal de parole et le bruit nous proposons d'utiliser une nouvelle version de l'algorithme PPPSO : Modified Predator Prey Particle Swarm Optimization.

Dans la suite de ce chapitre, nous allons présenter la formulation de l'algorithme MPPPSO, puis nous allons présenter la méthode d'implémentation du MPPPSO dans la structure Forward bi-capteurs. Nous allons présenter aussi les résultats comparatifs entre l'algorithme proposé (MPPPSO), sa version originale (PPPSO) et deux des méthodes classiques (2C-NLMS et 2C-VSSF) [109], puis nous terminerons ce chapitre par une conclusion.

## 4.2 Description de l'algorithme proposé Dual MPPPSO

Dans l'algorithme PPSO classique, chaque proie se déplace dans l'espace de recherche selon un mouvement d'échappement lié au prédateur le plus proche en plus de sa meilleure performance et sa meilleure position globale, après certaines itérations, le processus d'échappement peut cesser de progresser lorsque les prédateurs sont piégés dans une solution locale ou minimum local. Pour résoudre ce problème, une nouvelle stratégie appelée Modified Predator Prey Particle Swarm Optimization (MPPPSO) est proposée dans cette thèse. Cette nouvelle version consiste à maintenir les proies en mouvement, même si les prédateurs sont empilés dans des minimums locaux. Dans la suite, nous donnons les détails du MPPPSO.

## 4.3 Formulation de l'algorithme proposé MPPPSO

Dans cette section, nous donnons la formulation mathématique de l'algorithme proposé MPPPSO.

D'abord, on utilise l'équation (4.1) pour mettre à jour la vitesse des prédateurs :

$$v_{di}^{k+1}(f) = w_d v_{di}^k(f) + c_1 r_1 (b_{di}^k(f) - x_{di}^k(f)) + c_2 r_2 (g^k(f) - x_{di}^k(f)) \quad (4.1)$$

Et l'équation (4.2) pour mettre à jour la vitesse des proies :

$$v_{yj}^{k+1}(f) = w_y v_{yj}^k(f) + c_1 r_1 (b_{yj}^k(f) - x_{yj}^k(f)) + c_2 r_2 (g^k(f) - x_{yj}^k(f)) - P \operatorname{sign}(x_{di}^k(f) - x_{yj}^k(f)) \cdot \exp[-b|x_{di}^k(f) - x_{yj}^k(f)|] \quad (4.2)$$

Où :

$j$  représente l'index de la  $j^{\text{ème}}$  proie,  $w_y$  et  $w_d$  sont les inerties données par l'équation (3.9) et (3.10) respectivement. En analysant les relations (4.1) et (4.2) ; on peut remarquer que dans le MPPPSO, les proies échappent à tous les prédateurs dans la même itération de l'algorithme, tandis que dans le PPSO, les proies échappent uniquement du prédateur le plus proche, par conséquent la fonction objective est évaluée à chaque mouvement dans une seule itération. Ces

modifications améliorent nettement la qualité de la solution puisqu'on évite la convergence prématurée de l'algorithme.

De plus, on peut augmenter la chance d'atteindre l'optimum global en effectuant une réinitialisation de la position de toutes les proies dans le cas où la valeur du meilleur coût reste constante pendant un certain nombre d'itérations :

$$|f(\mathbf{g}(i)) - f(\mathbf{g}(i - 1))| \leq k_{th} \quad (4.3)$$

Où :

$|\cdot|$  désigne la valeur absolue et  $k_{th}$  un nombre suffisamment petit (proche du zéro) ;  $i$  représente l'index du  $i^{\text{ème}}$  meilleur coût global.

Le pseudo-code de l'algorithme MPPPSO implémenté dans la structure FBSS est donné dans le tableau (4.1), pour mettre à jour le filtre du premier canal  $w_{21}(n)$ , la mise à jour du filtre  $w_{12}(n)$  est obtenue de la même manière.

Tableau 4.1 : Implémentation du MPPPSO dans la structure Forward bi-capteurs

Détails de calcul	Variables
<ol style="list-style-type: none"> <li>1. Initialiser <math>w_{21}(0) = 0, L, Lf, N_d, N_y, MaxIt</math>.</li> <li>2. <i>for</i> <math>f = 1:Lf:L</math> <b>do</b></li> <li>3. <math>\mathbf{g} = \mathbf{w}_{21}(f - 1), c_1, c_2, w_d, w_p, P</math>.</li> <li>4. Initialiser <math>\mathbf{x}_{di}</math> et <math>\mathbf{v}_{di}</math> Avec des positions et vitesses aléatoires ;</li> <li>5. <i>for</i> <math>i = 1:N_d</math> <b>do</b></li> <li>6. <i>if</i> <math>f(\mathbf{x}_{di}) \leq f(\mathbf{b}_{di})</math></li> <li>7. <i>set</i> <math>\mathbf{b}_{di} = \mathbf{x}_{di}</math></li> <li>8. <i>endif</i></li> <li>9. <i>if</i> <math>f(\mathbf{x}_{di}) \leq f(\mathbf{g})</math></li> <li>10. <i>set</i> <math>\mathbf{g} = \mathbf{x}_{di}</math></li> <li>11. <i>endif</i></li> <li>12. Initialiser <math>\mathbf{x}_{yi}</math> et <math>\mathbf{v}_{yi}</math> Avec des positions et vitesses aléatoires ;</li> <li>13. <i>for</i> <math>i = 1:N_y</math> <b>do</b></li> <li>14. <i>if</i> <math>f(\mathbf{x}_{yi}) \leq f(\mathbf{b}_{yi})</math></li> <li>15. <i>set</i> <math>\mathbf{b}_{yi} = \mathbf{x}_{yi}</math></li> <li>16. <i>endif</i></li> <li>17. <i>if</i> <math>f(\mathbf{x}_{yi}) \leq f(\mathbf{g})</math></li> </ol>	<p><math>\mathbf{w}_{21}</math> : Filtre FIR adaptatif.  <math>L</math> : Longueur du filtre adaptatif.  <math>Lf</math> : Longueur de la trame  <math>N_d</math> : Nombre de prédateurs.  <math>N_y</math> : Nombre de proies.  <math>maxit</math> : Nombre d'itérations  <math>\mathbf{g}</math> : Meilleure position globale  <math>c_1</math> : Paramètre de cognitivité  <math>c_2</math> : Paramètre social  <math>w_d</math> : Inertie du prédateur  <math>w_y</math> : Inertie de la proie  <math>w_{max}, w_{min}</math> : Valeur maximale et minimale des inerties des proies choisies entre 0.9 et 0.2.  <math>P</math> : Variable binaire  <math>\mathbf{x}_{di}</math> : Position du <math>i^{\text{ème}}</math> prédateur  <math>\mathbf{v}_{di}</math> : Vitesse du <math>i^{\text{ème}}</math> prédateur  <math>\mathbf{b}_{di}</math> : Meilleure Position du <math>i^{\text{ème}}</math> prédateur</p>

```

18. set  $\mathbf{g} = \mathbf{x}_{yi}$ 
19. endif
20. while iter <= MaxIt do
21. for i = 1:  $N_d$  do
22. Mise à jour de la vitesse de  $i^{\text{ième}}$  prédateur
23. Mise à jour de la position du  $i^{\text{ième}}$  prédateur
24. iff( $\mathbf{x}_{di}$ ) <=  $f(\mathbf{b}_{di})$ 
25. set  $\mathbf{b}_{di} = \mathbf{x}_{di}$ 
26. endif
27. iff( $\mathbf{x}_{di}$ ) <=  $f(\mathbf{g})$ 
28. set  $\mathbf{g} = \mathbf{x}_{di}$ 
29. endif
30. for j = 1:  $N_y$  do
31. Mise à jour de la vitesse de  $j^{\text{ième}}$  proie
32. Mise à jour de la position de la  $j^{\text{ième}}$  proie
33. iff( $\mathbf{x}_{yj}$ ) <=  $f(\mathbf{b}_{yj})$ 
34. set  $\mathbf{b}_{yj} = \mathbf{x}_{yj}$ 
35. endif
36. iff( $\mathbf{x}_{yj}$ ) <=  $f(\mathbf{g})$ 
37. set  $\mathbf{g} = \mathbf{x}_{yj}$ 
38. endif
39. endfor
40. endfor
41.  $w_y = w_{max} - \frac{w_{max} - w_{min}}{MaxIt} iter, w_d =$ 
     $0.2 \exp(-10 \frac{iter}{MaxIt}) + 0.4$ 
42.  $c_1 = c_1 - \frac{2}{MaxIt}, c_2 = c_2 + \frac{2}{MaxIt}$ 
43. if  $|f(\mathbf{g}(iter)) - f(\mathbf{g}(iter - 1))| \leq k_{th}$ 
44. Réinitialiser la position de toutes les proies
45. endif
46. iter = iter + 1
47. endwhile
48. set  $\mathbf{w}_{21}(f) = \mathbf{g}$ 
49.  $\mathbf{e}_1(f) = \mathbf{p}_1(f) - \mathbf{p}_2^T(f) \mathbf{w}_{21}(f - 1);$  end for f.

```

$\mathbf{x}_{yj}$  : Position de la  $j^{\text{ième}}$  proie  
 $\mathbf{v}_{yj}$  : Vitesse de la  $j^{\text{ième}}$  proie  
 $\mathbf{b}_{yi}$  : Meilleure position de la  $j^{\text{ième}}$  proie

$f(\cdot)$  : Fonction objective avec :

$$f(\mathbf{x}) = \frac{1}{Lf} \sum_{n=0}^{Lf} [\mathbf{p}_1(f) - \mathbf{p}_2^T(f) \mathbf{x}]^2$$

$\mathbf{p}_1(f), \mathbf{p}_2(f)$ : Vecteur représentant la  $f^{\text{ième}}$  trame des signaux de mélange

$\mathbf{e}_1(f)$ : Vecteur représentant la  $f^{\text{ième}}$  trame du signal estimé (canal 1)

#### 4.4 Résumé de l'algorithme bi capteurs 2C-FNLMS

Plusieurs algorithmes adaptatifs ont été utilisés dans la structure FBSS pour adapter les filtres  $\mathbf{w}_{12}$  et  $\mathbf{w}_{21}$ . L'algorithme de gradient stochastique normalisé double (2CNLMS : Two-Channel Forward NLMS) est l'un des algorithmes les plus utilisés

en raison de sa simplicité d'implémentation [24], dans le NLMS les deux pas d'adaptations  $\mu_{12}$  et  $\mu_{21}$  sont normalisés par l'énergie des signaux d'entrée  $p_1(n)$  et  $p_2(n)$ , respectivement. Pour l'implémentation de l'algorithme 2C-FNLMS, en utilisant les deux structures représentées dans les deux figures 2.11 et 2.12. Les équations mathématiques nécessaires pour mettre à jour les coefficients des deux filtres adaptatifs sont données dans le tableau 2.2.

#### 4.5 Résumé de l'algorithme Forward à pas d'adaptations variables (2C-VSSF)

Bien que l'algorithme NLMS ait été appliqué avec succès dans de nombreux problèmes de traitement de la parole tels que l'annulation du bruit acoustique et l'annulation de l'écho acoustique, il présente des difficultés à éviter les conflits entre la convergence rapide et les faibles désajustements. Pour obtenir de meilleures performances et améliorer l'algorithme NLMS, de nombreuses variantes basées sur la taille de pas variable ont été développées. Dans cette sous-section. Nous présentons dans le tableau 2.3 l'algorithme Forward à pas d'adaptations variables (2C-VSSF : Two-Channel Variable Step-Sizes Forward).

#### 4.6 Évaluation des résultats de simulation

Dans cette section, nous présentons plusieurs résultats de simulations pour l'annulation du bruit par l'algorithme Forward bi-capteurs. Également nous allons présenter les résultats comparatifs entre l'algorithme proposé Dual MPPSO, sa version originale PPPSO, l'algorithme 2C-NLMS et l'algorithme 2C-VSSF, pour valider la performance et voir la supériorité de l'algorithme proposé Dual MPPSO, en faisons une comparaison basée sur les critères suivants : l'évolution temporelle (description) des signaux de tests et rehaussés, le désajustement et le rapport signal à bruit segmenté.

##### 4.6.1 Critère de désajustement (System Mismatch)

Le critère de désajustement (SM : System Mismatch) est un critère robuste très utilisé dans le domaine adaptatif pour évaluer les performances des algorithmes adaptatifs. Le SM est calculé à partir de la distance Euclidienne entre les coefficients du filtre réel et les coefficients du filtre adaptatif. Le but est de comparer la vitesse de convergence des systèmes de débruitage par des algorithmes adaptatifs.

Le critère de désajustement est défini par l'expression suivante :

$$SM_{dB} = 20 \log \left( \frac{\|h_{21} - w_{21}\|}{\|h_{21}\|} \right) \quad (4.4)$$

Où  $\|h_{21} - w_{21}\|$  représente la distance Euclidienne entre le vecteur des coefficients du filtre réel  $h_{21}$ , et le vecteur des coefficients du filtre adaptatif estimé  $w_{21}$ .

#### 4.6.2 Rapport signal à bruit segmenté

Le rapport signal à bruit segmenté (SegSNR : Segmental Signal-to-Noise Ratio) est le critère le plus utilisé pour tester la qualité du signal de parole rehaussé (niveau de la suppression du bruit). Mathématiquement, le SegSNR est calculé en utilisant le rapport du signal à bruit segmenté correspondant à la première sortie, il est donné par la relation suivante [110] :

$$SegSNR_{dB} = 10 \log \left( \frac{\sum_{i=0}^{U-1} |s(i)|^2}{\sum_{i=0}^{U-1} |s(i) - e_1(i)|^2} \right) \quad (4.5)$$

Où  $s(n)$  et  $e_1(i)$  représentent le signal de parole original et le signal estimé (rehausser) le paramètre  $U$  représente le nombre d'échantillons nécessaires pour obtenir la valeur moyenne de la sortie. On note qu'on utilisera un détecteur d'activité vocal manuel (MVAD: Manual Voice Activity Detector) pour tous les algorithmes, le MVAD permet d'adapter le filtre du premier canal  $w_{12}(n)$  uniquement pendant les périodes de parole et le filtre du deuxième  $w_{21}(n)$  pendant les périodes de silence [111]. On rappelle que les signaux mixés  $p_1(n)$  et  $p_2(n)$  sont traité trame par trame, chaque trame (*frame*) contient 256 échantillons qui correspond à une durée de 25 ms de parole [112].

#### 4.6.3 Description des signaux d'entrée

Dans toutes les simulations, les deux sources d'entrée du mélange convolutif bi-capteurs sont les signaux originaux suivants :

- La première source c'est un signal de parole constitué d'une phrase phonétiquement équilibrée (prononcée par un locuteur masculin en longue

Française), sa durée presque de 3.5 secondes échantillonnées à 8 kHz figure 4.1.

- Pour la deuxième source, on a utilisé deux signaux de tests le premier correspond à un bruit blanc Gaussien, dans lequel la densité spectrale de puissance est la même sur toutes les fréquences figure 4.1, le deuxième c'est un signal du bruit USASI (United States of America Standard Institute) le bruit USASI est un signal corrélé utilisé pour tester la vitesse de convergence des algorithmes adaptatifs [94].

Les deux réponses impulsionnelles  $h_{12}(n)$  et  $h_{21}(n)$  sont générées par une séquence aléatoire avec une fonction exponentielle décroissante [85], [113], ces deux réponses ont une longueur de  $L = 100$  coefficients et échantillonné a 8-kHz (figure 4.2), ces deux réponses impulsionnelles sont utilisées pour générer les deux signaux bruités, la figure 4.3 montre les signaux  $p_1(n)$  et  $p_2(n)$  obtenus du processus de mélange avec rapports signaux à bruits dans les deux voies à -3 dB ( $RSB_1 = RSB_2 = -3dB$ ).

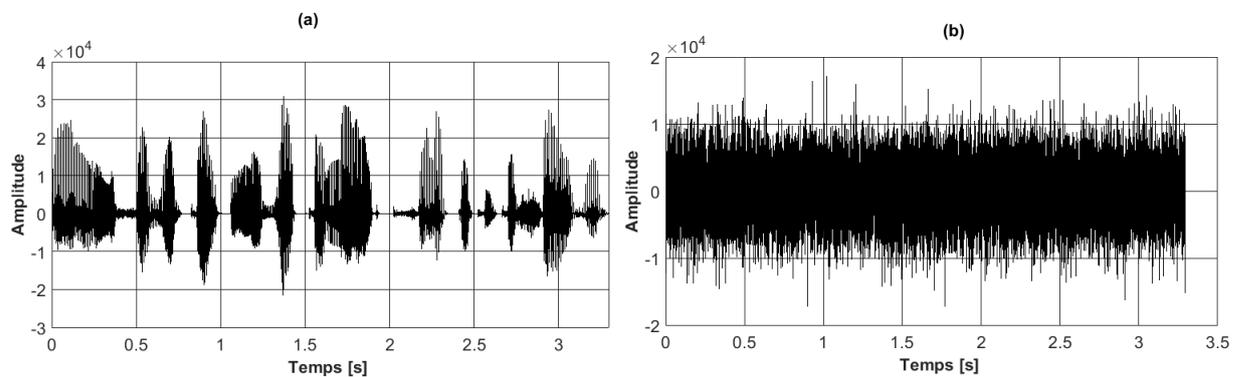


Figure 4.1 : Description temporelle des signaux originaux (a) : le signal de parole  $s(n)$ , (b) : le signal du bruit blanc  $b(n)$ .

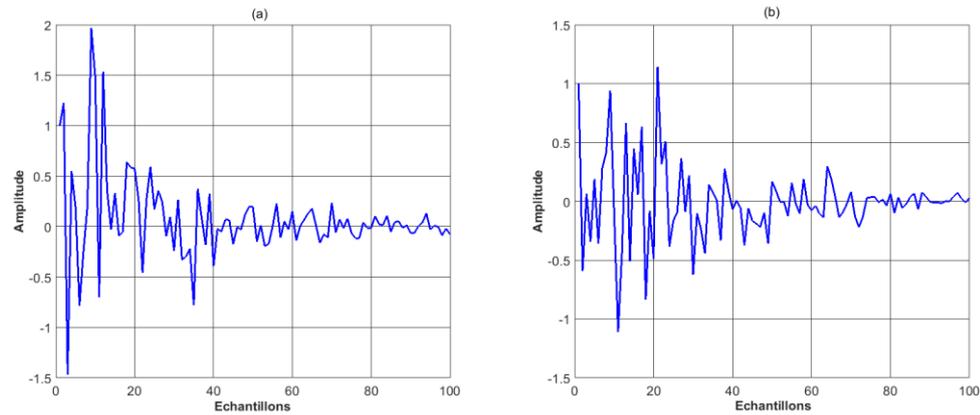


Figure 4.2 : Exemple des réponses impulsionnelles,  $L = 100$  (a) :  $h_{12}(n)$ , (b) :  $h_{21}(n)$ .

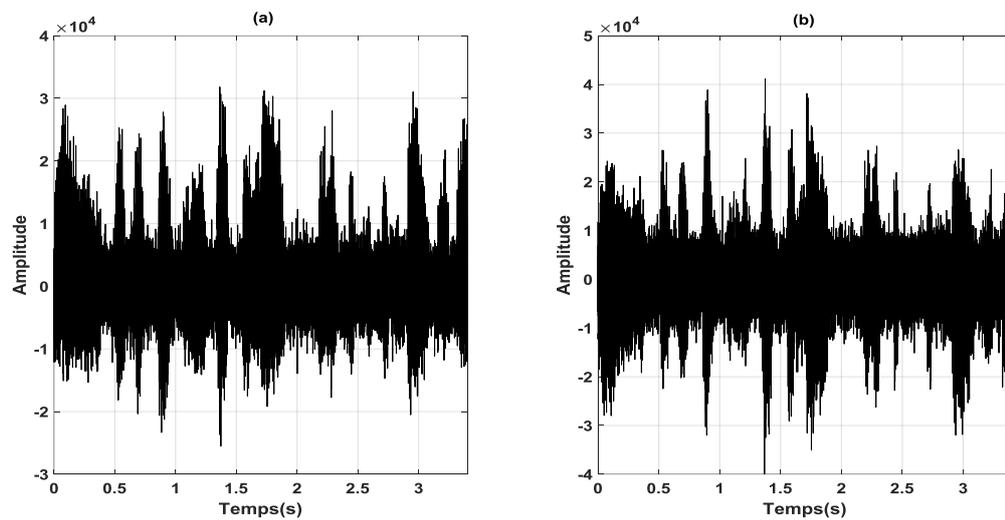


Figure 4.3 : Description temporelle des signaux du mélange (a) : le signal,  $p_1(n)$ , (b) : le signal  $p_2(n)$ .

#### 4.6.4 Test avec un bruit blanc

Dans la première simulation, on considère le cas où la deuxième source est un bruit blanc avec un rapport signal sur bruit dans les deux voies fixées à  $-3dB$ , ( $RSB_1 = RSB_2 = -3dB$ ).

Pour les deux algorithmes métaheuristiques à savoir le PPPSO et le Dual MPPSO le nombre de particule utilisé pour la résolution d'un problème d'optimisation en général dépend essentiellement de deux facteurs, la taille de l'espace de recherche et le rapport entre les capacités de calcul de machine et le temps maximum de

recherche. Il n'y a pas de règle fixe pour déterminer ces paramètres, En général le choix se fait aléatoirement [2].

Pour le 2C-FNLMS de nombreux essais ont permis de choisir les paramètres nécessaires pour avoir une bonne performance.

Les paramètres d'ajustement pour le Dual MPPPSO, sa version originale PPPSO, le 2C-FNLMS et le 2C-VSSF sont résumés dans le tableau suivant :

Tableau 4.2 : paramètres de simulation des algorithmes Dual MPPPSO, PPPSO, 2C-NLMS et 2C-VSSF.

Algorithme	Paramètres
PPPSO [104]	$N_y = 18; N_p = 8; c_1=c_2 = 2; w_{min} = 0.2; w_{max} = 0.9; P = 0.00055; maxit = 2000; \text{Espace de recherche } [-10, 10]^n .$
2C-FNLMS [52]	$\mu_{12} = \mu_{21} = 0.5, 1$
2C-VSSF [94]	$\mu_{12}(0) = 0.2, \mu_{12,max} = 2, \mu_{21}(0) = 0, \mu_{21,max} = 1, \alpha_1 = 0.98, \alpha_2 = 0.60, \delta = 10^{-4}, \gamma = \xi = 10^{-6} .$
Dual MPPPSO	$N_d = 18; N_y = 8; c_1=c_2 = 2; w_{min} = 0.2; w_{max} = 0.9; P = 0.00055; maxit = 2000; \text{Espace de recherche } [-10, 10]^n .$

#### 4.6.4.1 Description temporelle des signaux estimés et leurs spectrogrammes

Dans la figure 4.4, nous comparons l'évolution temporelle et le spectrogramme du signal d'entrée  $s(n)$  et le signal estimé  $e_1(n)$  obtenu par chaque algorithme (Dual MPPPSO, PPPSO, 2C-NLMS ( $\mu_{21} = 0.5, 1$ ) et 2C-VSSF) après convergence, dans ces simulations, les deux filtres adaptatifs  $w_{12}(n)$  et  $w_{21}(n)$  ont la même taille ( $L = 100$ ). Nous notons que ces signaux estimés sont obtenus par les quatre algorithmes combinés avec un détecteur d'activité vocale manuel (MDAV).

D'après la figure 4.4, on peut remarquer que les signaux de sortie obtenus par le Dual MPPPSO et le 2C-VSSF sont suffisamment débruités en les comparant avec ceux obtenus avec le PPPSO et le 2C-NLMS ( $\mu_{21} = 0.5, 1$ ), ce constat peut être confirmé en analysant les spectrogrammes des signaux de sortie, où on voit bien que le signal estimé par le MPPPSO est proche du signal original.

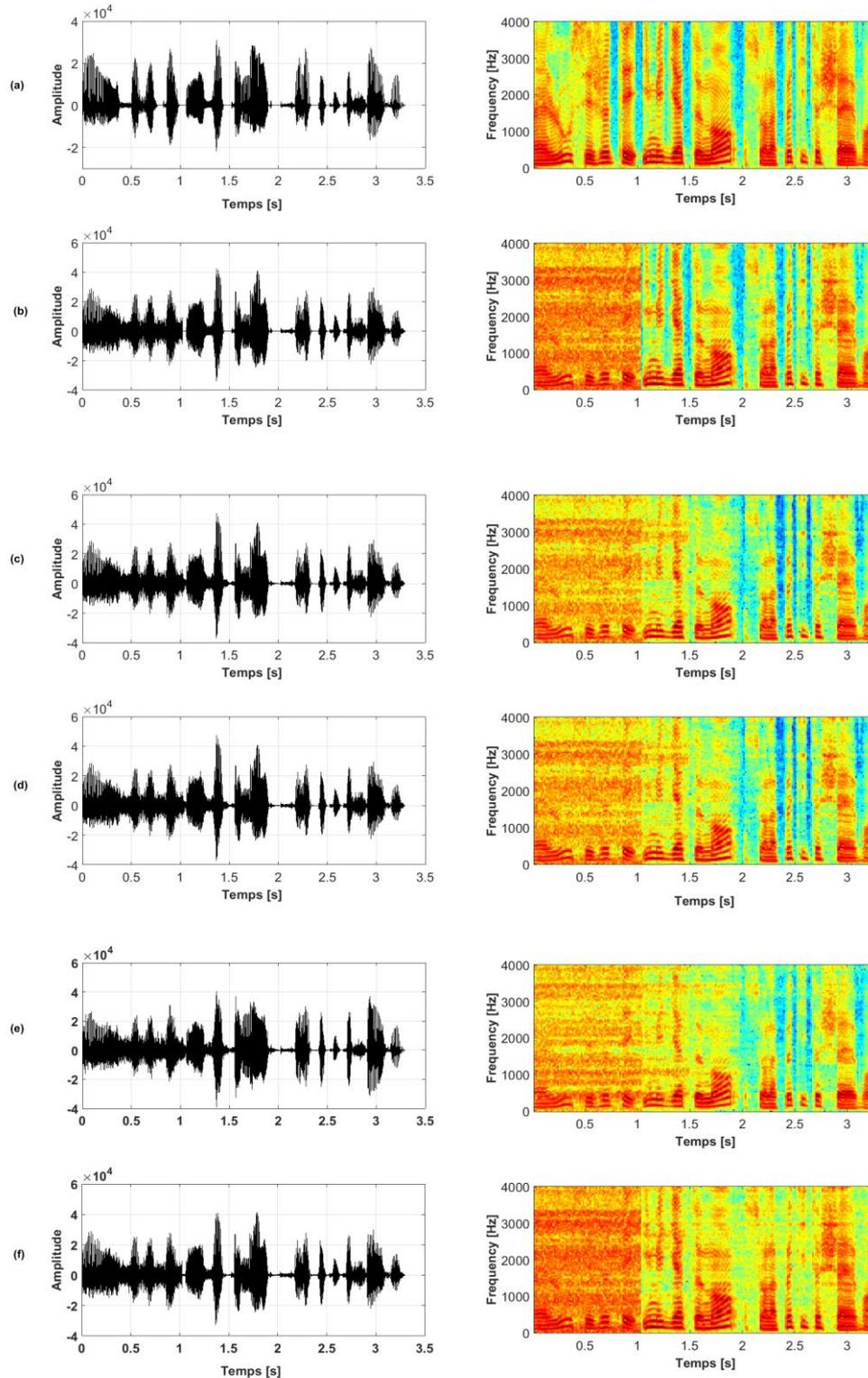


Figure. 4.4. Description temporelle et spectrogrammes de : (a) signal original (b) signal estimé obtenu par le Dual MPPSO (c) signal estimé obtenu par le 2C-VSSF (d) signal estimé obtenu par le 2C- FNLMS ( $\mu_{21}=1$ ) (e) signal estimé obtenu par le 2C- FNLMS ( $\mu_{21}=0.5$ ) (f) signal estimé obtenu par le PPPSO.  $RSB_1 = RSB_2 = -3dB$ . La deuxième source est un bruit blanc, la taille du filtre est  $L = 100$ .

Pour valider la supériorité de l'algorithme proposé par rapport aux autres algorithmes, nous allons utiliser d'autres pour qualifier objectivement les performances de chaque algorithme.

#### 4.6.4.2 Évaluation du System Mismatch (SM)

Pour faire une comparaison objective entre la vitesse de convergence obtenue par chaque algorithme, nous utilisons l'évaluation du System Mismatch (SM) du filtre adaptatif  $w_{21}(n)$  qui permet d'obtenir le signal de parole estimé.

La figure 4.5 représente l'évolution du System Mismatch obtenu par les quatre algorithmes Dual MPPPSO, PPPSO, 2C-NLMS ( $\mu_{21}=0.5,1$ ) et le 2C-VSSF et avec différents rapports signaux à bruits  $RSB_1 = RSB_2 = -3dB$ , et  $RSB_1 = RSB_2 = 0dB$ . La taille des filtres est  $L = 100$  figure 4.5 (en Haut) et  $L = 256$  figure 4.5 (en bas). D'après cette figure, nous pouvons voir clairement que la vitesse de convergence de l'algorithme proposé (Dual MPPPSO) est plus rapide que celle obtenue par les autres algorithmes dans toutes les situations.

#### 4.6.4.3 Évaluation du rapport signal sur bruit (SNR segmenté)

Dans le but d'évaluer la capacité de la réduction du bruit des algorithmes Dual MPPPSO, PPPSO, 2C-FNLMS ( $\mu_{21}=0.5,1$ ) et 2C-VSSF, on calcule le rapport du signal sur bruit segmenté correspondant à la première sortie (SegSNR). Le SegSNR est donné par la relation l'équation (4.5).

On note l'utilisation d'un MVAD qui permet d'estimer les valeurs de SegSNR seulement dans les périodes où il y a la parole.

Les résultats numériques obtenus par l'utilisation des quatre algorithmes sont représentés d'une manière graphique dans la figure 4.6.

À partir de la figure 4.6, on remarque que l'algorithme Dual MPPPSO a une importante supériorité de SegSNR en comparaison avec ceux obtenus par les autres algorithmes notamment pendant le régime de transition et cela dans toutes les situations (différent RSB d'entrée et taille de filtre).

On note aussi que le PPPSO est moins performant que les autres algorithmes dans tous les scénarios, à cause de la convergence prématurée de l'algorithme vers une solution optimale locale .

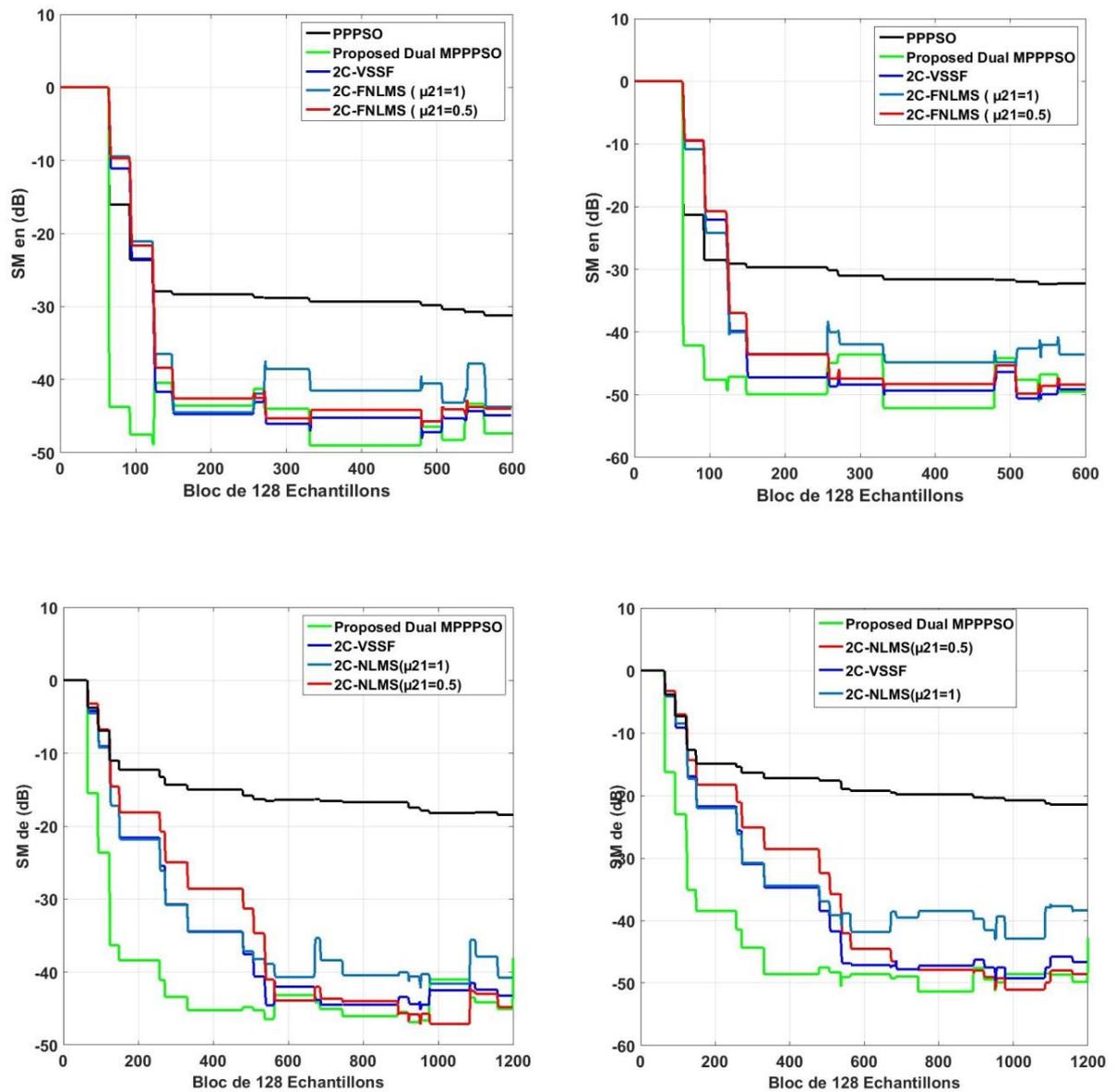


Figure 4.5 : Évaluation du système Mismatch du filtre adaptatif  $w_{21}(n)$  obtenue par Dual MPPPSO, PPPSO, 2C-FNLMS ( $\mu_{21} = 0.5, 1$ ) et 2C-VSSF. La deuxième source est un bruit blanc. la taille du filtre est  $L = 100$  [En haut] et  $L = 256$  [En bas]. [À gauche :  $RSB_1 = RSB_2 = -3dB$ ], [A droite :  $RSB_1 = RSB_2 = 0dB$ ].

#### 4.6.5 Test avec le bruit USASI

Dans la deuxième simulation, on considère le cas d'un bruit de type USASI utilisé dans le modèle de mélange avec différents RSB d'entrée, les différents paramètres

d'ajustement des algorithmes Dual MPPPSO, PPPSO, 2C-FNLMS ( $\mu_{21}=0.5,1$ ) et 2C-VSSF sont illustrés dans le tableau 4.5.

Tableau 4.3 : paramètres de simulation des algorithmes Dual MPPPSO, PPPSO, 2CNLMS et 2C-VSSF.

Algorithmes	Paramètres
PPPSO [104]	$N_y = 25; N_p = 5; c_1=c_2 = 2; w_p = 0.7; w_y = 0.7; P = 0.00055; MaxIt = 2000; \text{Espace de recherche } [-10,10]^n .$
2C-FNLMS [52]	$\mu_{12} = \mu_{21} = 0.5, 1 .$
2C-VSSF [94]	$\mu_{12}(0) = 0.2, \mu_{12,Max} = 2, \mu_{21}(0) = 0, \mu_{21,Max} = 1, \alpha_1 = 0.38, \alpha_2 = 0.88, \delta = 10^{-5}, \gamma = \xi = 10^{-6};$
Dual MPPPSO	$N_d = 18; N_y = 8; c_1=c_2 = 2; w_p = 0.7; w_y = 0.7; P = 0.00055; MaxIt = 2000; \text{Espace de recherche } [-10,10]^n .$

#### 4.6.5.1 Description temporelle des signaux estimés et leurs spectrogrammes

La figure 4.7 montre une comparaison entre l'évolution temporelle du signal de sorties des algorithmes Dual MPPPSO, PPPSO, 2C-FNLMS FNLMS ( $\mu_{21}=0.5,1$ ) et 2C-VSSF. En effet, nous avons utilisé une taille de filtre égale à 100 points ( $L = 100$ ) et un RSB d'entrée égale  $-3dB$  ( $RSB_1 = RSB_2 = -3dB$ ) les spectrogrammes de chaque sortie est représenté dans la même figure.

D'après la figure 4.7, on remarque que les signaux de parole estimés obtenue par les algorithmes PPPSO, 2C-FNLMS ( $\mu_{21}=0.5,1$ ) et 2C-VSSF sont mal débruités, alors qu'on voit clairement la supériorité de l'algorithme proposé du fait qu'il est très proche du signal de parole source.

#### 4.6.5.2 Évaluation du System Mismatch (SM)

Afin de tester la vitesse de convergence de chaque algorithme, on a évalué le critère de désajustement pour chaque algorithme avec un bruit USASI et deux différentes tailles du filtre (100 et 256 points).

Les résultats obtenus sont reportés dans la figure 4.8 ou on peut clairement observer la rapidité de convergence et le faible désajustement le plus bas de l'algorithme Dual MPPPSO par rapport aux trois autres algorithmes PPPSO, 2C-FNLMS et 2C-VSSF dans tous les scénarios

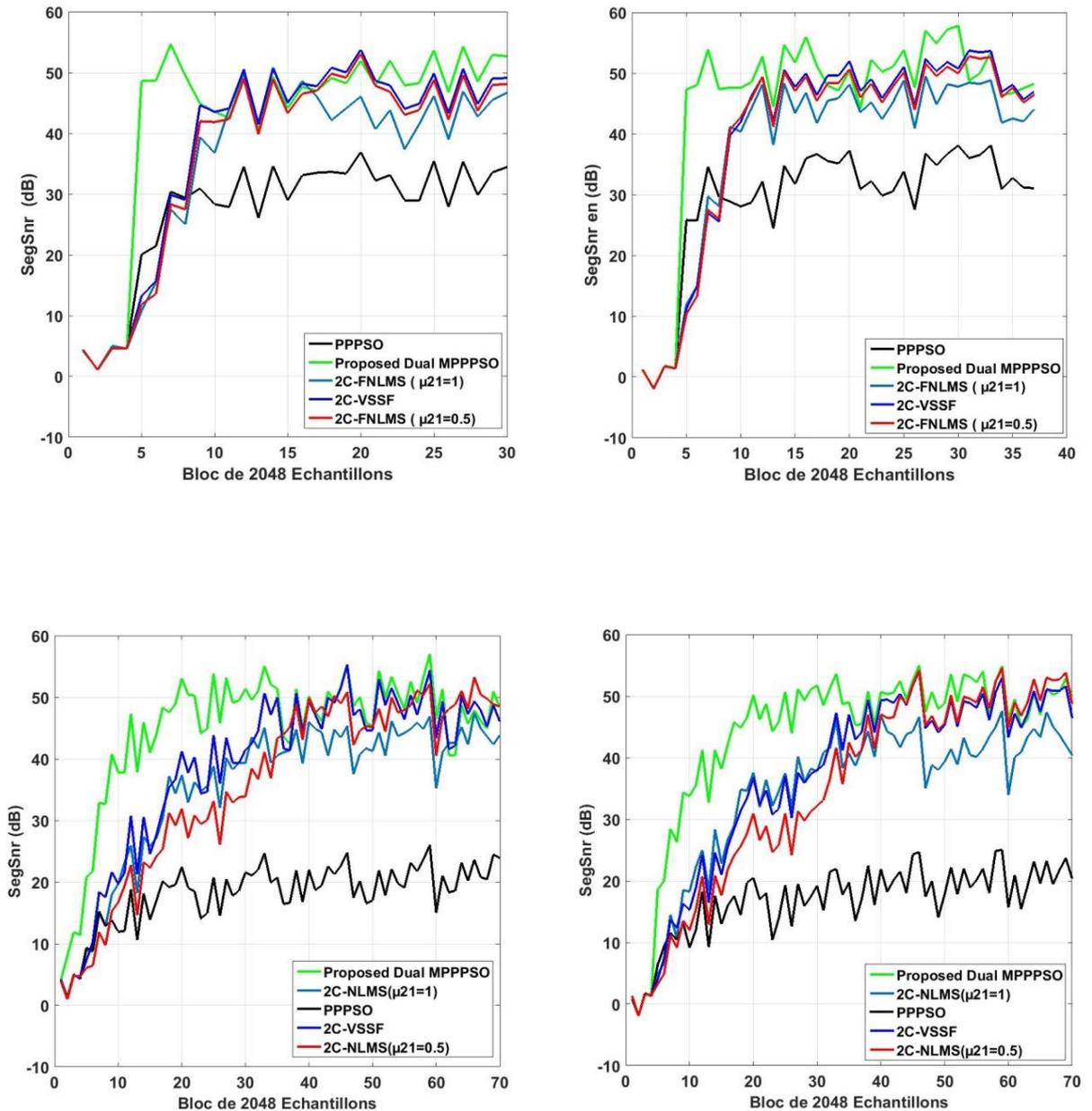


Figure 4.6 : Évaluation du SegSNR du filtre adaptatif  $w_{21}(n)$  obtenue par Dual MPPPSO, PPPSO, 2C-FNLMS ( $\mu_{21}=0.5,1$ ) et 2C-VSSF. La deuxième source est un bruit blanc. La taille du filtre est  $L = 100$  [En haut] et  $L = 256$  [En bas]. [À gauche :  $RSB_1 = RSB_2 = -3dB$ ], [À droite :  $RSB_1 = RSB_2 = 0dB$ ].

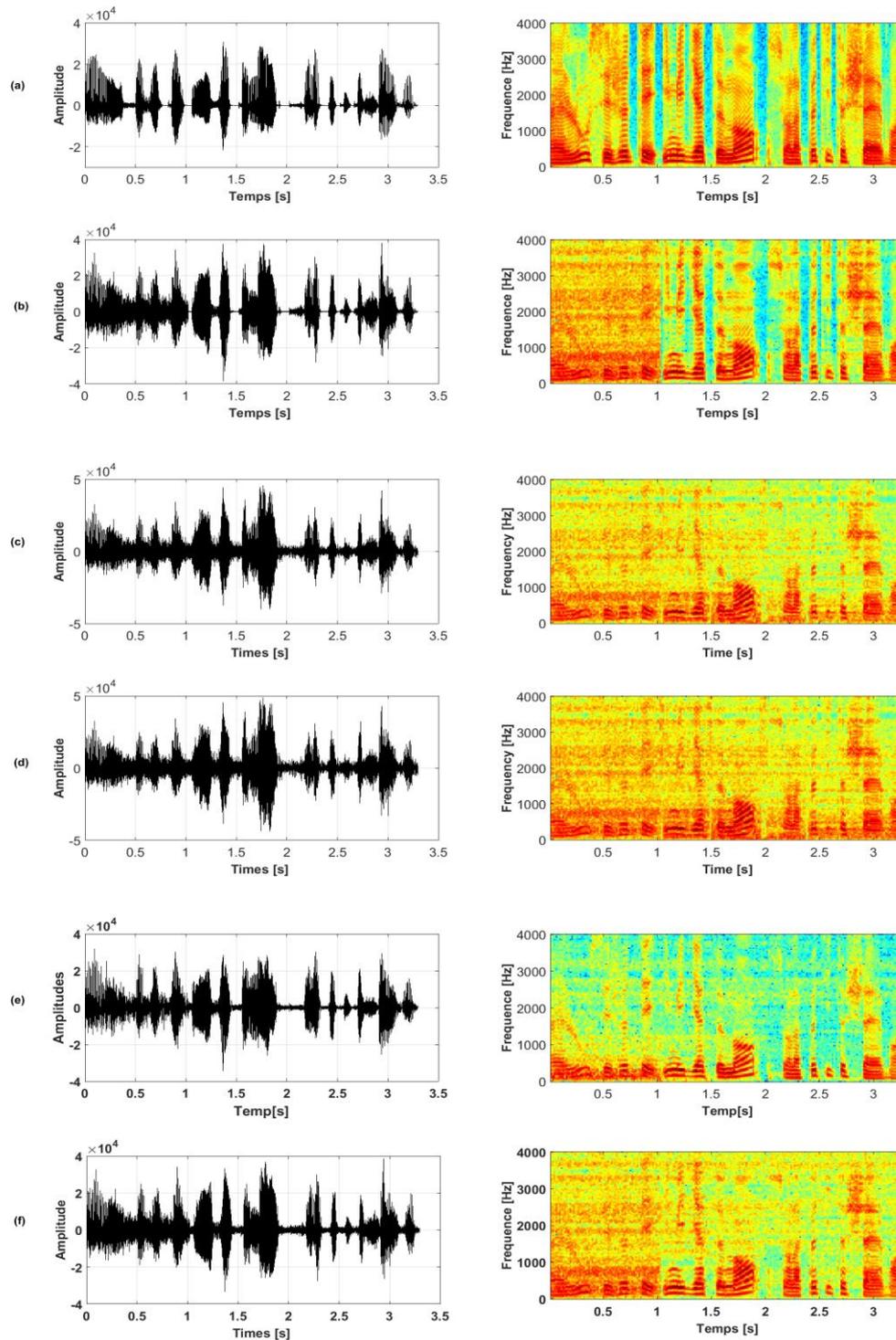


Figure. 4.7. Description temporelle et spectrogrammes de : (a) signal original (b) signal estimé obtenu par le Dual MPPSO (c) signal estimé obtenu par le 2C-VSSF (d) signal estimé obtenu par le 2C-FNLMS ( $\mu_{21}=1$ ) (e) signal estimé obtenu par le 2C-FNLMS ( $\mu_{21}=0.5$ ) (f) signal estimé obtenu par le PPPSO.  $RSB_1 = RSB_2 = -3dB$ . La deuxième source est un bruit USASI, la taille du filtre est  $L = 100$ .

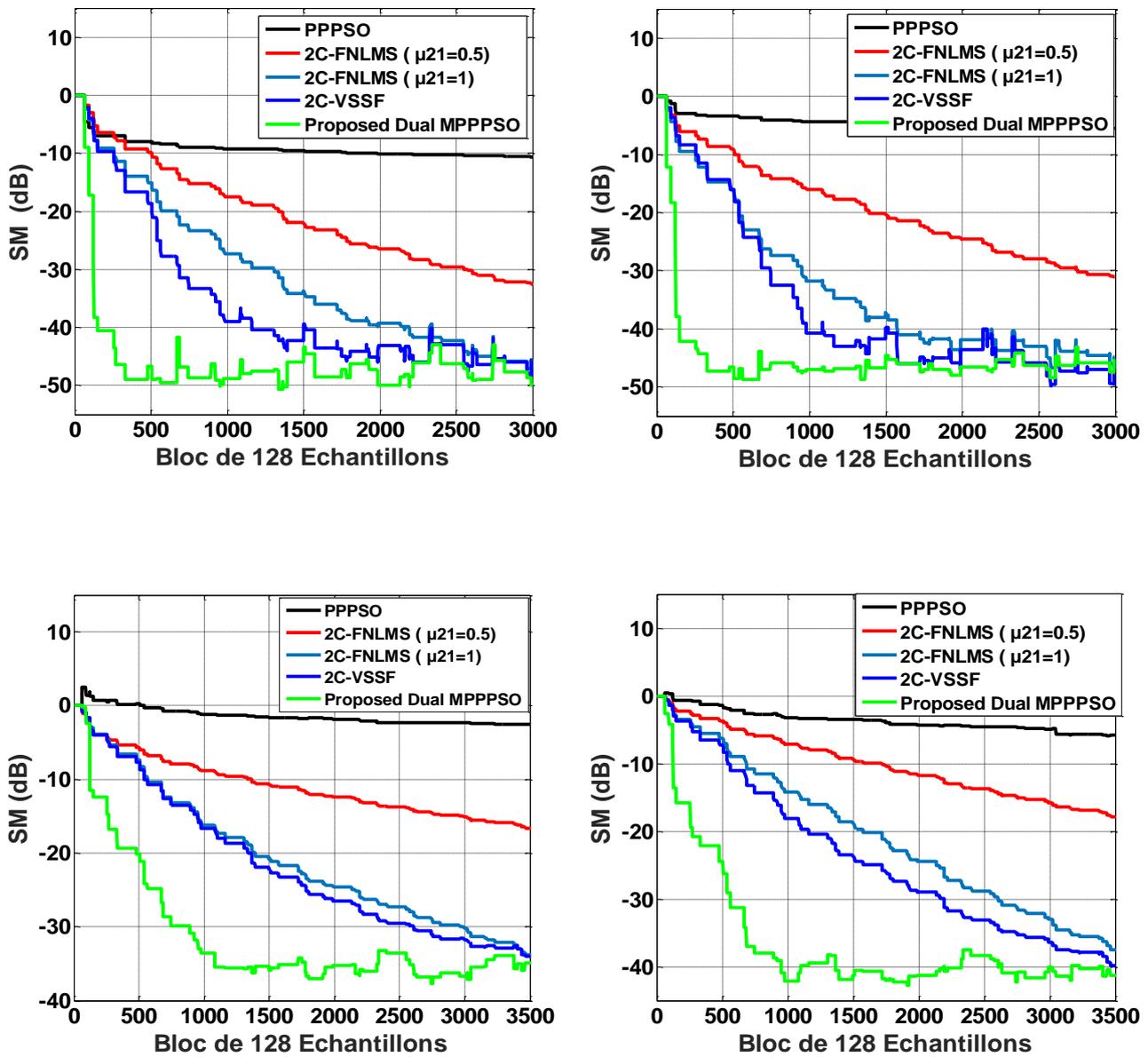


Figure 4.8 : Évaluation du système Mismatch du filtre adaptatif  $w_{21}(n)$  obtenue par Dual MPPPSO, PPPSO, 2C- FNLMS ( $\mu_{21}=0.5,1$ ) et 2C-VSSF. La deuxième source est un bruit USASI. La taille du filtre est  $L = 100$  [En haut] et  $L = 256$  [En bas]. [À gauche :  $RSB_1 = RSB_2 = -3\text{dB}$ ], [à droite :  $RSB_1 = RSB_2 = 0\text{dB}$ ].

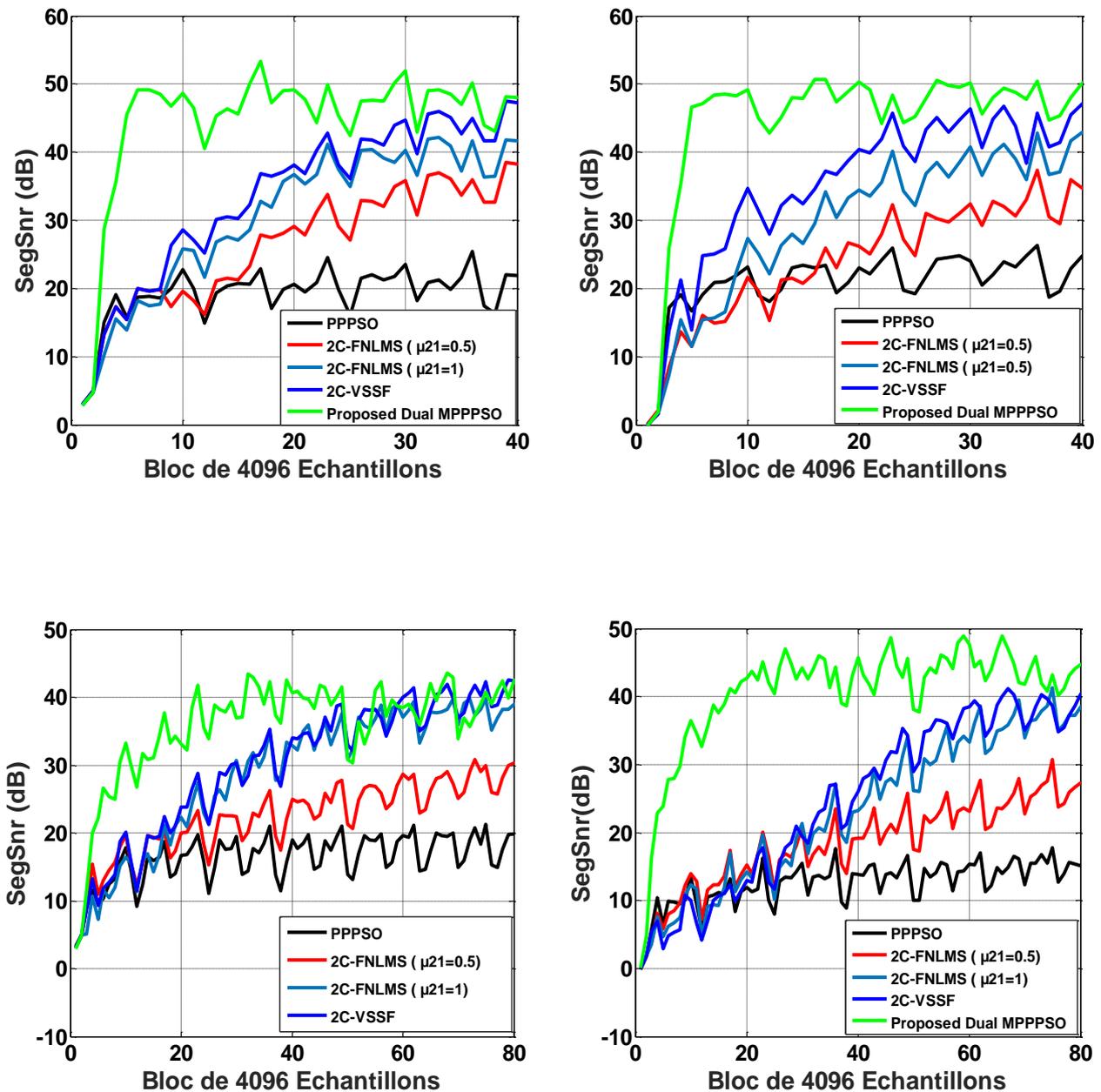


Figure 4.9 : Évaluation du SegSNR du filtre adaptatif  $w_{21}(n)$  obtenue par Dual MPPPSO, PPPSO, 2C- FNLMS ( $\mu_{21}=0.5,1$ ) et 2C-VSSF. La deuxième source est un bruit USASI. La taille du filtre est  $L = 100$  [En haut] et  $L = 256$  [En bas]. [À gauche :  $RSB_1 = RSB_2 = -3dB$ ], [à droite :  $RSB_1 = RSB_2 = 0dB$ ].

#### 4.6.5.3 Évaluation du rapport signal sur bruit (SNR segmenté)

Pour quantifier les caractéristiques de la réduction du bruit de l'algorithme proposé nous avons fait plusieurs simulations de l'annulation du bruit acoustique par la structure Forward bi-capteurs en utilisant les algorithmes Dual MPPPSO, PPPSO, 2C-FNLMS ( $\mu_{21} = 0.5, 1$ ) et 2C-VSSF, la figure 4.9 présente les valeurs du critère SegSnr obtenu. Dans ces simulations, on a utilisé un bruit USASI comme deuxième source, nous avons également utilisé deux différentes tailles de filtre (100 et 256 points) et deux différents RSB d'entrée  $0dB$  et  $-3dB$  ( $RSB_1 = RSB_2 = -3dB$  et  $RSB_1 = RSB_2 = 0dB$ ).

Tous les résultats obtenus du RSB confirment la supériorité de l'algorithme Dual MPPPSO par rapport à sa version originale et aux autres algorithmes.

D'après la figure 4.10, il est très clair que les valeurs du RSB de sortie obtenues par l'algorithme Dual MPPPSO sont les meilleures par rapport que celles obtenues par PPPSO, 2C-FNLMS ( $\mu_{21} = 0.5, 1$ ) et 2C-VSSF.

Finalement, d'après tous les résultats obtenus, nous avons validé la supériorité de cet algorithme en termes de vitesse de convergence, en minimisant aussi la distorsion du signal estimé (améliorer la qualité). Cet algorithme montre une bonne performance par rapport aux autres algorithmes.

#### 4.7 Conclusion

Dans ce chapitre, nous avons proposé un algorithme de séparation aveugle des sources implémenté dans une structure directe bi-capteurs. Il est basé sur les métaheuristiques dans le but d'annuler le bruit et améliorer la parole. L'algorithme proposé Dual MPPPSO offre l'avantage d'augmenter la chance d'éviter la convergence prématurée de l'algorithme vers des solutions locales en exploitant le mécanisme prédateur-proie pour maintenir tout l'essaim en mouvement. Cette propriété permet d'avoir de solution de bonne qualité, avec une vitesse de convergence rapide par rapport aux algorithmes PPPSO, 2C-FNLMS et 2C-VSSF et par conséquent ; une grande capacité à extraire le signal de parole même dans des environnements très bruités.

En effet, l'évaluation des performances montre l'avantage de l'algorithme proposé en tenant compte des différents critères d'évaluation nécessaires connus dans le domaine de rehaussement de parole et d'annulation de bruit.

## CONCLUSION GÉNÉRALE

Dans cette thèse, nous avons traité le problème de rehaussement de parole et de réduction de bruit acoustique, nous avons présenté des solutions qui permettent d'améliorer la qualité de parole par l'utilisation d'un nouvel algorithme métaheuristique implémenter dans une structure directe de séparation de source aveugle.

Nous avons entamé notre travail par une introduction portant sur les différentes approches d'optimisation à savoir les approches exactes, les approches heuristiques et métaheuristiques ainsi que leurs avantages et inconvénients avant de présenter leurs applications dans le domaine de réduction de bruit et définir les techniques et les structures utilisées dans l'application de rehaussement de parole, nous avons présenté les approches utilisées telles que les approches temporelles, les approches du domaine de transformation et les approches de sous-espace. Nous avons, également, présenté la technique de séparation de source aveugle combinée à la structure directe bi-capteurs nous avons présenté comment résoudre le problème à l'aide des algorithmes adaptatifs des gradients stochastiques et leur limite face à la grande taille de la réponse impulsionnelle acoustique.

Ce qui nous a amenés à utiliser dans ce travail des algorithmes métaheuristiques pour restaurer le signal de parole originale et améliorer sa qualité. Ce choix est justifié par le fait que les métaheuristiques permettent d'éviter d'une manière efficace les solutions locales. De ce fait, nous avons présenté une alternative pour implémenter les métaheuristiques dans les structures directes bi-capteurs.

Nous avons présenté une extension de l'algorithme d'optimisation par essaim particulière combiné a un mécanisme prédateur-proie qui permet d'éviter la convergence de tout l'essaim vers une solution prématurée ce qui influe sur la qualité du signal estimé, nous avons aussi proposé une nouvelle version modifiée de ce dernier afin de garder le plus possible la diversification de tout l'essaim et rendre l'algorithme plus solide en matière d'exploration et exploitation.

Nous avons utilisé ces deux algorithmes dans la structure directe bi-capteurs dans le but d'extraire et restaurer le signal de parole d'origine, les résultats obtenus ont illustré l'efficacité de l'algorithme proposé par rapport à ça version originale et deux des algorithmes classiques à savoir de l'algorithme du gradient stochastique

normalisé double canal (2C-FNLMS) et l'algorithme à pas d'adaptations variables doubles canal (2C-VSSF).

Nous avons employé l'algorithme dans plusieurs scénarios avec deux types de bruit de différente taille de filtre, le résultat à montrer la supériorité de l'algorithme proposé en termes de la vitesse de convergence et meilleur désajustement par rapport aux autres algorithmes.

Au terme de ces travaux, trois perspectives méritent d'être explorées :

- Il s'avère important serait, dans un premier temps, de valider l'algorithme par l'exploitation des données en temps réel.
- Il est intéressant d'utiliser les métaheuristiques dans le domaine fréquentiel.
- Envisager de faire une hybridation entre les métaheuristiques et les algorithmes du gradient afin d'essayer d'améliorer la qualité de la solution et réduire le temps de calcul.

## LISTE DES SYMBOLES ET DES ABRÉVIATIONS

## ABRÉVIATIONS

2C-VSSF : Two-channel variable Step-Sizes Forward  
2C-FLMS : Two-channel Forward least-mean square  
2C-FNLMS : Two-channel Forward normalized least-mean square  
2C-LMS : Two-channel least-mean square  
2C-NLMS : Two-channel normalized least-mean square  
ABC : Artificial bee colony algorithm  
ANC : Adaptive noise cancellation  
AR : Autoregressive  
BA : Bat algorithm  
BSS : Blind source separation  
DA : Differentiel evolution  
DAV : Détecteur d'activité vocale  
DSP : Densité spectrale de puissance  
DOA : Direction Of Arrivals  
EEG : Électroencéphalographie  
EQM : Erreur quadratique moyenne  
FBSS : Forward blind source separation  
GA : Genitic algorithm  
GSA : Gravitational Search algorithm  
GWO : Grey wolf optimizer  
iFFT: Inverse Fast Fourier Transform  
IIR : Infinite impulse response  
LMS : Least-mean square  
LPC: linear predictive coding  
MSE : Mean square error  
NLMS : Normalized least-mean square  
PPPSO : Predator prey particle swarm optimization

PSO : Particle swarm optimization

PSOGSA : Hybrid Particle Swarm Optimization and Gravitational Search

MPPPSO : Modified predator prey particle swarm optimization

RIF : Réponse impulsionnelle finie

RSB : Rapport signal à bruit

RSB1 : Rapport signal à bruit d'entrée du premier signal bruité

RSB2 : Rapport signal à bruit d'entrée du deuxième signal bruité

SAS : Séparation aveugle de source

SM : Système Mismatch

SNR : Signal-to-noise ratio

SSA : Soustraction spectrale d'amplitude

SSP : Soustraction spectrale de puissance

TFD : Transformée de Fourier discrète

USASI : United state of America Standard Institute

VAD : Voice Activity Detector

VSS-LMS : Variable step-size least-mean square

VSS-NLMS : Variable step-size normalized least-mean square

## SYMBOLES

$C$  : Nombre des capteurs

$f$  : Indice de fréquence

$L$  : Taille des filtres  $M$  : Indice de retard

$n$  : Indice de temps

$Q$  : Nombre des sources

$T$  : Nombre d'échantillons de la trame

$w_{12}$  : Filtre adaptatif du premier canal

$w_{21}$  : Filtre adaptatif du deuxième canal.

$L_f$  : Longueur de la trame.

$N_d$  : Nombre des prédateurs.

$N_y$  : Nombre des proies.

$maxit$  : Maximum nombre d'itérations.

$g$  : Meilleure position globale.

$c_1$  : Paramètre de cognitivité.

$c_2$  : Paramètre social

$w_d$  : Poids d'inertie d'un prédateur.

$w_y$  : Poids d'inertie d'une proie

$w_{max}$  : Valeur maximale d'inertie des prédateurs.

$w_{min}$  : Valeur minimale d'inertie des proies.

$P$  : Variable binaire

$x_{di}$  : La position du  $i^{ième}$  prédateur.

$v_{di}$  : La vitesse du  $i^{ième}$  prédateur.

$b_{di}$  : La meilleure performance du  $i^{ième}$  prédateur

$x_{yj}$  : La position de la  $j^{ième}$  proie.

$v_{yj}$  : La position de la  $j^{ième}$  proie.

$b_{yj}$  : La meilleure performance de la  $j^{ième}$  proie

$f(.)$  : la fonction objective.

$p_1$  : Premier signal observation

$p_2$  : Deuxième signal observation

$e_1$  : Signal de la première source estimée.

$e_2$  : Signal de la deuxième source estimée.

$\alpha_1$  et  $\alpha_2$  : Petite constante positive .

$\xi, \gamma$  : constante positive très petite.

$\mu_{12}(n)$  et  $\mu_{21}(n)$  : Pas d'adaptation fixe

$m$  : Indice du Délai

## RÉFÉRENCES

- [1] P. Borne, D. Popescu, F. G. Filip, D. Stefanoiu, and B. Dubuisson, *Optimization in Engineering Sciences: Exact Methods*. Wiley, 2013.
- [2] O. Bozorg-Haddad, M. Solgi, and H. A. Loáiciga, *Meta-Heuristic and Evolutionary Algorithms for Engineering Optimization*, vol. 294. John Wiley & Sons, 2017.
- [3] R. Shreeve, H. Ratschek, and J. Rokne, *New Computer Methods for Global Optimization*, vol. 73, no. 466. New York, NY, USA: Halsted Press, 1989.
- [4] J. Puchinger and G. R. Raidl, “Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification,” in *Lecture Notes in Computer Science*, 2005, vol. 3562, no. PART II, pp. 41–53.
- [5] H. H. Hoos and T. Stützle, *AAAI-04 Tutorial: Stochastic Local Search - Foundations and Applications*. Elsevier, 2004.
- [6] E. Alba, *Parallel Metaheuristics: A New Class of Algorithms*, vol. 47. John Wiley & Sons, 2005.
- [7] F. Glover, “Future paths for integer programming and links to artificial intelligence,” *Comput. Oper. Res.*, vol. 13, no. 5, pp. 533–549, 1986.
- [8] C. Blum and A. Roli, “Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison,” *ACM Comput. Surv.*, vol. 35, no. 3, pp. 268–308, 2003.
- [9] E. G. Talbi, *Metaheuristics: From Design to Implementation*, vol. 74. John Wiley & Sons, 2009.
- [10] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science (80-. )*, vol. 220, no. 4598, pp. 671–680, 1983.

- [11] V. Černý, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *J. Optim. Theory Appl.*, vol. 45, no. 1, pp. 41–51, 1985.
- [12] P. Hansen, "The Speepest Ascent Mildest Descent Heuristic for Propositional Programming," in *Congress on Numerical Methods in Combinatorial Optimization*, 1986, pp. 70–145.
- [13] J. P. Marques-Silva and K. A. Sakallah, "GRASP: A search algorithm for propositional satisfiability," *IEEE Trans. Comput.*, vol. 48, no. 5, pp. 506–521, 1999.
- [14] P. Hansen and N. Mladenović, "Variable neighborhood search," *Handb. Heuristics*, vol. 1–2, no. 11, pp. 759–787, 2018.
- [15] H. R. Lourenço, O. C. Martin, and T. Stützle, "Iterated local search," in *Handbook of metaheuristics*, Springer, 2003, pp. 320–353.
- [16] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, 1995, vol. 4, pp. 1942–1948.
- [17] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014.
- [18] X. S. Yang, "A new metaheuristic Bat-inspired Algorithm," in *Studies in Computational Intelligence*, vol. 284, Springer, 2010, pp. 65–74.
- [19] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360)*, 1998, pp. 69–73.
- [20] S. R. Diniz, "Paulo,—Adaptive Filtering: Algorithms and Practical Implementationsll." Kluwer Academic Publisher, 2008.
- [21] N. Wiener, *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*. MIT press Cambridge, 1950.

- [22] P. S. R. Diniz and P. S. R. Diniz, *Fundamentals of Adaptive Filtering*. John Wiley & Sons, 2020.
- [23] D. G. Manolakis, V. K. Ingle, and S. M. Kogon, *Statistical and adaptive signal processing: spectral estimation, signal modeling, adaptive filtering, and array processing*. 2000.
- [24] S. Rogers, *Adaptive filter theory*, vol. 4, no. 11. Pearson Education India, 1996.
- [25] B. Widrow and M. E. Hoff, "Adaptive switching circuits," Stanford Univ Ca Stanford Electronics Labs, 1989.
- [26] B. Widrow, M. G. Larimore, C. R. Johnson, and J. M. Mccool, "Stationary and Nonstationary Learning Characteristics of the LMS Adaptive Filter," in *Proceedings of the IEEE*, vol. 64, no. 8, Springer, 1976, pp. 1151–1162.
- [27] S. S. Haykin, *Adaptive filter theory*. Pearson Education India, 2008.
- [28] B. Widrow, J. McCool, M. G. Larimore, and C. R. Johnson, "Stationary and nonstationary learning characteristics of the LMS adaptive filter," in *Aspects of signal processing*, Springer, 1977, pp. 355–393.
- [29] B. Widrow *et al.*, "Adaptive Noise Cancelling: Principles and Applications," *Proc. IEEE*, vol. 63, no. 12, pp. 1692–1716, 1975.
- [30] H. Alaeddine, "Application de la transformée en nombres entiers à la conception d'algorithmes de faible complexité pour l'annulation d'échos acoustiques." 2007.
- [31] W. Der Chang, "Two-dimensional fractional-order digital differentiator design by using differential evolution algorithm," *Digit. Signal Process. A Rev. J.*, vol. 19, no. 4, pp. 660–667, 2009.
- [32] M. K. Ahirwal, A. Kumar, and G. K. Singh, "Adaptive filtering of EEG/ERP through Bounded Range Artificial Bee Colony (BR-ABC) algorithm," *Digit. Signal Process. A Rev. J.*, vol. 25, no. 1, pp. 164–172, 2014.
- [33] A. Neubauer, "Non-linear adaptive filters based on genetic algorithms with

- applications to digital signal processing,” in *Proceedings of the IEEE Conference on Evolutionary Computation*, 1995, vol. 2, pp. 527–532.
- [34] K. V. V. S. R. Prajna Kunche, *Metaheuristic Applications to Speech Enhancement*. Springer, 2016.
- [35] M. Geravanchizadeh and L. B. Asl, “Asexual Reproduction-based Adaptive Quantum Particle Swarm Optimization algorithm for dual-channel speech enhancement,” in *Final Program and Abstract Book - 4th International Symposium on Communications, Control, and Signal Processing, ISCCSP 2010*, 2010, pp. 1–4.
- [36] L. B. Asl and V. M. Nezhad, “Speech enhancement using particle swarm optimization techniques,” in *2010 International Conference on Measuring Technology and Mechatronics Automation, ICMTMA 2010*, 2010, vol. 3, pp. 441–444.
- [37] L. B. Asl and V. M. Nezhad, “Improved particle swarm optimization for dual-channel speech enhancement,” in *2010 International Conference on Signal Acquisition and Processing, ICSAP 2010*, 2010, pp. 13–17.
- [38] S. Ghalami Osgouei and M. Geravanchizadeh, “Dual-channel speech enhancement based on a hybrid particle swarm optimization algorithm,” in *2010 5th International Symposium on Telecommunications, IST 2010*, 2010, pp. 873–877.
- [39] P. Kunche, G. Sasi Bhushan Rao, K. V. V. S. Reddy, and R. Uma Maheswari, “A new approach to dual channel speech enhancement based on hybrid PSO-GSA,” *Int. J. Speech Technol.*, vol. 18, no. 1, pp. 45–56, 2014.
- [40] K. Prajna, G. Sasibhushana Rao, K. V. V. S. Reddy, and R. Uma Maheswari, “Application of Bat Algorithm in dual channel speech enhancement,” in *International Conference on Communication and Signal Processing, ICCSP 2014 - Proceedings*, 2014, pp. 1457–1461.
- [41] J. Hérault, C. Jutten, and B. Ans, “Détection de grandeurs primitives dans un message composite par une architecture de calcul neuromimétique en

- apprentissage non supervisé,” in *10<sup>th</sup> Colloque sur le traitement du signal et des images, FRA, 1985*, 1985, vol. 3, pp. 1017–1022.
- [42] C. Jutten and J. Herault, “Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture,” *Signal Processing*, vol. 24, no. 1, pp. 1–10, 1991.
- [43] M. Feng and K. D. Kammeyer, “Blind source separation for communication signals using antenna arrays,” in *ICUPC 1998 - IEEE 1998 International Conference on Universal Personal Communications, Conference Proceedings*, 1998, vol. 1, pp. 665–669.
- [44] A. J. Bell and T. J. Sejnowski, “An information-maximization approach to blind separation and blind deconvolution,” *Neural Comput.*, vol. 7, no. 6, pp. 1129–1159, 1995.
- [45] S. I. Amari and J. F. Cardoso, “Blind source separation-semiparametric statistical approach,” *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2692–2700, 1997.
- [46] M. Pal, R. Roy, J. Basu, and M. S. Bepari, “Blind source separation: A review and analysis,” in *2013 International Conference Oriental COCOSDA Held Jointly with 2013 Conference on Asian Spoken Language Research and Evaluation, O-COCOSDA/CASLRE 2013*, 2013, pp. 1–5.
- [47] C. Jutten, L. Nguyen Thi, E. Dijkstra, E. Vittoz, and J. Caelen, “Blind separation of sources: an algorithm for separation of convolutive mixtures,” in *International Signal Processing Workshop on Higher Order Statistics*, 1991, pp. 273–276.
- [48] M. Ali, “Contribution a La Separation Aveugle De Sources.” Grenoble INPG, 1997.
- [49] H. Boumaraf, “Séparation Aveugle de Mélanges Convolutifs de Sources.” 2001.
- [50] H. Boumaraf, D. T. Pham, and C. Servière, “Blind separation of convolutive mixture of speech signals,” in *13th European Signal Processing Conference*,

*EUSIPCO 2005*, 2005, pp. 1051–1054.

- [51] S. Van Gerven, “Adaptive noise cancellation and signal separation with applications to speech enhancement,” 1996.
- [52] S. Van Gerven and D. Van Compernelle, “Feedforward and Feedback in a Symmetric Adaptive Noise Canceller: Stability Analysis in a Simplified Case,” in *Signal Processing VI, Theories and Applications*, 1992, pp. 1081–1084.
- [53] E. Vincent, T. Virtanen, and S. Gannot, *Audio Source Separation and Speech Enhancement*. John Wiley & Sons, 2018.
- [54] S. F. Boll, “Suppression of Acoustic Noise in Speech Using Spectral Subtraction,” *IEEE Trans. Acoust.*, vol. 27, no. 2, pp. 113–120, 1979.
- [55] P. J. Bloom and G. D. Cain, “Evaluation of two-input speech dereverberation techniques,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 1982, vol. 1982-May, pp. 164–167.
- [56] W. M. Kushner, V. Goncharoff, C. Wu, V. Nguyen, and J. N. Damosoulakis, “Effects of subtractive-type speech enhancement/noise reduction algorithms on parameter estimation for improved recognition and coding in high noise environments,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 1989, vol. 1, pp. 211–214.
- [57] J. S. Lim and A. V. Oppenheim, “Enhancement and Bandwidth Compression of Noisy Speech,” *Proc. IEEE*, vol. 67, no. 12, pp. 1586–1604, 1979.
- [58] M. Berouti, R. Schwartz, and J. Makhoul, “Enhancement of speech corrupted by acoustic noise,” in *ICASSP’79. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1979, vol. 4, pp. 208–211.
- [59] B. E. Mukul Bhatnagar, “a Modified Spectral Subtraction Method Combined With Perceptual Weighting for Speech Enhancement,” *Perform. Eval.*, p. 110, 2002.
- [60] R. M. Udrea, N. D. Vizireanu, and S. Ciochina, “An improved spectral

- subtraction method for speech enhancement using a perceptual weighting filter,” *Digit. Signal Process. A Rev. J.*, vol. 18, no. 4, pp. 581–587, 2008.
- [61] N. Virag, “Single channel speech enhancement based on masking properties of the human auditory system,” *IEEE Trans. Speech Audio Process.*, vol. 7, no. 2, pp. 126–137, 1999.
- [62] D. O. Walter, “A posteriori ‘Wiener filtering’ of average evoked responses.,” *Electroencephalogr. Clin. Neurophysiol.*, p. Suppl-27, 1968.
- [63] L. Lin, W. H. Holmes, and E. Ambikairajah, “Speech enhancement based on a perceptual modification of Wiener filtering,” *7th Int. Conf. Spok. Lang. Process. ICSLP 2002*, vol. 38, no. 23, pp. 781–784, 2002.
- [64] J. Benesty, J. Chen, and Y. A. Huang, “Linear Prediction,” in *Springer Handbooks*, Springer, 2008, pp. 121–134.
- [65] A. Amehraye, D. Pastor, and A. Tamtaoui, “Perceptual improvement of Wiener filtering,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2008, pp. 2081–2084.
- [66] C. C. Hsu, T. E. Lin, J. H. Chen, and T. S. Chi, “Spectro-temporal subband Wiener filter for speech enhancement,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2012, pp. 4001–4004.
- [67] J. Sohn, “A statistical model-based voice activity detection,” *IEEE Signal Process. Lett.*, vol. 6, no. 1, pp. 1–3, 1999.
- [68] S. G. Tanyer and H. Ozer, “Voice activity detection in nonstationary noise,” *IEEE Trans. speech audio Process.*, vol. 8, no. 4, pp. 478–482, 2000.
- [69] R. Martin, “An Efficient Algorithm to Estimate The Instantaneous SNR of Speech Signals,” in *Third European Conference on Speech Communication and Technology (EUROSPEECH '93)*, 1993, pp. 1093–1096.
- [70] R. Martin, “Spectral subtraction based on minimum statistics,” *Power*, vol. 6, p. 8, 1994.

- [71] B. A. Pearlmutter and L. C. Parra, "Maximum likelihood blind source separation: A context-sensitive generalization of ICA," in *Advances in Neural Information Processing Systems*, 1997, pp. 613–619.
- [72] C. Jutten and J. Karhunen, "Advances in Nonlinear Blind Source Separation," in *Proc of the 4th Int Symp on Independent Component Analysis and Blind Signal Separation ICA2003*, 2003, vol. 4, no. April, pp. 245–256.
- [73] L. Parra and C. Spence, "Convolutional blind separation of non-stationary sources," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 3, pp. 320–327, 2000.
- [74] R. H. L. Marcel Joho, Heinz Mathis, "Overdetermined Blind Source Separation: Using More Sensors than Source Signals in a Noisy Mixture," in *Ica 2000*, 2000, pp. 19–22.
- [75] N. Q. K. Duong, E. Vincent, and R. Gribonval, "Under-determined reverberant audio source separation using a full-rank spatial covariance model," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 18, no. 7, pp. 1830–1840, 2010.
- [76] S. Makino, T.-W. Lee, and H. Sawada, *Blind Speech Separation*, vol. 4. Springer, 2007.
- [77] K. Torkkola, "Blind separation of delayed sources based on information maximization," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 1996, vol. 6, pp. 3509–3512.
- [78] T. W. Lee, A. J. Bell, and R. H. Lambert, "Blind separation of delayed and convolved sources," in *Advances in Neural Information Processing Systems*, 1997, pp. 758–764.
- [79] R. Mukai, S. Araki, and S. Makino, "Separation and dereverberation performance of frequency domain blind source separation for speech in a reverberant environment," in *EUROSPEECH 2001 - SCANDINAVIA - 7th European Conference on Speech Communication and Technology*, 2001, pp. 2599–2602.

- [80] R. Mukai, S. Araki, H. Sawada, and S. Makino, "Evaluation of separation and dereverberation performance in frequency domain blind source separation," *Acoust. Sci. Technol.*, vol. 25, no. 2, pp. 119–126, 2004.
- [81] P. J. Smaragdis, "Information Theoretic Approaches to Source Separation." Massachusetts Institute of Technology, pp. 1–88, 1997.
- [82] P. Smaragdis, "Blind separation of convolved mixtures in the frequency domain," *Neurocomputing*, vol. 22, no. 1–3, pp. 21–34, 1998.
- [83] N. Mitianoudis and M. E. Davies, "Audio source separation of convolutive mixtures," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 5, pp. 489–497, 2003.
- [84] H. Bousbia-Salah, A. Belouchrani, and K. Abed-Meraim, "Jacobi-like algorithm for blind signal separation of convolutive mixtures," *Electron. Lett.*, vol. 37, no. 16, pp. 1049–1050, 2001.
- [85] M. Djendi, "Advanced techniques for two-microphone noise reduction in mobile communications," *Univ. Rennes, Fr.*, 2010.
- [86] M. Djendi, P. Scalart, and A. Gilloire, "Analysis of two-sensors forward BSS structure with post-filters in the presence of coherent and incoherent noise," *Speech Commun.*, vol. 55, no. 10, pp. 975–987, 2013.
- [87] M. Feder, A. V. Oppenheim, and E. Weinstein, "Maximum Likelihood Noise Cancellation Using The Em Algorithm," *IEEE Trans. Acoust.*, vol. 37, no. 2, pp. 204–216, 1989.
- [88] W. A. Harrison, J. S. Lim, and E. Singer, "A New Application of Adaptive Noise Cancellation," *IEEE Trans. Acoust.*, vol. 34, no. 1, pp. 21–27, 1986.
- [89] M. J. Al-Kindi and J. Dunlop, "Improved adaptive noise cancellation in the presence of signal leakage on the noise reference channel," *Signal Processing*, vol. 17, no. 3, pp. 241–250, 1989.
- [90] S. Ikeda and A. Sugiyama, "An adaptive noise canceller with low signal-distortion in the presence of crosstalk," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E82-A, no. 8, pp. 1517–1524, 1999.

- [91] M. Djendi, A. Gilloire, and P. Scalart, "New frequency domain post-filters for noise cancellation using two closely spaced microphones," in *European Signal Processing Conference*, 2007, pp. 218–221.
- [92] H. L. NGUYEN THI, C. Jutten, and J. Caelen, "Séparation aveugle de parole et de bruit dans un mélange convolutif," in *13° Colloque sur le traitement du signal et des images, FRA*, 1991, 1991.
- [93] M. Djendi and R. Bendoumia, "A new efficient two-channel backward algorithm for speech intelligibility enhancement: A subband approach," *Appl. Acoust.*, vol. 76, pp. 209–222, Feb. 2014.
- [94] R. Bendoumia and M. Djendi, "Two-channel variable-step-size forward-and-backward adaptive algorithms for acoustic noise reduction and speech enhancement," *Signal Processing*, vol. 108, pp. 226–244, 2015.
- [95] A. Vallauri, "L'étude et le développement de méthodes de reconnaissance de la parole et de réduction du bruit, et application." Ph. D. thesis, 1992.
- [96] M. Djendi and A. Sayoud, "A new dual subband fast NLMS adaptive filtering algorithm for blind speech quality enhancement and acoustic noise reduction," *Int. J. Speech Technol.*, vol. 22, no. 2, pp. 391–406, 2019.
- [97] R. Henni, M. Djendi, and M. Djebbari, "A new efficient two-channel fast transversal adaptive filtering algorithm for blind speech enhancement and acoustic noise reduction," *Comput. Electr. Eng.*, vol. 73, pp. 349–368, 2019.
- [98] P. C. Loizou, *Speech Enhancement: Theory and Practice*, 2nd ed. Boca Raton, FL, USA: CRC Press, Inc., 2013.
- [99] G. C. Burnett, N. J. Petit, A. M. Asseily, and A. E. Einaudi, "Microphone and voice activity detection (VAD) configurations for use with communication systems." Google Patents, Jun-2013.
- [100] D. K. Freeman, G. Cosier, C. B. Southcott, and I. Boyd, "The voice activity detector for the Pan-European digital cellular mobile telephone service," in *International Conference on Acoustics, Speech, and Signal Processing*, 1989, pp. 369–372.

- [101] H. Garudadri, "Method and apparatus for transmitting speech activity in distributed voice recognition systems." Google Patents, 10-Apr-2007.
- [102] B. Nakisa and M. N. Rastgoo, "A survey: Particle swarm optimization based algorithms to solve premature convergence problem," *J. Comput. Sci.*, vol. 10, no. 9, pp. 1758–1765, 2014.
- [103] A. Silva, A. Neves, and E. Costa, "An empirical comparison of particle swarm and predator prey optimisation," in *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, 2002, vol. 2464, pp. 103–110.
- [104] M. Higashitani, A. Ishigame, and K. Yasuda, "Particle swarm optimization considering the concept of predator-prey behavior," in *2006 IEEE Congress on Evolutionary Computation, CEC 2006*, 2006, pp. 434–437.
- [105] W. Jang, H. Kang, and B. Lee, "Optimized fuzzy clustering by predator prey particle swarm optimization," in *International Conference on Intelligent Computing*, 2007, pp. 370–379.
- [106] H. Duan, S. Li, and Y. Shi, "Predator-prey brain storm optimization for DC brushless motor," *IEEE Trans. Magn.*, vol. 49, no. 10, pp. 5336–5340, 2013.
- [107] H. Duan, P. Li, and Y. Yu, "A predator-prey particle swarm optimization approach to multiple UCAV air combat modeled by dynamic game theory," *IEEE/CAA J. Autom. Sin.*, vol. 2, no. 1, pp. 11–18, 2015.
- [108] Y. Zhang *et al.*, "Multivariate approach for Alzheimer's disease detection using stationary wavelet entropy and predator-prey particle swarm optimization," *J. Alzheimer's Dis.*, vol. 65, no. 3, pp. 855–869, 2018.
- [109] R. Bendoumia, M. Djendi, and A. Guessoum, "Two-Channel Acoustic Noise Reduction by New Backward Normalized Decorrelation Algorithm," in *International Conference on Electrical Engineering and Control Applications*, 2017, pp. 464–478.
- [110] K. Kondo, *Subjective Quality Measurement of Speech: Its Evaluation, Estimation, and Estimation*. Springer Science & Business Media, 2012.

- [111] Y. Hu and P. C. Loizou, "Evaluation of objective quality measures for speech enhancement," *IEEE Trans. Audio. Speech. Lang. Processing*, vol. 16, no. 1, pp. 229–238, 2007.
- [112] K. Prajna, K. V. V. S. Reddy, G. S. B. Rao, and R. U. Maheswari, "A comparative study of BA, APSO, GSA, hybrid PSOGSA and SPSO in dual channel speech enhancement," *Int. J. Speech Technol.*, vol. 18, no. 4, pp. 663–671, 2015.
- [113] M. Djendi, A. Gilloire, and P. Scalart, "Noise cancellation using two closely spaced microphones: Experimental study with a specific model and two adaptive algorithms," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 2006*, vol. 3, pp. III–III.