

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE



UNIVERSITE SAAD DAHLEB-BLIDA-  
FACULTE DES SCIENCES DE L'INGENIEUR



Département D'aéronautique

Mémoire de fin d'études en vue de l'obtention du diplôme  
d'ingénieur d'état en Aéronautique.  
Spécialité : Installations.

# Thème

Synthèse de Filtre Numérique RII  
En utilisant la méthode d'Optimisation  
par Essaim Particulaire  
« OEP ».

**Promoteur :**

**Mr: BENOUARED.A.**

**Réalisé par :**

**\* MECHRAOUI Malika.  
\* TOBAL Sabah.**

**ANNEE UNIVERSITAIRE 2010-2011.**

# REMERCIEMENTS

*Nous remercions tout d'abord Le Grand Dieu  
Le tout puissant de nous avoir donné la volonté Et le  
courage pour accomplir ce mémoire. Nous tenons à  
exprimer nous remerciments Les plus sincères à ceux et  
celles de près ou de Loin ont contribués à l'élaboration  
de ce travail.*

*Tout particulièrement à Notre Promoteur Mr :  
BENOUARED.A qui nous à aidés à élaborer ce  
travail ainsi pour ses conseils précieux ses orientations.*

*Sans oublier les enseignants du Département  
d'Aéronautique, nous remerciment également les  
membres de jury d'avoir accepté d'examiner et juger ce  
travail de recherche.*

*Ainsi les étudiants de notre promotion 2010-2011.*

## ABREVIATION

<b>RII</b>	<b>R</b> éponse <b>I</b> mpulsionnel <b>I</b> nfini
<b>PSO</b>	<b>P</b> article <b>S</b> warm <b>O</b> ptimization
<b>OEP</b>	<b>O</b> ptimisation par <b>E</b> ssaims <b>P</b> articulaire
<b>PC</b>	<b>P</b> ersonnel <b>C</b> omputer
<b>DSP</b>	<b>D</b> igital <b>S</b> ignal <b>P</b> rocessor
<b>G best</b>	<b>G</b> lobal <b>b</b> est
<b>L best</b>	<b>L</b> ocal <b>b</b> est
<b>P best</b>	<b>P</b> osition <b>b</b> est
<b>F best</b>	<b>F</b> itness <b>b</b> est
<b>AG</b>	<b>A</b> lgorithme <b>G</b> énétique
<b>GPS</b>	<b>G</b> lobal <b>P</b> osition <b>S</b> ystem

:

خوارزمية التحسين بحشد من الوحدات هي خوارزمية مقتبسة من التصرف الاجتماعي لحيوانات تنشد حشد واحد, مثل سرب من الطيور المهاجرة أو سرب من السمك, الوحدات في نفس الحشد تتصل فيما بينها بكيفية مباشرة طويلة فترة البحث من أجل إيجاد حل للمشكلة المطروحة و ذلك بالارتكاز على خبرتهم الجماعية .  
الهدف من عملنا هذا هو استعمال طريقة التحسين هذه من أجل تصميم مرشحات رقمية بحيث نجد معادل تحويل مشابهة لمعادلة المرشح النموذجي معاملات مع المحافظة على استقرار المرشح.

### **Résumé:**

L'algorithme d'OEP s'inspire du comportement social des animaux évoluant en essaim, tels que les oiseaux migrateurs ou les poissons. Les particules d'un même essaim communiquent de manière directe entre elles tout au long de la recherche pour construire une solution au problème posé, en s'appuyant sur leur expérience collective.

L'objectif de notre travail consiste à utiliser cette méthode d'optimisation afin de synthétiser les filtres numériques RII pour trouver des fonctions de transfert proche au celle de filtre idéal afin de conserver la stabilité de ces filtres.

### **Abstract:**

PSO algorithm is inspired from the social behavior of animals living in swarm, such as bird flocks or fish schools. The particles of the swarm use a direct way of communication in order to build a solution to the considered problem, based on their collective experience.

The objective of our work consists to use this method of optimization in order to synthesize the design filter for find the function of transfer such as the filter ideal for conserved the stability of this filter.

## SOMMAIRE

INTRODUCTION GENERALE.....	01
----------------------------	----

### CHAPITRE 1 : GENERALITES SUR LES FILTRES NUMERIQUES RII

1.1 Introduction .....	02
1.2 Numérisation d'un système à temps continue.....	03
1.2.1 Acquisition de la suite $x_n$ .....	03
a. Dans l'espace du temps .....	03
b. Dans l'espace des amplitudes.....	04
1.2.2 Restitution du signal $y(t)$ .....	04
1.3 Le filtrage numérique .....	05
1.3.1 Définition d'un filtre numérique.....	05
1.4 Filtres à réponse impulsionnelle infinie (RII) .....	07
1.4.1 Fonction de transfert .....	07
1.4.2 Structures de réalisation d'un filtre RII.....	08
1.4.2.1 Structure directe.....	08
1.4.2.2 Structure décomposée.....	09
1.5 Stabilité des filtres IIR.....	09
1.5.1 Cellule du premier ordre.....	09
1.5.2. Cellule du second ordre.....	10
1.5.2.1 Cas des pôles réels.....	11
1.5.2.2 Cas des pôles complexes.....	11
1.6 Cellule du second ordre générale .....	12
1.7 La synthèse des filtres numériques RII.....	13
1.7.1 Synthèse par la méthode des pôles et zéros.....	13
1.7.2 Synthèse par la transformation bilinéaire.....	13
🚩 Objectif.....	13

🚩 Principe.....	13
1.7.2.1 Approximation de Butter Worth (maximally flat).....	14
1.7.2.2 Approximation de Tchebychev.....	15
1.7.2.3 Filtre Elliptique.....	17
1.8 Le gabarit d'un filtre .....	18
1.9 Avantages et inconvénients des filtres numériques.....	19
❖ Avantages.....	19
❖ Inconvénients.....	19
1.10 Les domaines d'application les filtres numériques RII.....	19
1.11 Conclusion .....	20

## **CHAPITRE 2 : L'OPTIMISATION PAR ESSAIMS PARTICULAIRES**

2.1 Introduction.....	21
2.2 Processus d'optimisation.....	23
2.2.1 Variables de conception.....	24
2.2.2 Espace de recherche.....	24
2.2.3 Fonction objectif .....	24
2.2.4 Contraintes de conception .....	24
2.2.5 Méthodes d'optimisation .....	25
2.2.5.1 La méthode de l'algorithme génétique .....	26
2.2.5.2 la méthode du Recuit simulé .....	27
2.2.5.3 La méthode Tabou.....	28
2.2.5.4 la méthode de colonies de fourmi.....	30
2.2.5.5 la méthode d'optimisation par essaim particulaire.....	31
2.3 Optimisation par Essaim Particulaire (OEP) .....	32
2.3.1 Principe général de la méthode .....	33
2.3.2 Principales caractéristiques.....	33
2.3.3 Modélisation mathématique de la méthode.....	34
2.3.4 Paramètres importants qu'influencent sur l'OEP.....	35
🚩 Nombre de particules .....	36
🚩 Topologie de voisinage.....	36
🚩 Coefficients de confiance .....	37

🚦 Facteur d'inertie.....	37
🚦 Vitesse maximale et coefficient de constriction .....	38
2.3.5 Déroulement de l'algorithme .....	39
❖ Algorithme avec une topologie en étoile.....	40
2.4 Exemple simple de la méthode OEP.....	42
2.5 Exemple complexe de la méthode OEP.....	47
2.6 discussion.....	52
2.7 Conclusion.....	52

### **CHAPITRE 3 : SYNTHÈSE DU FILTRE NUMÉRIQUE RII PAR LA MÉTHODE D'OPTIMISATION « OEP »**

3.1 Introduction.....	53
3.2 La conception de filtre numérique IIR.....	53
3.2.1 Description du problème.....	53
3.2.2 L'application d'OEP aux filtres numérique IIR.....	54
3.2.3 Synthèse de filtre numérique.....	56
❖ Paramètres reliés au filtre .....	57
❖ Les paramètres reliés à l'algorithme OEP .....	57
3.3 RESULTATS ET INTERPRÉTATION.....	60
3.3.1 Simulation n°=1.....	60
3.3.2 Simulation n°=2 .....	69
1) Bande de fréquence large .....	69
2) La Bande de fréquence petite.....	76
3.4 Conclusion.....	77
CONCLUSION GÉNÉRALE.....	78
REFERENCES .....	79
REFERENCES .....	80

## LISTE DES FIGURES

### CHAPITRE 1 : SYNTHÈSE DE FILTRE NUMÉRIQUE RII

Figure 1.1 : Traitement des signaux numériques .....	02
Figure 1.2 : Traitement numérique du signal $x(t)$ .....	03
Figure 1.3 : Signal échantillonné.....	03
Figure 1.4 : Opération de quantification.....	04
Figure 1.5 : Chaîne d'acquisition.....	04
Figure 1.6 : Chaîne de restitution.....	05
Figure 1.7 : Exemple de structure d'un filtre récursif .....	07
Figure 1.8.a : Structure directe type I d'un filtre RII.....	08
Figure 1.8.b : Structure direct type II d'un filtre RII .....	08
Figure 1.8.c : Structure parallèle.....	09
Figure 1.8.d : Structure cascade .....	09
Figure 1.9 : Structure d'une cellule de premier ordre .....	09
Figure 1.10 : Structure d'une cellule de second ordre .....	10
Figure 1.11 : Le triangle de stabilité .....	12
Figure 1.12 : Structure d'une cellule du second ordre.....	12
Figure 1.13 : La transformation bilinéaire .....	13
Figure 1.14 : Réponse d'amplitude du filtre Butter Worth.....	15
Figure 1.15 : Réponse d'amplitude du filtre Tchebychev I .....	15
Figure 1.16 : réponse d'amplitude de filtre Tchebychev II.....	16
Figure 1.17 : Réponse d'amplitude du filtre Tchebychev II.....	16
Figure 1.18 : Réponse d'amplitude du filtre elliptique .....	17
Figure 1.19 : Réponse d'amplitude du filtre passe-bas elliptique.....	18
Figure 1.20 : Le gabarit d'un filtre.....	18

### CHAPITRE 2 : L'OPTIMISATION PAR ESSAIS PARTICULAIRES

Figure 2.1: Diagramme de Classification des méta- heuristique.....	23
Figure 2.2: Schéma du processus d'optimisation .....	23
Figure 2.3: Diagramme d'Algorithmes Génétiques .....	26
Figure 2.4: Diagramme d'Algorithmes du recuit simulé.....	28
Figure 2.5: Diagramme d'Algorithmes de recherche de Tabou.....	29
Figure 2.6: La méthode Colonie fourmi.....	30
Figure 2.7: Groupe de : (a) oiseux, (b) poissons.....	32
Figure 2.8: Schéma de principe du déplacement d'une particule.....	34
Figure 2.9: Les topologies du voisinage d'un essaim.....	37
Figure 2.10: Organigramme de l'algorithme de la méthode OEP.....	41
Figure 2.11: Cas d'un nombre de particules insuffisant.....	43
Figure 2.12 : Cas d'un facteur d'inertie très grand.....	44
Figure 2.13 : Cas d'un espace de recherche vaste.....	45
Figure 2.14: Cas où les paramètres de contrôle sont assez réglés.....	46
Figure 2.15: Cas de modification dans les paramètres de confiances.....	48



Figure 2.16: Cas d'un facteur d'inertie très grand.....	49
Figure 2.17: L'influence de l'espace de recherche.....	50
Figure 2.18: Cas où les paramètres de contrôle sont assez réglés.....	51

### **CHAPITRE 3 : SYNTHÈSE DU FILTRE NUMÉRIQUE RII PAR LA MÉTHODE D'OPTIMISATION « OEP »**

Figure 3.1 : identification du filtre IIR pour déterminer Les coefficients.....	54
Figure 3.2 : Les paramètres d'un filtre.....	55
Figure 3.3 : organigramme d'OEP modifié.....	58
Figure 3.4 : Gabarit de filtre passe-bas.....	59
Figure 3.5 : Gabarit de filtre passe-haut.....	59
Figure 3.6 : Gabarit de filtre passe-bande.....	59
Figure 3.7 : Gabarit de filtre top-bande.....	59
Figure 3.8 : Réponse du filtre optimisée.....	60
Figure 3.9 : Evolution de la fitness.....	60
Figure 3.10 : Répartition des pôles.....	60
Figure 3.11 : Réponse du filtre optimisée.....	61
Figure 3.12 : Evolution de la fitness.....	61
Figure 3.13 : Répartition des pôles.....	61
Figure 3.14 : Réponse du filtre optimisée.....	62
Figure 3.15 : Evolution de la fitness.....	62
Figure 3.16 : Répartition des pôles.....	62
Figure 3.17 : Réponse du filtre optimisée.....	63
Figure 3.18 : Evolution de la fitness.....	63
Figure 3.19 : Répartition des pôles.....	63
Figure 3.20 : Réponse du filtre optimisée.....	64
Figure 3.21 : Evolution de la fitness.....	64
Figure 3.22 : Répartition des pôles.....	65
Figure 3.23 : Réponse du filtre optimisée.....	66
Figure 3.24 : Evolution de la fitness.....	66
Figure 3.25 : Répartition des pôles.....	66
Figure 3.26 : Réponse du filtre optimisée.....	67
Figure 3.27 : Evolution de la fitness.....	67
Figure 3.28 : Répartition des pôles.....	67

Figure 3.29 : Réponse du filtre optimisée.....	68
Figure 3.30 : Evolution de la fitness.....	68
Figure 3.31 : Répartition des pôles.....	69
Figure 3.32 : Réponse du filtre optimisée.....	70
Figure 3.33 : Evolution de la fitness.....	70
Figure 3.34 : Répartition des pôles.....	71
Figure 3.35: Réponse du filtre optimisée.....	72
Figure 3.36 : Evolution de la fitness.....	72
Figure 3.37 : Répartition des pôles.....	72
Figure 3.38 : Réponse du filtre optimisée.....	73
Figure 3.39 : Evolution de la fitness.....	73
Figure 3.40 : Répartition des pôles.....	74
Figure 3.41 : Réponse du filtre optimisée.....	75
Figure 3.42 : Evolution de la fitness.....	75
Figure 3.43: Répartition des pôles.....	75

## INTRODUCTION GENERALE

Cette thèse part du constat simple que les problèmes d'optimisation occupent une place de plus en plus importante dans les processus décisionnels. Dans de multiples domaines, le besoin toujours croissant d'efficacité et de rentabilité a conduit les décideurs à ne plus se fier seulement à l'expérience des experts, mais aussi aux résultats numériques obtenus par la résolution des problèmes d'optimisation. Une telle émulation a donné naissance à de nouvelles méthodes de résolution de problèmes d'optimisation.

Comme la plupart de problèmes d'engineering, la conception de filtres numériques, nécessitent une multitude de critères et spécifications de conception qui sont parfois contradictoires, et pour trouver une conception optimale, la tâche n'est pas du tout facile. Les méthodes itératives convergent généralement vers des conceptions sous-optimales, et par conséquent il était nécessaire de faire appel aux méthodes d'optimisation de plus en plus évoluées.

Dans ces dernières années, une nouvelle variété d'algorithmes d'optimisation a été proposée basant sur le mécanisme d'une inspiration biologique, connue sous l'appellation: la méthode d'optimisation par essaim particulaire (OEP), dans notre thèse on va utiliser cette méthode pour faire la conception des filtres numériques à réponse impulsionnelle infinie RII et pour trouver les coefficients d'un filtre pour que ce filtre suive un gabarit idéal.

Dans le premier chapitre, on exposera d'une manière succincte, un certain nombre de rappels qui seront utilisés tout au long de ce travail. On présentera les caractéristiques de filtre numérique à réponse impulsionnelle infinie (RII) et les méthodes de réalisation de ces filtres.

Le deuxième chapitre sera consacré à l'étude de la méthode d'optimisation par essaim de particules (OEP). Nous décrivons les paramètres et les différentes étapes de l'algorithme avec un exemple démonstratif de l'influence du réglage de ces paramètres sur la précision des résultats obtenus.

Dans le dernier chapitre, nous allons exposer les résultats de synthèse du filtre RII par l'algorithme d'optimisation (OEP) pour améliorer les coefficients de ce filtre.

Enfin, la conclusion récapitule les contributions de cette thèse.



# Généralités sur les filtres numériques RII

# CHAPITRE 1

## SYNTHESE DE FILTRE NUMERIQUE RII

### 1.1 Introduction

L'évolution spectaculaire de la technologie de circuits intégrés complexes explique certainement l'intérêt que suscitent les techniques numériques pour le traitement des signaux ; ces techniques s'introduites actuellement dans les domaines les plus divers, depuis l'analyse spectrale des signaux biologiques jusqu'aux dispositifs de filtrage et de modulation utilisés en télécommunication.

Nous venons de voir comment analyser le comportement d'un filtre numérique dont la fonction de transfert  $H(z)$  est connue. Sert à déterminer la réponse temporelle à une entrée fixée (impulsion, échelon) et la réponse fréquentielle au régime sinusoïdal permanent. Tout ceci suppose que le filtre soit déjà déterminé. Or, en général, le problème qui se pose est l'inverse, on désire déterminer la fonction de transfert  $H(z)$  d'un filtre qui doit avoir une réponse temporelle imposée ou une réponse fréquentielle entrant dans un gabarit précis. On dit alors que l'on fait la synthèse du filtre numérique RII.

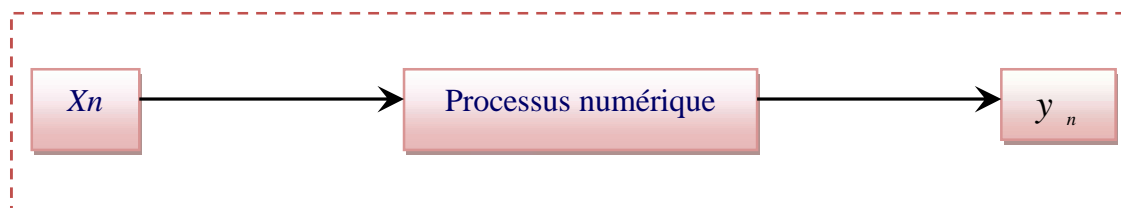
En réalité la plupart des signaux que l'on doit traiter et analyser tels que la parole, les signaux biologiques, sismiques, radars, audio ou vidéo sont analogiques par nature.

C'est-à-dire qu'ils sont fonction d'une variable continue (le temps), et qu'eux-mêmes varient de manière continue.

Ces signaux peuvent être traités de deux façons:

- ✚ Par un système de traitement analogique qui travaille directement sur le signal à temps continu.
- ✚ Par un système de traitement numérique qui travaille sur des nombres.

Dans le second cas, le système transforme une suite de données numériques  $[x_n]$  en une suite  $[y_n]$  selon un algorithme déterminé.



**Figure 1.1 :** Traitement des signaux numériques.

## 1.2 Numérisation d'un système à temps continue

Pour utiliser un tel système avec un signal analogique, il faut résoudre deux problèmes celui de l'acquisition de la suite  $x_n$  (représentation numérique du signal analogique  $x(t)$ ) celui de restitution du signal analogique de sortie  $y(t)$  à partir de la suite  $y_n$  [1].

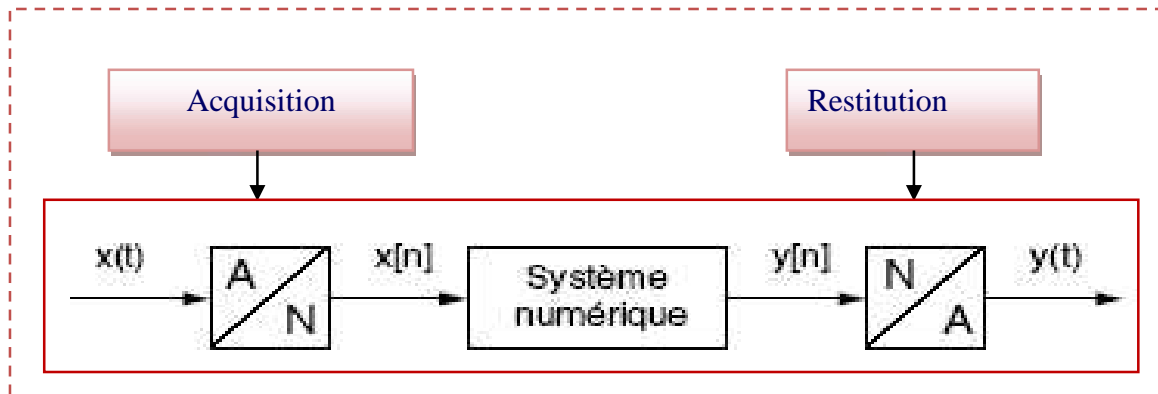


Figure 1.2 : Traitement numérique du signal  $x(t)$ .

### 1.2.1 Acquisition de la suite $x_n$

Le signal analogique d'entrée  $x(t)$  doit être transformé en une suite  $x_n$  équivalente, cette transformation nécessite une double approximation :

#### a. Dans l'espace du temps :

Le signal d'entrée ne sera défini que pour des valeurs discrètes du temps, ces valeurs étant des multiples entiers de durée  $T_e$  supérieure au temps d'exécution du processeur. (La relation 1.1)

$$x(t) \rightarrow x_e(t) = [x(nT_e)] \quad n = 0, 1, \dots \quad (1.1)$$

Ce découpage temporel constitue une discrétisation du signal dans l'espace du temps : c'est l'opération d'échantillonnage.

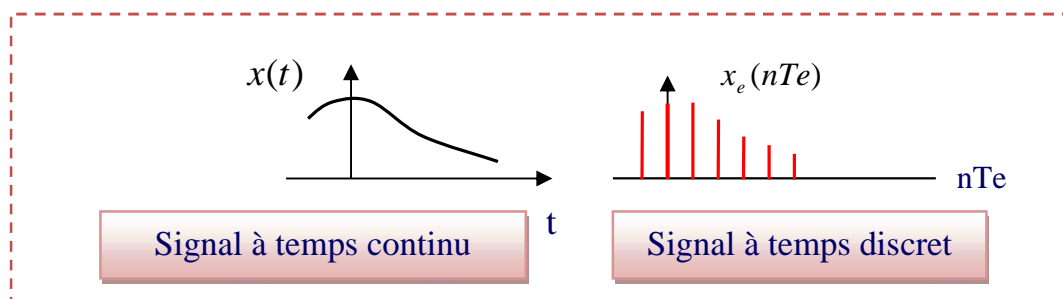


Figure 1.3 : Signal échantillonné.

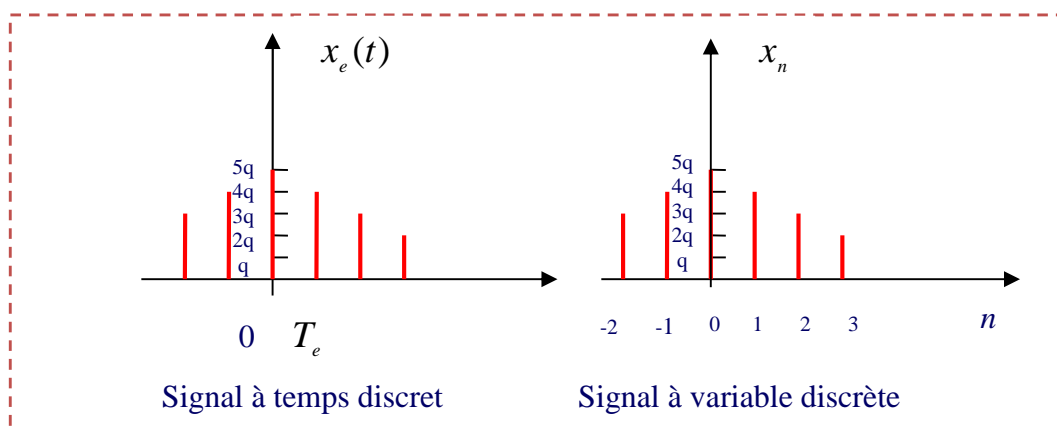
Le signal obtenu  $x_e(t)$  est appelé signal échantillonné ou à temps discret.  $T_e$  est la période d'échantillonnage.

**b. Dans l'espace des amplitudes :**

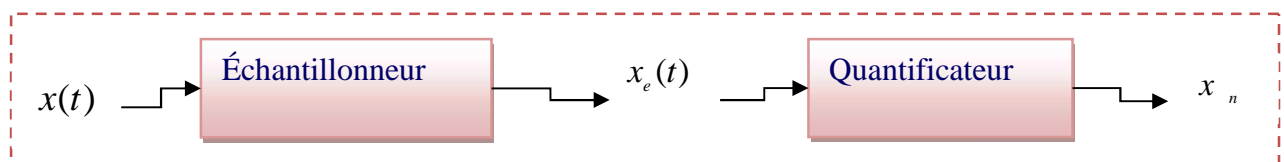
L'amplitude  $x(nT_e)$  d'un échantillon ne peut être représentée que par un nombre fini de valeurs discrètes. Si  $q$  est la valeur que représente le chiffre de plus faible poids, les valeurs  $x(nT_e)$  des différents échantillons ne peuvent être approchées que par un multiple entier de la quantité élémentaire  $q$ . (relation 1.2).

$$x(nT_e) \rightarrow x_n \cdot q \tag{1.2}$$

Cette discrétisation dans l'espace des amplitudes est l'opération de *quantification*.



**Figure 1.4 :** Opération de quantification.



**Figure 1.5 :** Chaîne d'acquisition.

**1.2.2 Restitution du signal  $y(t)$**

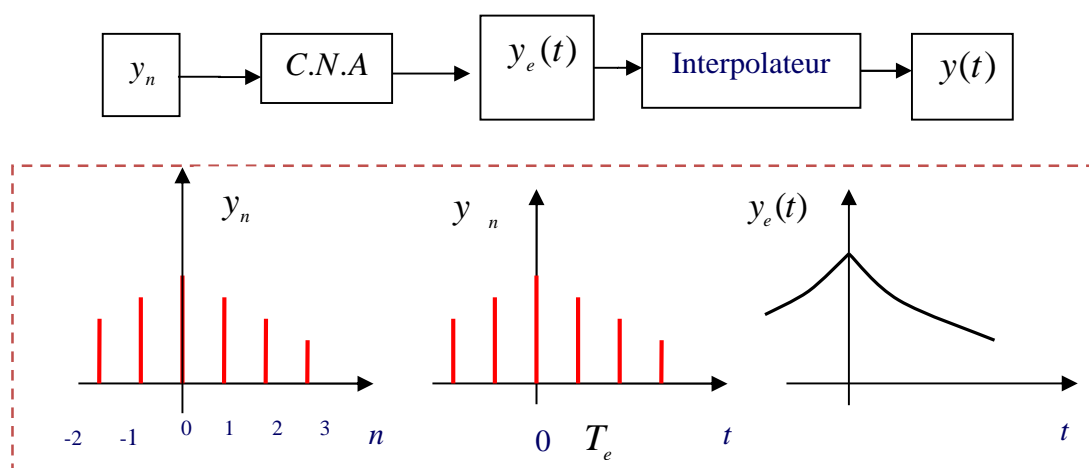
La suite  $y_n$  obtenue après traitement numérique des données d'entrée doivent être transformée en un signal analogique de sortie  $y(t)$ .

Les valeurs de sortie  $y_n$  sont fournies d'instant en instants multiples entiers de la durée  $T_e$ . ces valeurs sont transformées en une grandeur analogique par un convertisseur numérique analogique (C.N.A).

Le signal de sortie de C.N.A n'est donc défini que pour de valeurs discrètes de temps : c'est un signal à temps discret  $y_e(t)$ . Le signal  $y(t)$  n'est connu qu'aux instants  $nT_e$  et peut prendre n'importe quelle valeur entre deux instants de discrétisation.

Pour restituer le signal analogique, il faut alors choisir une procédure d'interpolation réalisée par un interpolateur.

La figure suivante illustre une chaîne de restitution.



**Figure 1.6 :** Chaîne de restitution.

### 1.3 Le filtrage numérique

#### 1.3.1 Définition d'un filtre numérique

En électronique, un filtre numérique est un élément qui effectue un filtrage à l'aide d'une succession d'opérations mathématiques sur un signal discret. C'est-à-dire qu'il modifie le contenu spectral du signal d'entrée en atténuant ou éliminant certaines composantes spectrales non-désirées [1]. Contrairement aux filtres analogiques, qui sont réalisés à l'aide d'un agencement de composants physiques (résistance, condensateur, inductance, transistor, etc.), les filtres numériques quant à eux, ils sont réalisés soit par programmation dans des circuits dédiés (processeurs de traitement du signal DSP,  $\mu$  Contrôleur, PC), ou sur des architectures programmables.

Les filtres numériques peuvent, en théorie, réaliser la totalité des effets de filtrage pouvant être définis par des fonctions mathématiques ou des algorithmes. Les deux principales limitations des filtres numériques sont la vitesse et le coût. La vitesse du filtre numérique est limitée par la vitesse (l'horloge, le "clock" en anglais) du processeur. Pour ce qui est du coût, celui-ci dépend du type et des performances du processeur utilisé.

On appelle filtrage numérique de fréquences ou filtrage numérique linéaire toute combinaison linéaire des échantillons d'entrée et de sortie d'un système échantillonné (système numérique dans la quasi-totalité des cas).



Si les signaux d'entrée ont été échantillonnés à la fréquence  $f_e$  (soit un pas d'échantillonnage ou période d'échantillonnage  $T_e$ ), la relation linéaire la plus générale est: (la relation 1.3)

$$b_0 Y(kT_e) = \sum_{m=0}^{M-1} a_m X(kT_e - mT_e) + \sum_{q=1}^{Q-1} b_q Y(kT_e - qT_e) \quad (1.3)$$

Que l'on écrit le plus souvent : (la relation 1.4)

$$b_0 Y_k = \sum_{m=0}^{M-1} a_m X(k - m) + \sum_{q=1}^{Q-1} b_q Y(k - q) \quad (1.4)$$

Avec: X: Signal d'entrée, Y: Signal de sortie,  $b_0$ ,  $a_i$ ,  $b_i$ : coefficients du filtre.

D'une façon générale, pour simplifier, on prend  $b_0 = 1$ .

En passant au domaine fréquentiel, moyennant la transformée en Z, on déduit de l'équation (1.3), l'expression de la fonction de transfert, qui elle aussi permet de définir un filtre numérique. Ainsi, la fonction de transfert générale d'ordre Q d'un filtre numérique est la suivante: (la relation 1.5)

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{m=0}^{M-1} a_m z^{-m}}{1 - \sum_{q=1}^{Q-1} b_q z^{-q}} \quad (1.5)$$

Rappelons que la réponse en fréquence s'obtient en remplaçant  $z$  par **[1]** : (la relation 1.6)

$$\exp(\pi j f) = \cos \pi f_n + j \sin \pi f_n \quad (1.6)$$

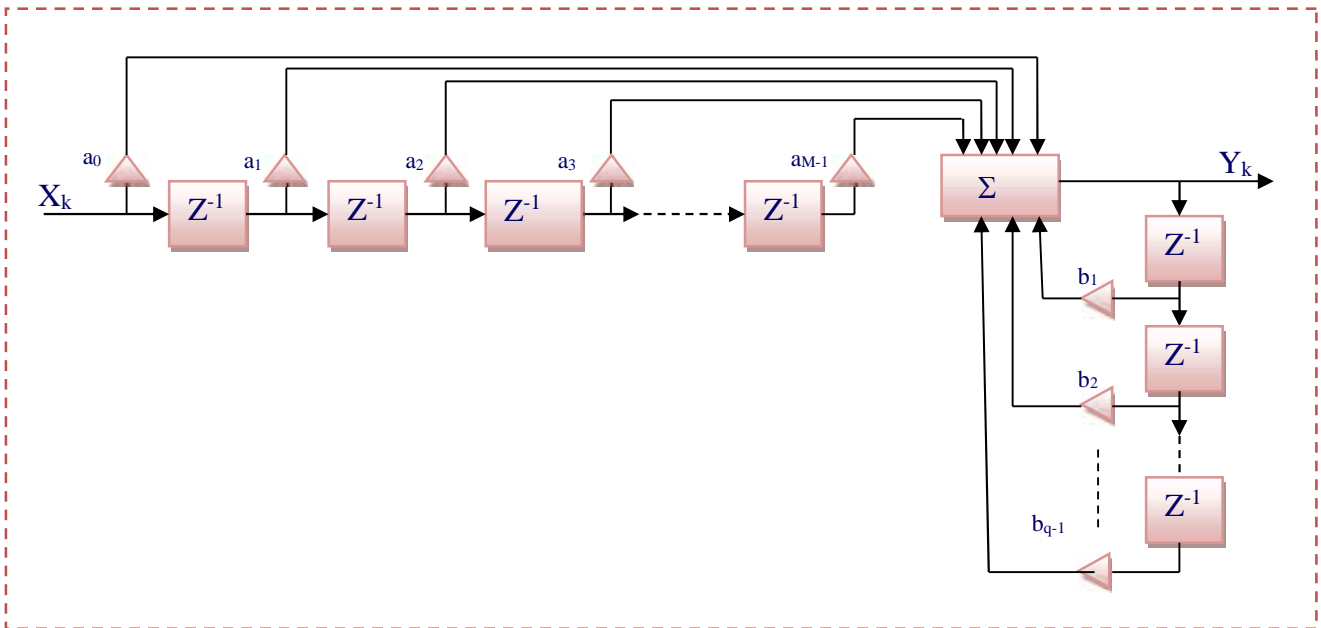
Avec  $f_n = 2(f/f_e)$ ,  $f_e$  : fréquence d'échantillonnage.

La relation (1.5) montre que la fonction de transfert d'un filtre numérique linéaire a obligatoirement la forme d'un rapport de deux polynômes algébriques en  $z$  **[3]**, les valeurs des coefficients  $a_k$  et  $b_k$  fixent le type du filtre (passe-bas, passe-haut, etc....).

#### 1.4 Filtres à réponse impulsionnelle infinie (RII)

Les filtres numériques à réponse impulsionnelle infinie sont des systèmes linéaires discrets invariants dans le temps dont le fonctionnement est régi par une équation de convolution portant sur une infinité de termes. En principe, ils conservent une trace des signaux qui leur ont été appliqués pendant une durée infinie **[3]**: ils sont à mémoire infinie.

Une telle mémoire est réalisée par une boucle de réaction de la sortie sur l'entrée, d'où la dénomination courante de *filtre récursif*.



**Figure 1.7** : Exemple de structure d'un filtre récursif.

#### 1.4.1 Fonction de transfert

Le filtre est spécifié par deux types de coefficients  $a_k, 1 \leq k < n$  et  $b_k, 0 \leq k < m$  :

La relation entrée-sortie d'un filtre récursif est donnée par (La relation 1.7) :

$$y(k) = \sum_{m=0}^{M-1} b_m x(k-m) - \sum_{n=1}^{N-1} a_n y(k-n) \quad (1.7)$$

Si on passe à la transformée en  $Z$  on a la relation suivante (1.8) :

$$H(z) = \frac{\sum_{m=0}^{M-1} b_m z^{-m}}{1 - \sum_{n=1}^{N-1} a_n z^{-n}} \quad (1.8)$$

Pour aborder l'étude des filtres RII, il est plus simple de considérer d'abord les cellules de filtres élémentaires du premier et du second ordre [1]. En fait, l'intérêt de ces structures simples va bien au-delà d'une introduction aux propriétés des filtres RII, car en pratique, les filtres RII même les plus complexes se présentent sous la forme d'une combinaison d'un ensemble de ces cellules élémentaires [3].

## 1.4.2 Structures de réalisation d'un filtre RII

### 1.4.2.1 Structure directe

On peut distinguer deux formes différentes [1]: (La relation 1.9)

$$H(z) = \frac{Y(z)}{R(z)} \cdot \frac{R(z)}{X(z)} = H_1(z) \cdot H_2(z) \quad (1.9)$$

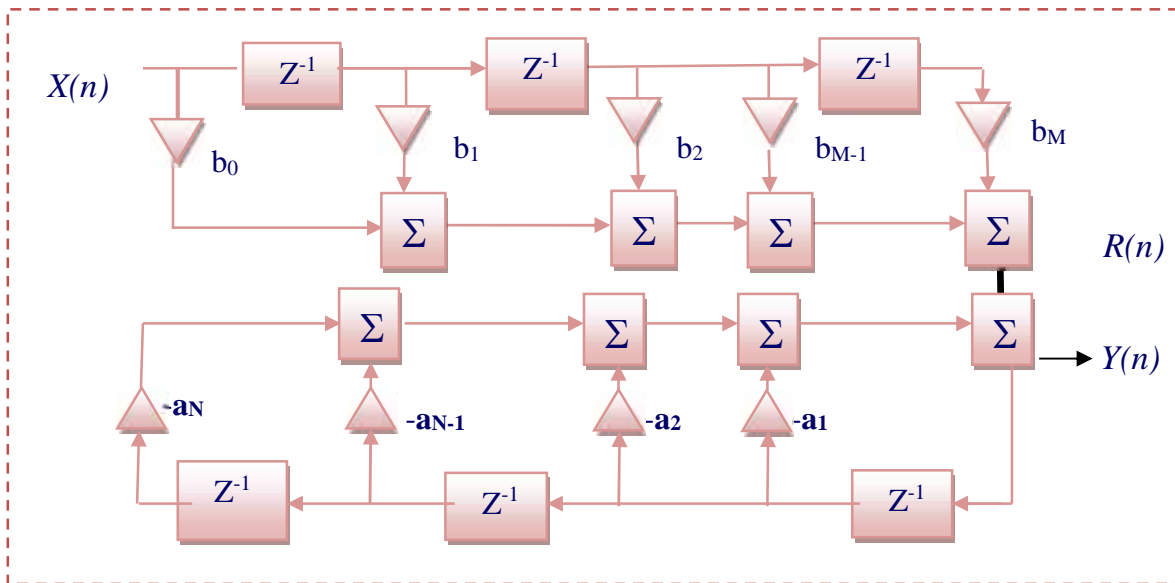


Figure 1.8.a : Structure directe type I d'un filtre RII.

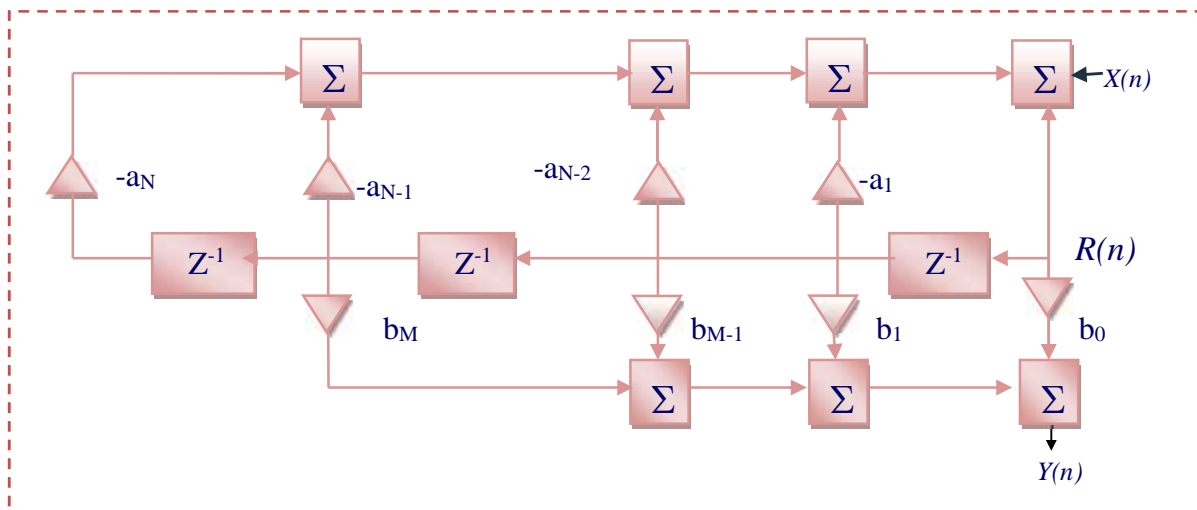
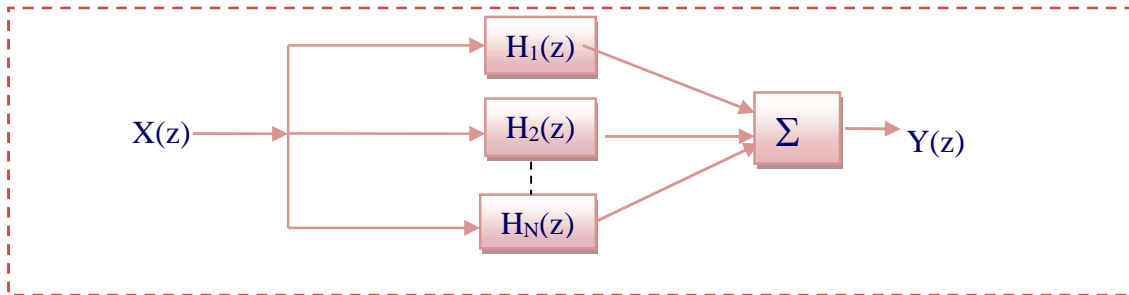


Figure 1.8.b: Structure direct type II d'un filtre RII.

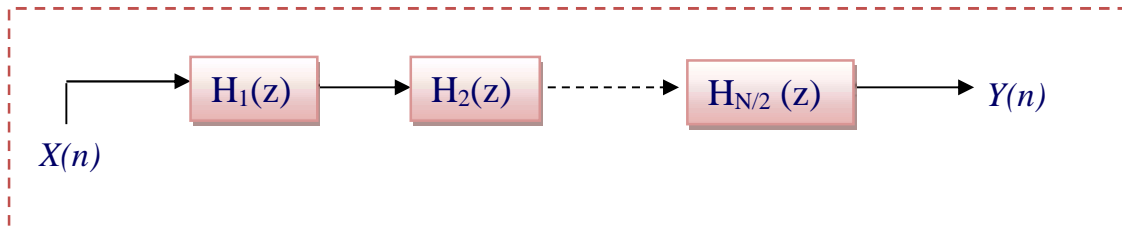
### 1.4.2.2 Structure décomposée

En pratique, la structure directe est peu utilisée, donc au lieu de réaliser  $H(z)$  directement, on effectue une décomposition en somme (structure parallèle) ou en produit (structure cascade) des fonctions élémentaires du premier ou de second ordre réalisées séparément [3]

On peut représenter les deux formes comme suite:



**Figure 1.8.c :** Structure parallèle.



**Figure 1.8.d :** Structure cascade.

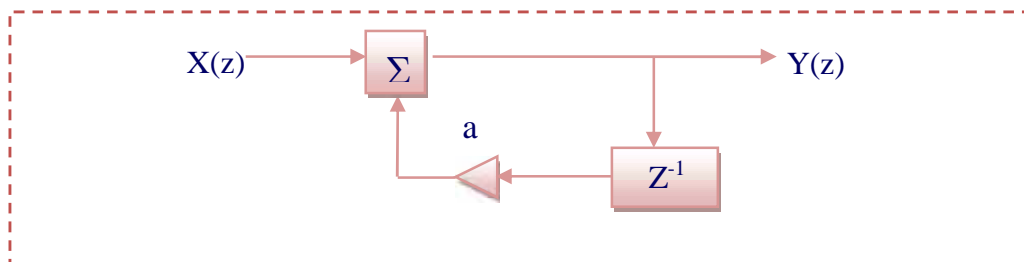
## 1.5 Stabilité des filtres IIR

### 1.5.1 Cellule du premier ordre

Sa réponse impulsionnelle est donnée par (la relation 1.10) de récurrence suivante :

$$y(n) - x(n) - a.y(n-1) \quad \text{Avec} \quad a \in \mathcal{J} \quad (1.10)$$

Le circuit comprend une mémoire de données et une mémoire de coefficients. Il faut pour chaque nombre de sortie effectuer une multiplication et une addition [1].



**Figure 1.9 :** Structure d'une cellule de premier ordre.

A partir de la relation de récurrence on tire : (la relation 1.11)

$$H(z) = \frac{y(z)}{x(z)} = \frac{1}{1 + a.z^{-1}} \quad (1.11)$$

Pour que la fonction  $H(z)$  soit stable, il faut que tous les pôles se trouvent à l'intérieur de cercle unité, c'est-à-dire  $|a| < 1$ .

Dans cette analyse, il résulte que la cellule du premier ordre offre des possibilités restreintes car elle ne possède qu'un seul pôle, qui doit être réel pour que le filtre soit à coefficient réels, et sa réponse en fréquence une fonction monotone [1].

Par contre, la cellule de second ordre offre des possibilités beaucoup plus variées.

C'est la structure la plus utilisée en filtrage numérique en raison de la modularité qu'elle apporte dans la réalisation des filtres même les plus complexes.

### 1.5.2 Cellule du second ordre

Soit un système qui à la suite de donnée  $x(n)$  fait correspondre la suite  $y(n)$  telle que : (la relation 1.12).

$$y(n) = x(n) - a_1 \cdot y(n-1) - a_2 \cdot y(n-2) \quad (1.12)$$

Cette structure correspond à un circuit à deux mémoires de données et deux mémoires de coefficients [1].

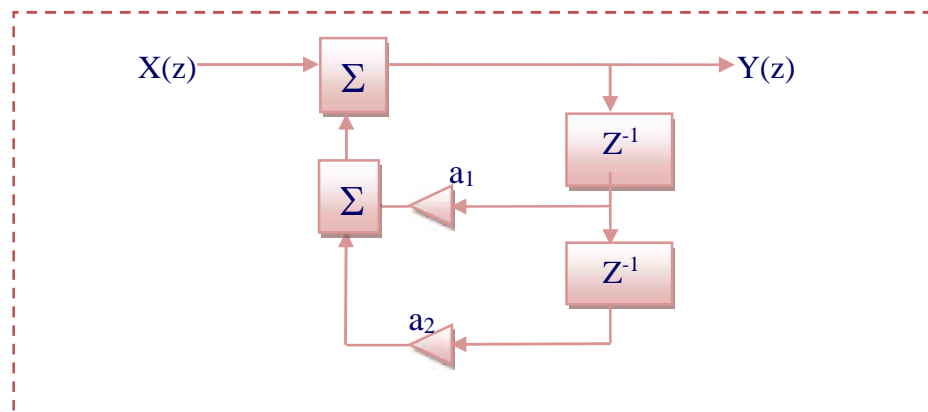


Figure 1.10 : Structure d'une cellule de second ordre.

Pour chaque échantillon de sortie, il faut deux additions et deux multiplications.

Pour faciliter l'écriture de la fonction de transfert en  $z$ ,  $H(z)$  du système, donnée par :

(La relation 1.13)

$$H(z) = \frac{1}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}} \quad (1.13)$$

Cette fonction possède, suivant le signe de  $a_1^2 - 4a_2$  des pôles réels ou complexes conjugués :

**Si  $a_1^2 > 4a_2$  :** On a deux pôles réels, leurs images, sont situés sur l'axe des réels.

La fonction de transfert est alors le produit de deux fonctions du premier ordre, ces fonctions sont identiques si  $a_1^2 = 4a_2$ .

Si  $a_1^2 < 4.a_2$  : Les deux pôles sont complexes conjugués. Leurs images sont  $p$  et  $p^*$  telle que : (la relation 1.14)

$$z_p = \frac{1}{2} \left( -a_1 \pm j\sqrt{4.a_2 - a_1^2} \right) \quad (1.14)$$

Pour que le filtre soit stable, il faut que le module de ses pôles soit inférieur à 1 [1].

### 1.5.2.1 Cas des pôles réels

(La relation 1.15)

$$\frac{1}{2} \left( -a_1 + \sqrt{a_1^2 - 4.a_2} \right) < 1 \quad \text{et} \quad \frac{1}{2} \left( -a_1 - \sqrt{a_1^2 - 4.a_2} \right) > -1 \quad (1.15)$$

D'où :

$$|a_1| < 1 + a_2 \quad (1.16)$$

### 1.5.2.2 Cas des pôles complexes

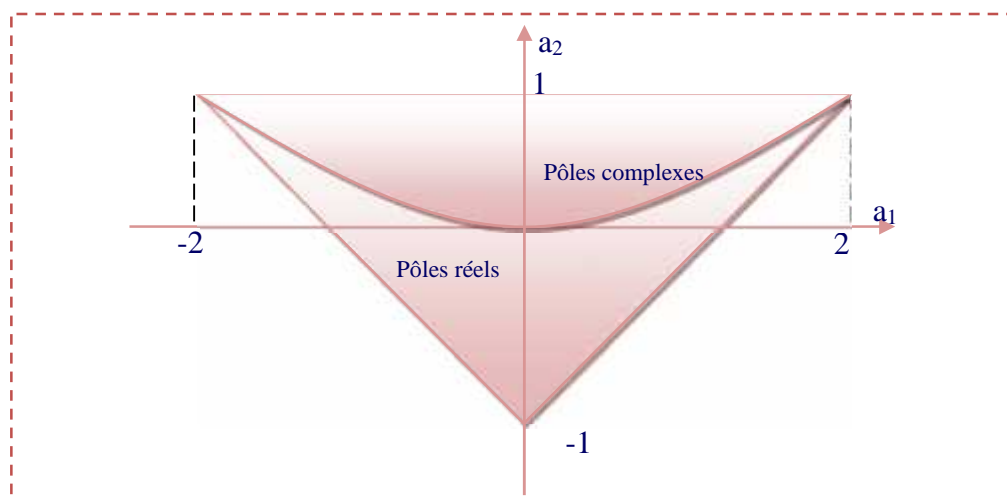
Pour qu'un filtre à pôles complexes soit stable il faut que :  $|a_2| < 1$  (1.17)

Le domaine de stabilité est donc un triangle délimité par trois droites d'équations :

$$a_2 = 1 \quad a_2 = -1 - a_1 \quad a_2 = a_1 - 1$$

De plus, le domaine des pôles réels est séparé de celui des pôles complexes par la parabole d'équation:

$$a_1^2 = 4.a_2. \quad (1.18)$$



**Figure 1.11** : Le triangle de stabilité montrant les valeurs des coefficients du filtre  $a_1$  et  $a_2$  pour lesquels le filtre est stable.

## 1.6 Cellule du second ordre générale

La cellule du second ordre la plus générale fait intervenir dans le calcul d'un élément de la suite de sortie  $y(n)$  à l'instant  $n$ , les données aux instants précédents  $x(n-1)$  et  $x(n-2)$ , on l'a définie par l'équation suivante : (la relation 1.19)

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) - a_1 y(n-1) - a_2 y(n-2) \quad (1.19)$$

Il en résulte la fonction de transfert en  $z$  suivante (la relation 1.20) :

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (1.20)$$

Cette expression correspond à la structure ci-dessous :

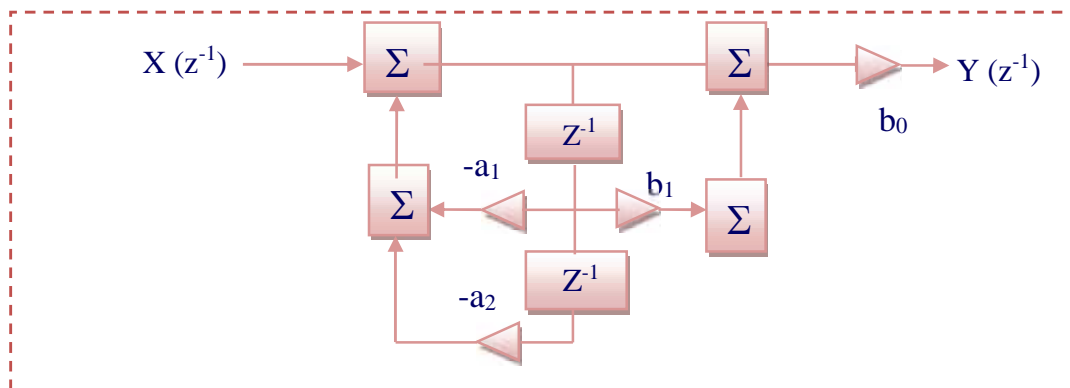


Figure 1.12 : Structure d'une cellule du second ordre.

## 1.7 La synthèse des filtres numériques RII

La synthèse d'un filtre est un ensemble de processus qui débute par la définition des caractéristiques du filtre, jusqu'à sa réalisation informatique et/ou électronique, en passant par la détermination de ses coefficients, Il existe de nombreuses méthodes permettant de synthétiser un filtre numérique récursif, parmi ces méthodes en effet des techniques discrètes itératives qui permettent de calculer directement les coefficients d'un filtre RII à partir d'un gabarit donné [4].

### 1.7.1 Synthèse par la méthode des pôles et zéros

Afin de faciliter l'étude des filtres récursifs, mais aussi pour améliorer la stabilité numérique de ceux ci, il est commode de représenter respectivement le numérateur, et le dénominateur des filtres récursifs comme étant le produit de filtres du premier ordre (un pôle ou un zéro) et du second ordre (paire de pôles, ou de zéros, complexes conjugués).

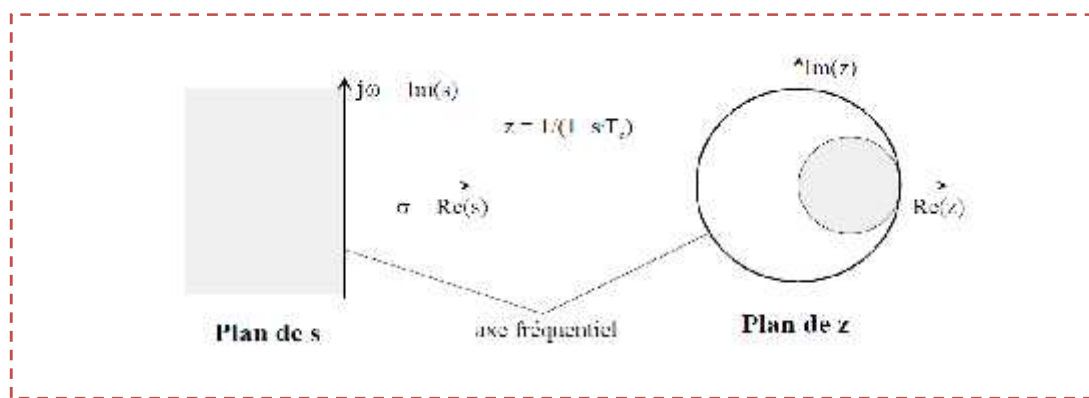
On factorisera donc la fonction de transfert de la forme suivante :

$$H(z) = \frac{K(z - z_1)(z - z_2) \dots (z - z_p)}{(z - p_1)(z - p_2) \dots (z - p_q)} \quad (1.21)$$

### 1.7.2 Synthèse par la transformation bilinéaire

**Objectif :** obtention de la fonction de transfert  $H(z)$  d'un filtre numérique qui a la même réponse fréquentielle qu'un filtre analogique de référence  $H(p)$ , autrement dit le même gabarit.

**Principe :** cette méthode a pour objectif de faire coïncider au mieux les domaines analogique et numérique. elle permet d'obtenir un filtre numérique qui possède approximativement la même réponse temporelle qu'un filtre analogique quelle que soit l'excitation utilisée.



**Figure 1.13 :** La transformation bilinéaire.

On en conclut la transformation suivante : (La relation 1.22)

$$s = \frac{1}{T_s} \cdot (1 - z^{-1})$$

$$z = (1 - s \cdot T_s)^{-1} \quad (1.22)$$

La transformation bilinéaire entraîne une relation non linéaire entre les fréquences  $f_a$  du domaine analogique et les fréquences  $f_d$  du domaine numérique

En pratique, cette transformation ne permet pas de conserver la de la fonction de transfert, il faut donc rechercher des méthodes d'approximation polynômiale d'un gabarit aboutissant aux filtres de Butter Worth, Tchebychev (I,II), elliptique [4].

#### 1.7.2.1 Approximation de Butter Worth (maximally flat)

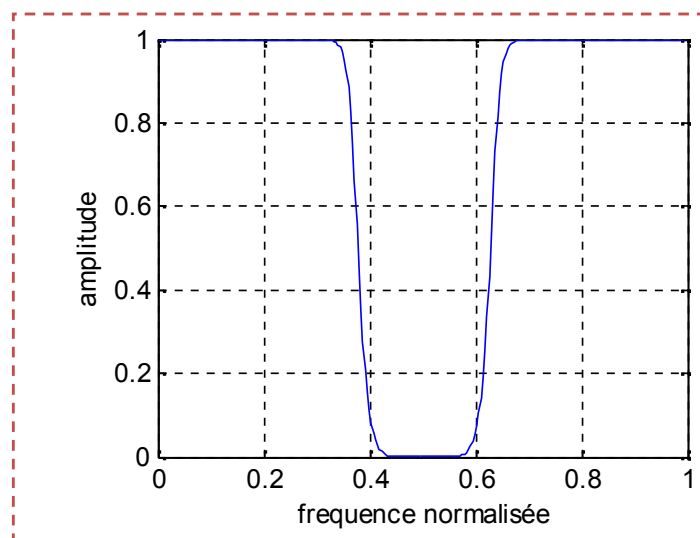
Le filtre Butter Worth a l'avantage d'une réponse en fréquence décroissante monotone continue permettant d'obtenir un gabarit qui ne possède pas des ondulations dans la bande passante et dans la bande atténuée, mais un tel modèle représente une large bande passante.



### ❖ Conception de filtre stop-bande de Butter Worth :

Pour ce filtre l'ordre est 8, et la réponse en amplitude représentée à la figure 1.14, agit comme un filtre entaille. Il peut être utilisé pour chasser une fréquence unique à laquelle l'atténuation est supérieure à 65dB. Puisque cette fréquence est  $\omega = 0.2$ , c'est 20% de la fréquence de Nyquist ou 10% de la fréquence d'échantillonnage. Cependant si la fréquence d'échantillonnage est choisie à 600Hz, on peut utiliser ce filtre pour filtrer l'indésirable bourdonnement à la fréquence de 60Hz du à l'alimentation du secteur dans les équipements radio [4].

Les coefficients du filtre numérique sont copiés ci-dessous à partir de la sortie du code MATLAB vu auparavant. Avec les spécifications:  $Wp1 = 0.18$ ,  $Wp2 = 0.24$ ,  $Ws1 = 0.2$ ,  $Ws2 = 0.205$ ,  $Rp = 0.5$ ,  $Rs = 50$ .



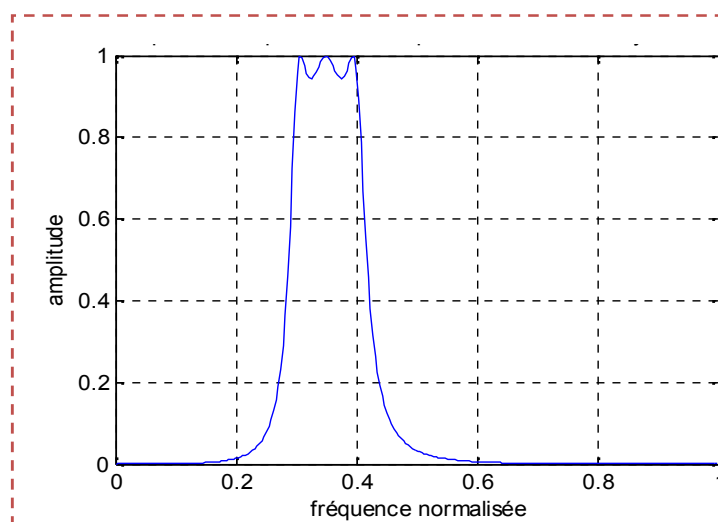
**Figure 1.14 :** Réponse d'amplitude du filtre stop-bande de Butter Worth.

#### 1.7.2.2 Approximation de Tchebychev

Le modèle de Tchebychev présente des filtres dont les caractéristiques de la réponse en fréquence ont une réponse d'amplitude équi-ondulations dans la bande passante, une réponse d'amplitude monotone décroissante dans la bande affaiblie et une pente de transition plus abrupte que celle des filtres de Butter Worth [3] [4].

### ❖ Conception de filtre passe-bande de Tchebychev I :

Avec les spécifications:  $W_{p1} = 0.3$ ,  $W_{p2} = 0.4$ ,  $W_{s1} = 0.25$ ,  $W_{s2} = 0.45$   $R_p = 0.5$ ,  $R_s = 50$ .



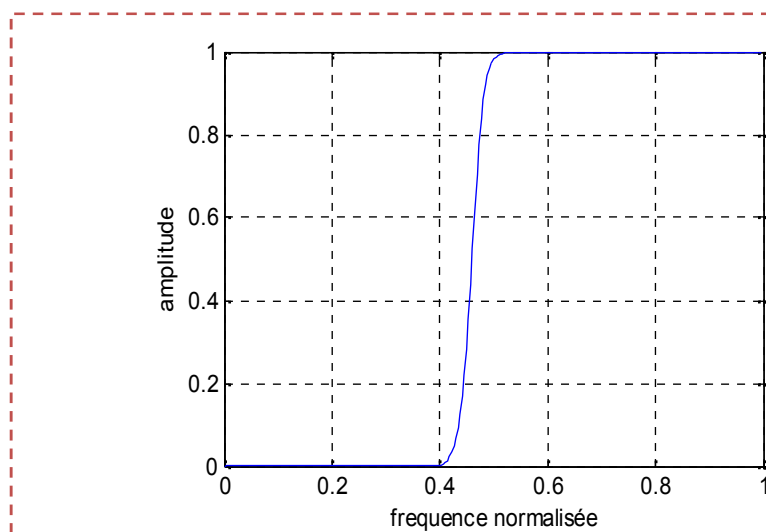
**Figure 1.15 :** Réponse d'amplitude du filtre passe-bande de Tchebychev I.

Dans le cas de filtre passe-bande de Tchebychev I la transition plus rapide, dont l'ordre trouvé est 7 .

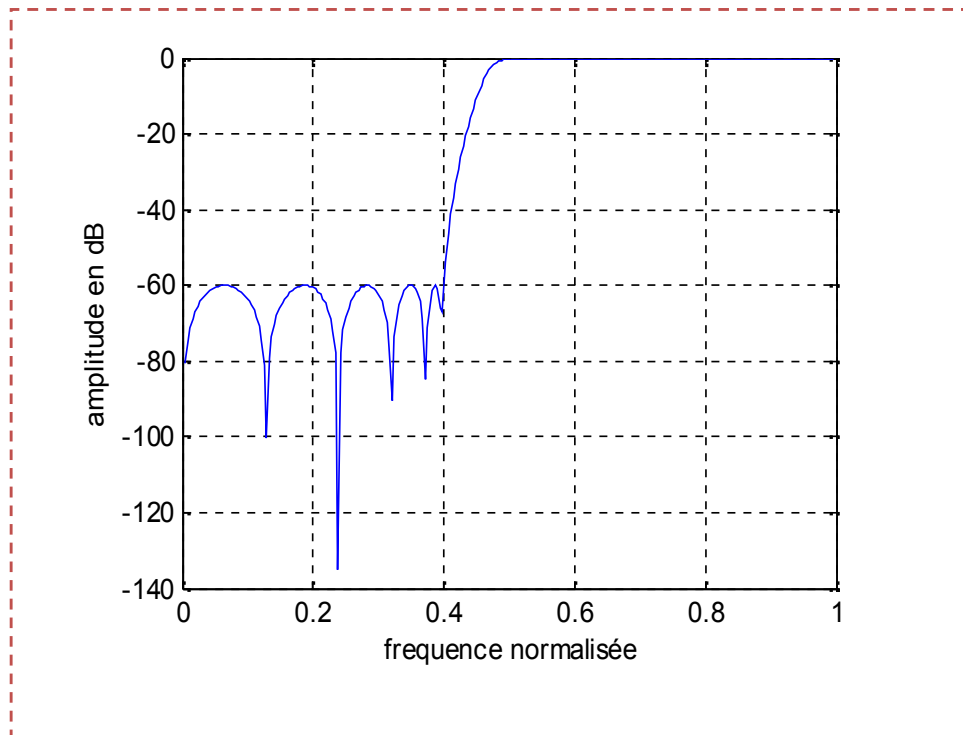
### ❖ Conception de filtre passe-haut de Tchebychev 2

Avec les spécifications:  $W_p = 0.5$ ,  $W_s = 0.4$ ,  $R_p = 0.5$ ,  $R_s = 60$ .

L'ordre de ce filtre est 11.



**Figure 1.16 :** réponse d'amplitude du filtre passe-haut de Tchebychev II.



**Figure 1.17 :** Réponse d'amplitude du filtre passe- haut de Tchebychev II.

La réponse en amplitude de ce filtre dont l'ordre à trouvé égal à 11, est montré dans les figures (1.16 et 1.17).

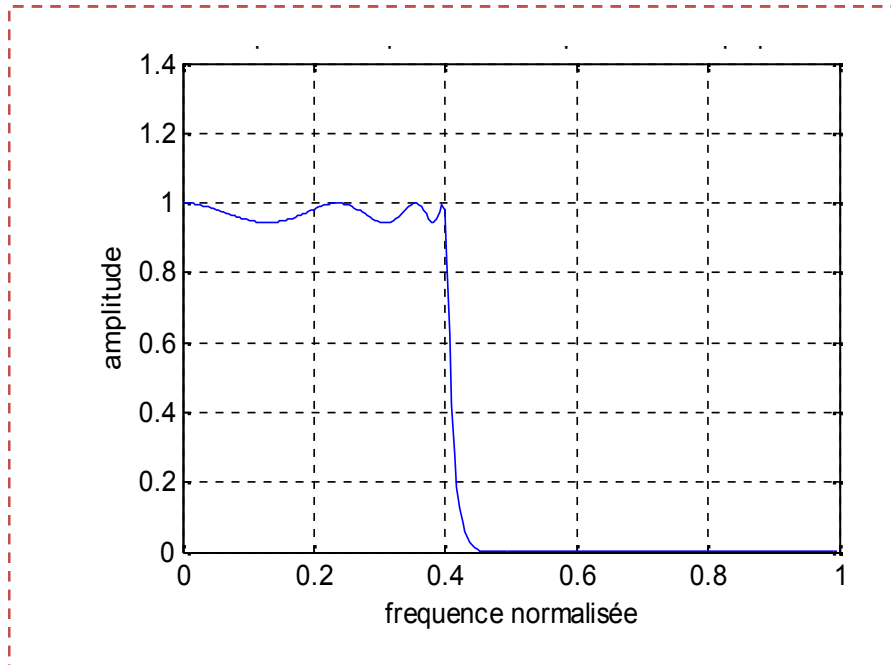
### 1.7.2.3 Filtre Elliptique

Les filtres elliptiques minimisent l'erreur maximum en la distribuant sur la bande passante et sur le stop bande.

Les équi-ondulations dans ces bandes passantes caractérisent la réponse d'amplitude des filtres elliptiques. Si on la compare à la conception des filtres de Butter Worth ou de Tchebychev de même ordre, la conception elliptique fournit la transition la plus abrupte entre la bande passante et le stop bande. C'est la raison pour laquelle les filtres elliptiques sont couramment utilisés [4].

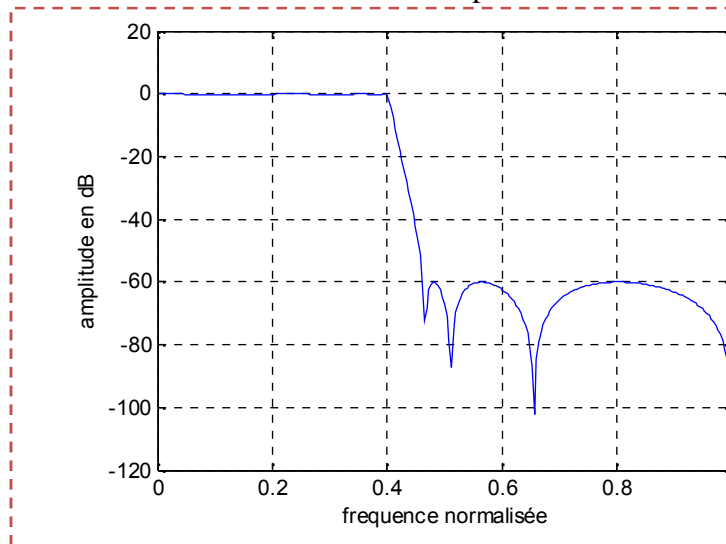
❖ **Conception de filtre passe-bas elliptique:**

Avec les spécifications:  $W_p = 0.4$ ,  $W_s = 0.5$ ,  $R_p = 0.5$ ,  $R_s = 60$ .



**Figure 1.18 :** Réponse d'amplitude du filtre passe-bas elliptique.

Cette figure montre une ondulation dans la bande passante.



**Figure 1.19 :** Réponse d'amplitude du filtre passe-bas elliptique.

La figure 1.19 montre l'amplitude de l'ondulation (en dB) dans la bande coupée, et l'atténuation minimale est de 60dB. L'ordre minimal de ce filtre trouvé est 7.

## 1.8 Le gabarit d'un filtre

Le gabarit d'un filtre numérique peut être caractérisé par :

✓ Les fréquences caractéristiques définies par rapport à bandes passantes et atténuées: par exemple : dernière fréquence passante et première fréquence atténuée pour un filtre passe bas [1].

✓ Les erreurs tolérées par rapport à la réponse en amplitude idéale. elles sont désignées par ondulations et définies pour chacune des bandes de fréquences.

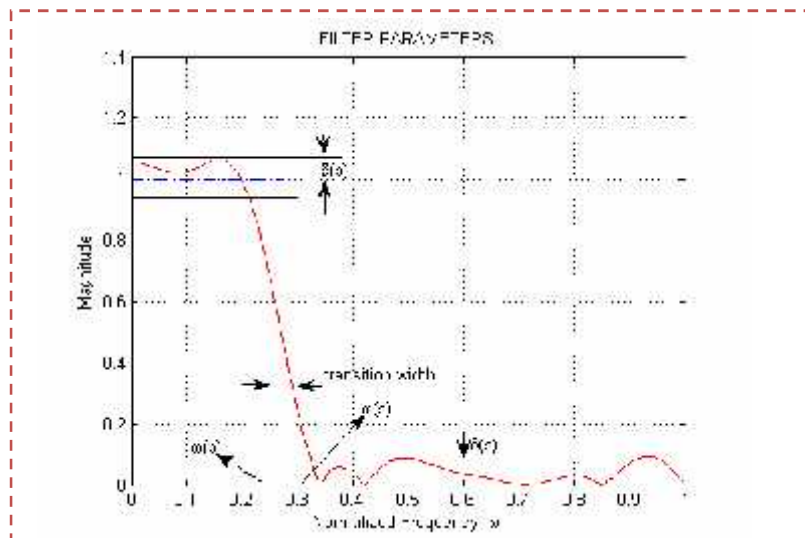


Figure 1.20 : Le gabarit d'un filtre.

On spécifie les caractéristiques d'un filtre avec un gabarit en donnant les paramètres :

p : L'ondulation en bande passante.

a : L'ondulation en bande atténuée.

$\omega_p$  : Dernière fréquence passante.

$\omega_a$  : Première fréquence atténuée.

## 1.9 Avantages et inconvénients des filtres numériques

Les filtres numériques présentent par rapport aux filtres analogiques les avantages et les inconvénients suivants:

### ❖ Avantages :

- ✓ une bande de transition étroite pour un ordre équivalent à celui d'un filtre non récursif;
- ✓ Les filtres numériques sont insensibles aux conditions extérieures (chaleur, humidité, etc....);
- ✓ Les filtres numériques ne sont pas sensibles aux non- idéalités d'un amplificateur opérationnel ;
- ✓ leur faible coût en calcul.

### ❖ Inconvénients :

- ✓ filtres numériques nécessitent un filtrage analogique anti-repliement à l'échantillonnage et à la restitution ;
- ✓ Les performances d'un filtre sont directement proportionnelles à la puissance de l'unité de calcul (processeur ou DSP) ;
- ✓ leur non-linéarité en phase ;
- ✓ leur instabilité numérique.

### 1.10 Les domaines d'application des filtres numériques RII

Le champ d'applications du filtrage numérique est très vaste puisqu'il va :

- des télécommunications au traitement de la parole ;
- les systèmes d'asservissement ;
- les radars et sonars ;
- la prospection sismique ;
- Modems (réseau câblés) ;
- Contrôleur de disques durs ;
- Cartes sons ; télévisions ;
- Récepteur GPS ;
- Traitement d'image (satellite, médicale...).

Dans de nombreuses applications sont utilisés des DSP dédiés c'est -à-dire qu'ils sont directement programmés avec un logiciel particulier à la conception (soit en ROM, soit câblé).

### 1.11 Conclusion

En effet, dans ce chapitre, il est possible de faire la synthèse des filtres numériques par l'approche récursive selon différentes méthodes de synthèse, et ce avec spécification des paramètres du filtre. Pour cela on utilise dans le chapitre suivant une méthode d'optimisation par essaims de particule OEP pour faire la synthèse du filtre numérique RII.



# CHAPITRE 2

## Optimisation par essaim de particule OEP

## CHAPITRE 2

### L'OPTIMISATION PAR ESSAIMS PARTICULAIRES (OEP)

#### 2.1 Introduction

L'optimisation est un sujet central en recherche opérationnelle, un grand nombre de problèmes d'aide à la décision pouvant en effet être décrits sous la forme de problèmes d'optimisation. Les problèmes d'identification, d'apprentissage supervisé de réseaux de neurones ou encore la recherche du plus court chemin sont, par exemple, des problèmes d'optimisation [14].

Un problème d'optimisation en général est défini par un espace de recherche  $s^* \in S$  et une fonction objectif  $f(s^*)$ . Le but est de trouver la solution  $s^* \in S$  de meilleure qualité  $f(s^*)$ . Suivant le problème posé, on cherche soit le minimum soit le maximum de la fonction  $f$  a plusieurs variables [6].

$$s^* = \min (f(s) / s \in S)$$

**Heuristique** est un algorithme qui fournit rapidement une solution réalisable, Une heuristique, ou méthode approximative, est donc le contraire d'un algorithme exact qui trouve une solution optimale pour un problème donné. Les algorithmes de résolution exacts étant de complexité exponentielle, il est généralement plus judicieux de faire appel à des méthodes heuristiques pour des problèmes difficiles. On retiendra cependant que des méthodes de résolution exactes sont de complexité exponentielle mais parfois plus efficaces en pratique qu'une méthode heuristique [5]. L'usage d'une heuristique est pertinent pour calculer une solution approchée d'un problème et ainsi accélérer le processus de résolution exacte. Généralement une heuristique est conçue pour un problème particulier, en s'appuyant sur sa structure propre, mais les approches peuvent contenir des principes plus généraux. On parle de **méta-heuristique** pour les méthodes approximatives générales, pouvant s'appliquer à différents problèmes [15] (comme le recuit simulé par exemple).

**Les méta-heuristiques** forment une famille d'algorithmes d'optimisation visant à résoudre approximativement des problèmes d'optimisation [17], pour lesquels on ne connaît pas de méthode classique plus efficace. Elles sont généralement utilisées comme des méthodes



génériques pouvant traiter une large gamme de problèmes différents, sans nécessiter de changements profonds dans l'algorithme employé [2].

Les méta-heuristiques sont assez facilement sujettes à extensions. Parmi celles-ci, on peut citer :

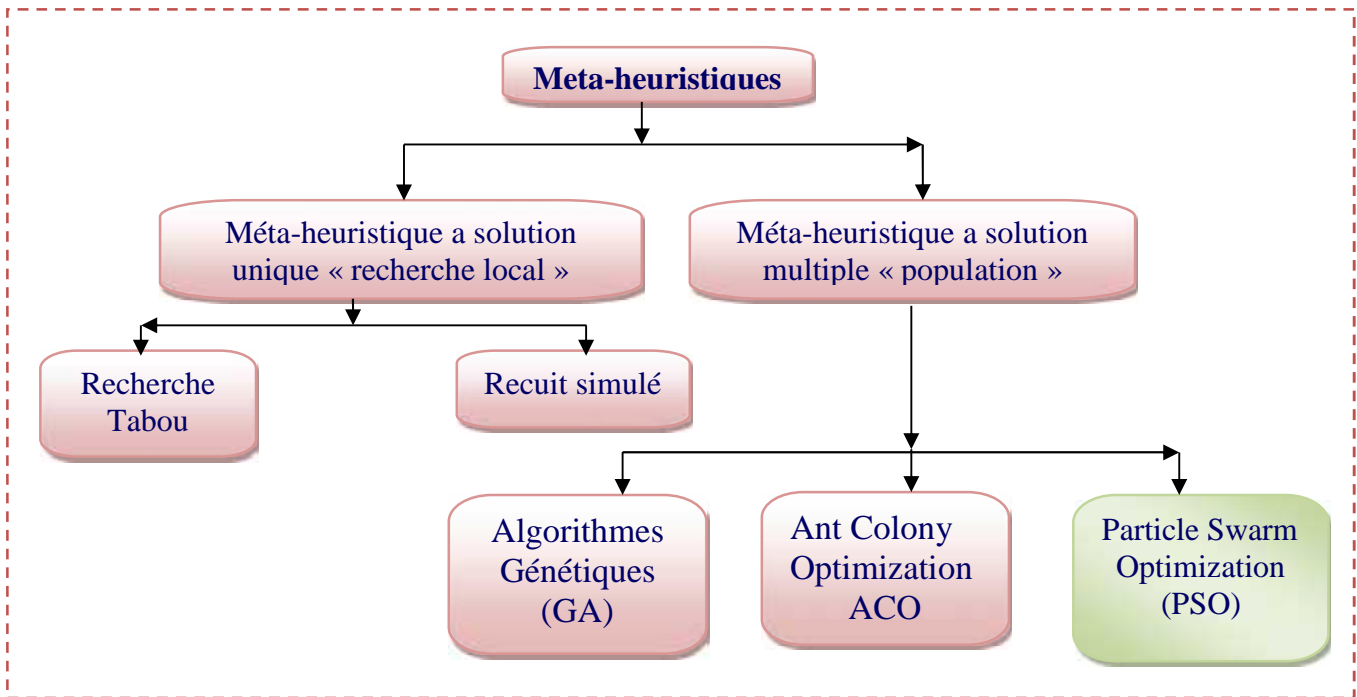
- Les méta-heuristiques pour l'optimisation multi-objective. Le but est ici non pas de trouver un optimum global mais de trouver un ensemble d'optima qui forment une surface de compromis pour les différents objectifs du problème.
- Les méta-heuristiques pour l'optimisation multimodale, où l'on ne cherche plus l'optimum global, mais l'ensemble des meilleurs optima locaux.
- Les méta-heuristiques pour l'optimisation dynamique, où il faut approcher l'optimum à chaque pas de temps, car la fonction objectif change de topologie au cours du temps.
- Les méta-heuristiques hybrides, qui combinent différentes méta-heuristiques, afin d'en tirer les avantages respectifs.
- Les méta-heuristiques parallèles, pour lesquelles on cherche à accélérer le calcul, en distribuant la charge de calcul sur plusieurs calculateurs.

L'utilisateur est, certes, demandeur de méthodes rapides et efficaces, mais il est aussi demandeur de méthodes simples d'utilisation. Un enjeu majeur des méta-heuristiques est donc de faciliter le choix des méthodes et de simplifier leurs réglages, afin de les adapter au mieux aux problèmes posés [15].

Le monde des méta-heuristiques est un monde en constante évolution. De nombreuses méthodes sont proposées chaque année pour tenter d'améliorer la résolution des problèmes les plus complexes. du fait de cette activité permanente, un grand nombre de classes de méta-heuristiques existe actuellement [10].

Donc le but d'une méta-heuristique est de résoudre un problème d'optimisation donné : elle cherche un objet mathématique (une permutation, un vecteur, etc.) minimisant (ou maximisant) une fonction objectif, qui décrit la qualité d'une solution au problème.

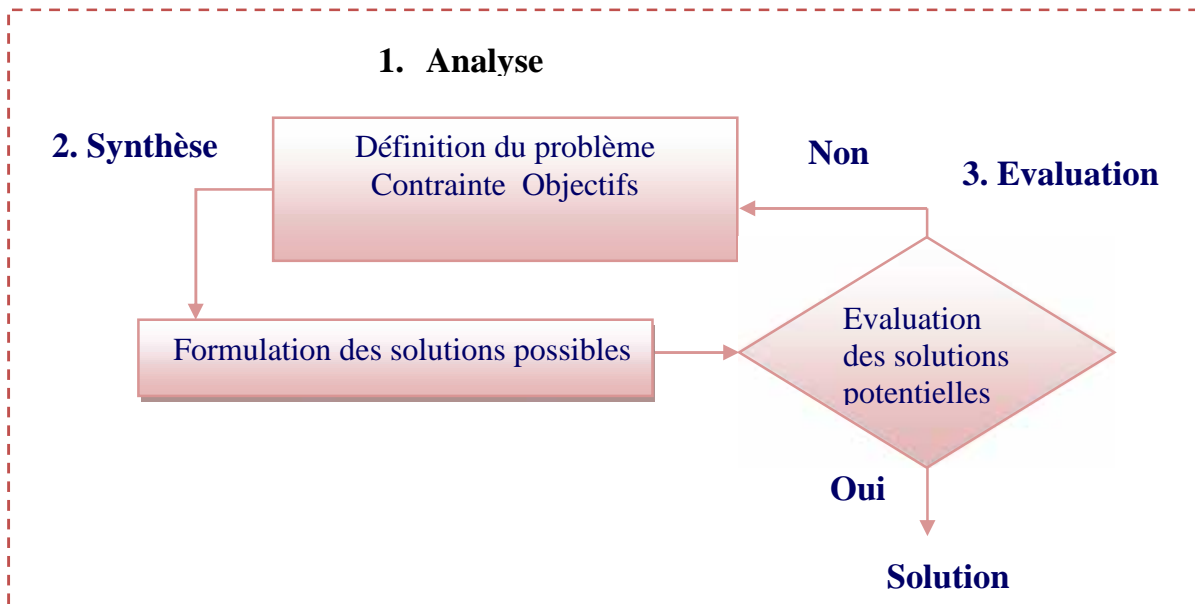
L'ensemble des solutions possibles forme l'espace de recherche. L'espace de recherche est au minimum borné [17].



**Figure 2.1:** Diagramme de Classification des méta-heuristiques.

## 2.2 PROCESSUS D'OPTIMISATION

La figure (2.2) présente les trois étapes du processus d'optimisation : analyse synthèse et évaluation. Tout d'abord, il convient d'analyser le problème et d'opérer un certain de choix préalables [2].



**Figure 2.2:** Schéma du processus d'optimisation.

- ❖ **Variables du problème** : quels sont les paramètres intéressants à faire varier ?
- ❖ **Espace de recherche** : dans quelles limites faire varier ces paramètres ?
- ❖ **Fonction objectif** : quels sont les objectifs à atteindre ?
- ❖ **Méthode d'optimisation** : quelle méthode choisir ?

Une fois effectués ces choix, la méthode choisie synthétise les solutions potentielles qui sont évaluées, puis éliminées jusqu'à obtention d'une solution acceptable.

### **2.2.1 Variables de conception**

Les variables de la fonction objectif sur lesquelles porte l'optimisation sont les variables de conception qui peuvent être de natures diverses. Par exemple, pour un réseau d'antennes, il peut s'agir de la position des antennes, de leurs courants d'alimentation, etc.

C'est à l'utilisateur de définir les variables du problème. Il peut avoir intérêt à faire varier un grand nombre de paramètres afin d'augmenter les degrés de liberté de l'algorithme.

### **2.2.2 Espace de recherche**

Dans certains algorithmes d'optimisation, tels que les stratégies d'évolution, l'espace de recherche est infini : seule la population initiale est confinée dans un espace fini, mais dans le cas des algorithmes de type OEP, il est généralement nécessaire de définir un espace de recherche fini.

### **2.2.3 Fonction objectif**

Cette fonction objectif ou d'évaluation (fitness) permet de mesurer l'efficacité de la solution. La pertinence des solutions potentielles dépend essentiellement de la formulation de cette fonction. En effet, quelle que soit sa définition, l'algorithme convergera vers un optimum de cette fonction. Elle doit donc exprimer le plus fidèlement possible sous forme mathématique le ou les objectifs à atteindre.

### **2.2.4 Contraintes de conception**

Les contraintes de conception peuvent être de différentes natures.

**Exemple :** Pour un réseau d'antennes, il s'agira :

- Du niveau maximum de lobes secondaires à ne pas dépasser.
- De l'ouverture du faisceau à ne pas dépasser.
- Du nombre maximal d'antennes formant le réseau.

### **2.2.5 Méthodes d'optimisation**

Une fois définie la fonction à optimiser, il s'agit de choisir une méthode adaptée au problème posé. Il existe de nombreuses méthodes déterministes (ou exactes) qui permettent de résoudre certains types de problèmes d'optimisation en un temps fini. Cependant, ces méthodes nécessitent que la fonction objectif présente un certain nombre de caractéristiques, telles que la convexité, la continuité ou encore la dérivabilité. Parmi ces méthodes les plus connues, on peut citer la méthode de Newton, la méthode du gradient [2].

Certains problèmes restent cependant trop complexes à résoudre pour les méthodes déterministes. Certaines caractéristiques peuvent être problématiques pour ces méthodes. Parmi celles-ci, on peut citer la discontinuité, la non-dérivabilité. Dans ce cas, le problème

d'optimisation est dit difficile, car aucune méthode déterministe ne peut résoudre ce problème en un temps raisonnables.

Les problèmes d'optimisation se divisent en deux catégories [7] :

Les problèmes à variables discrètes et les problèmes à variables continues. De façon générale, un problème d'optimisation à variables discrètes, ou combinatoire, consiste à trouver dans un ensemble discret, la meilleure solution réalisable. Le problème majeur réside ici dans le fait que le nombre de solutions réalisables est généralement très élevé, donc il est très difficile de trouver la meilleure solution dans un temps raisonnables.

Les problèmes à variables continues sont, eux, moins formalisés que les précédents. En effet, la grande majorité des méta-heuristiques existantes ont été créées pour résoudre des problèmes à variables discrètes. Cependant, la nécessité croissante des méthodes pouvant résoudre ce type de problèmes a poussé les chercheurs à adapter leurs méthodes au cas continu [2].

Par dispositif expérimental. Il est à noter qu'il existe des problèmes à variables mixtes, c'est-à-dire que le problème présente à la fois des variables discrètes et continues.

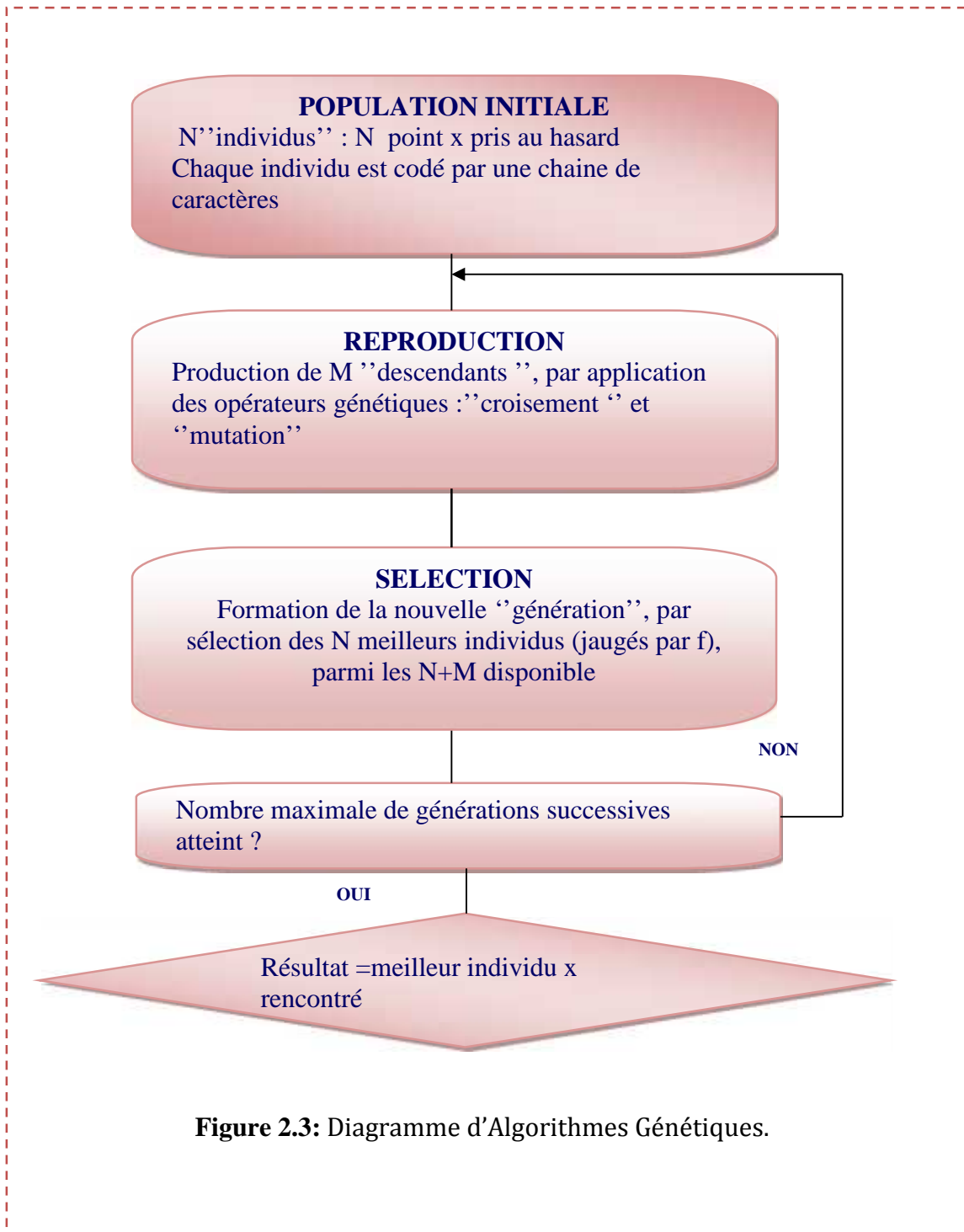
Les méta-heuristiques les plus courantes sont le recuit simulé, la recherche tabou, les algorithmes évolutionnaires, les algorithmes de colonies et fourmis ou encore l'optimisation par essaim particulaire.

Dans le paragraphe suivant on va généraliser quelques méthodes d'optimisations, après on s'intéressera à notre but de travail c'est l'optimisation par essaim particulaire.

#### **2.2.5.1 La méthode de l'algorithme génétique**

L'algorithme génétique est un algorithme d'optimisation s'appuyant sur des techniques dérivées de la génétique et de l'évolution naturelle : croisement, mutation, sélection, etc. L'algorithme génétique a déjà une histoire relativement ancienne puisque les premiers travaux de John Holland sur les systèmes adaptatifs remontent à 1962[11].

L'algorithme génétique recherche le ou les extrema d'une fonction définie sur un espace de données. Pour l'utiliser, on doit disposer des cinq éléments suivants John Holland [12].



**Figure 2.3:** Diagramme d'Algorithmes Génétiques.

### 2.2.5.2 La méthode du Recuit simulé

Le recuit simulé est une méthode d'optimisation apparue en 1982 est inspirée de la métallurgie, avec la publication dans la revue science d'un article de Kirpatrick. Elle permet de trouver des solutions, non optimales, mais souvent satisfaisantes, en un temps défini à l'avance. Elle se fonde sur une analogie entre les problèmes d'optimisation et ceux de la physique statistique. L'application principale du recuit simulé est la recherche opérationnelle c'est-à-dire tous les problèmes d'optimisation où l'on désire maximiser ou minimiser une quantité. Le recuit simulé a déjà obtenu d'excellents résultats dans divers domaines ce qui a incité les mathématiciens, les physiciens et les ingénieurs à s'intéresser à cette méthode et à la développer. Il peut être utilisé pour résoudre des problèmes continus en discrétisant l'espace de recherche [14].

L'intérêt du recuit simulé ou d'autres méthodes dérivées par rapport à une recherche locale, est dû au fait que ces algorithmes acceptent de façon probabiliste des configurations d'énergie plus élevées, ce qui permet de ne pas rester piégé par un minimum local.

#### Algorithme de recuit simulé

**Déterminer** une configuration aléatoire

**Choix** des mécanismes de perturbation d'une configuration

**Initialiser** la température T

**Tant que** la condition d'arrêt n'est pas atteinte faire

**Tant que** l'équilibre n'est pas atteint faire

**Tirer** une nouvelle configuration S'

**Appliquer** la règle de Metropolis

**Si**  $f(S0) < f(S)$

$S_{min} = S0$

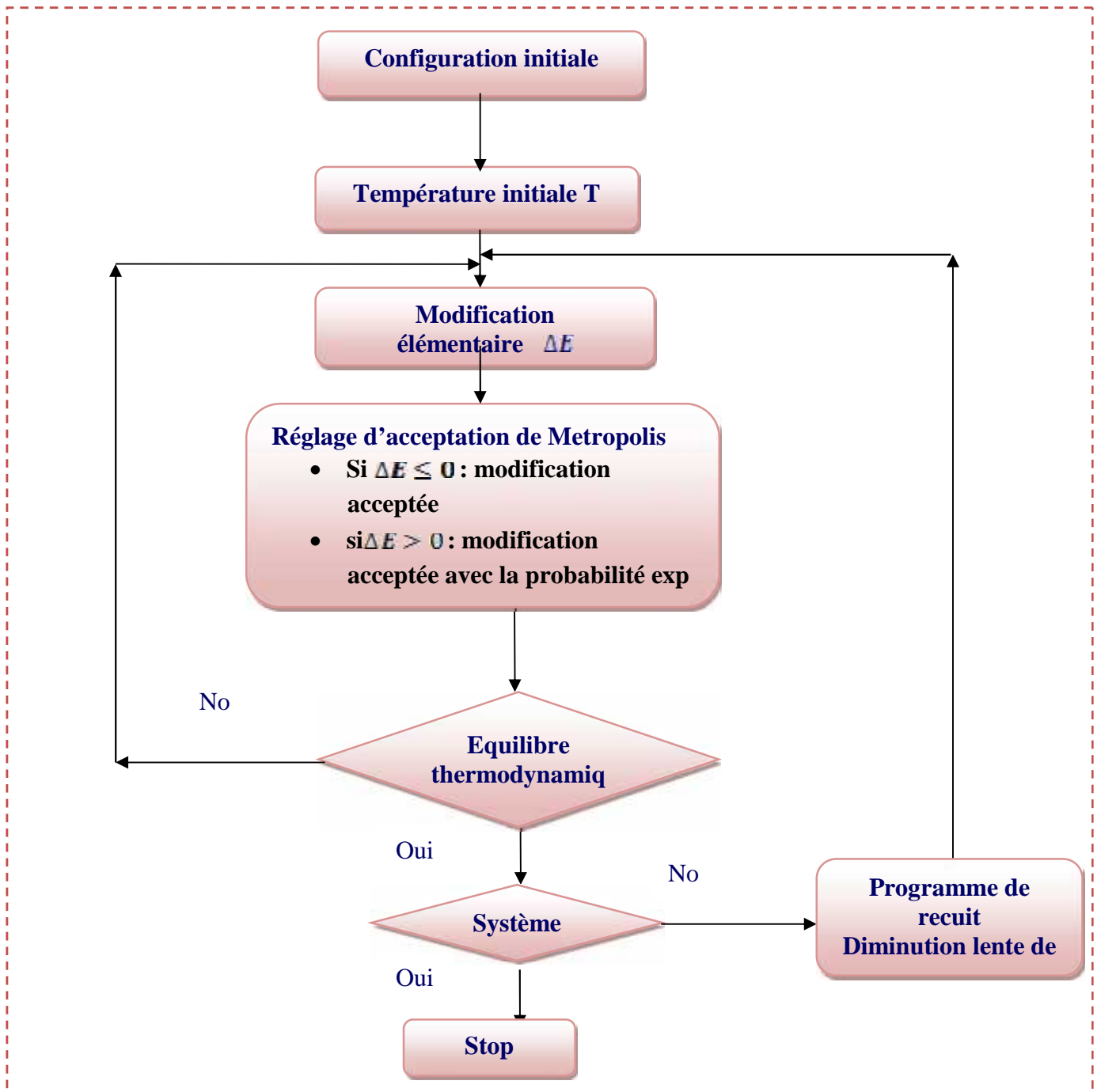
$f_{min} = f(S0)$

**Fin Si**

**Fin Tant que**

**Décroître** la température

**Fin Tant que.**



**Figure 2.4:** Diagramme d'Algorithmes du recuit simulé.

### 2.2.5.3 LA METHODE TABOU

La méthode Tabou est une technique de recherche dont les principes ont été proposés pour la première fois par Fred Glover dans les années 80, et elle est devenue très classique en optimisation combinatoire [16]. Elle se distingue des méthodes de recherche locale simples par le recours à un historique des solutions visitées, de façon à rendre la recherche un peu moins « aveugle ». Il devient donc possible de s'extraire d'un minimum local, mais, pour éviter

d'y retomber périodiquement, certaines solutions sont bannies, elles sont rendues « taboues » [11].

### 🚩 Algorithme de Recherche Tabou

**Déterminer** une configuration aléatoire  $s$

**Initialiser** une liste tabou vide

**Tant que** le critère d'arrêt n'est pas atteint **faire**

**Perturbation** de  $s$  suivant  $N$  mouvements non tabous

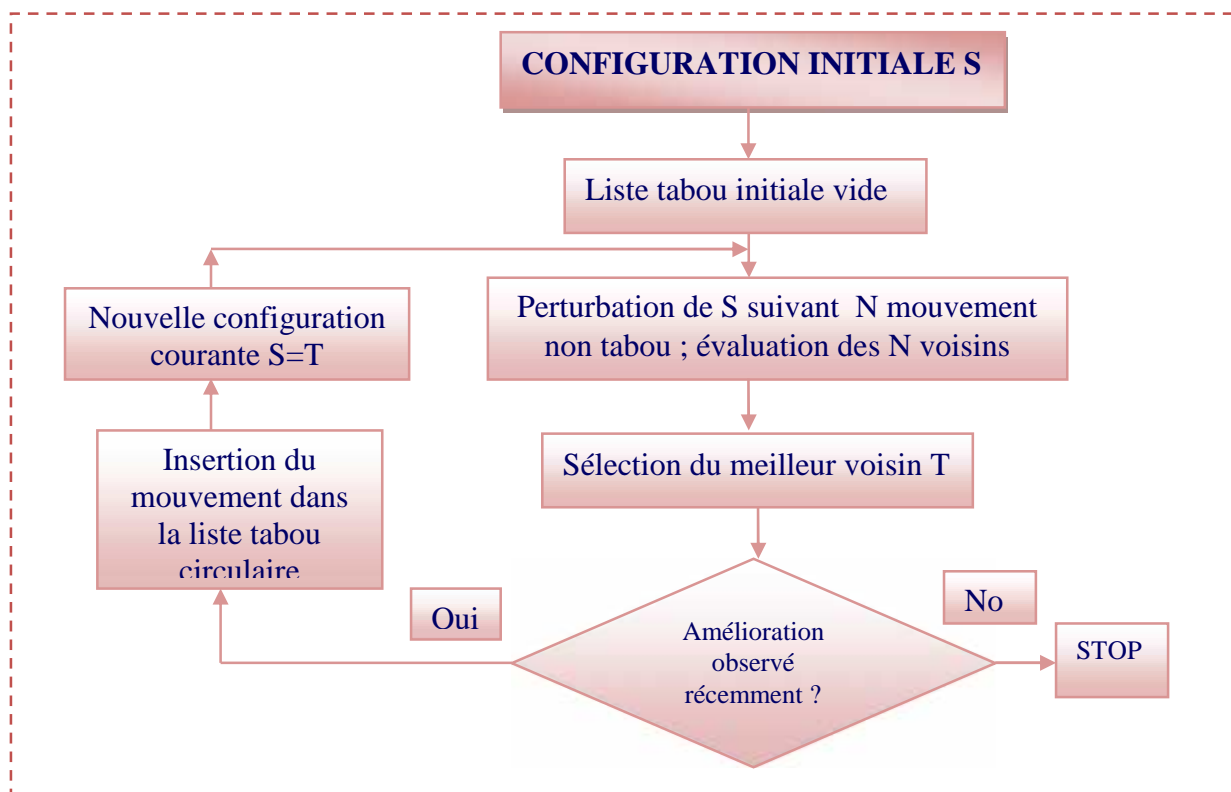
**Évaluation** des  $N$  voisins

**Sélection** du meilleur voisin  $t$

**Actualisation** de la meilleure position connue  $s_*$

**Insertion** du mouvement  $t$  !  $s$  dans la liste tabou

$s = t$  **Fin Tant que.**

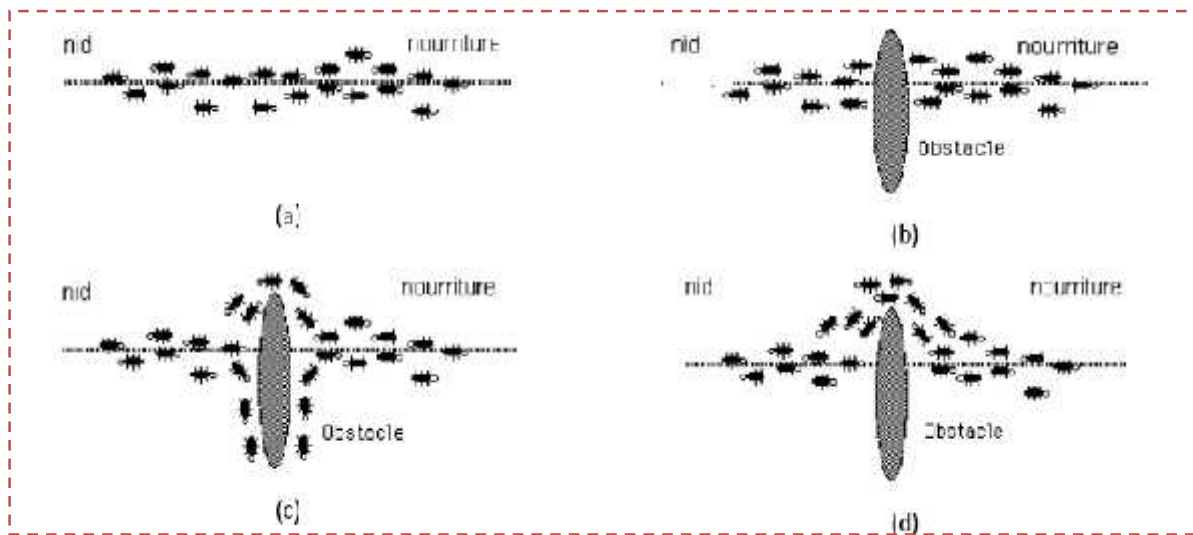


**Figure 2.5:** Diagramme d'Algorithmes de recherche de Tabou.

#### 2.2.5.4 La méthode de colonies de fourmi

Cet algorithme est encore inspiré de la nature et de son organisation, et a été mis au point par Dorigo au début des années 90. Son principe repose sur le comportement particulier des fourmis qui, lorsqu'elles quittent leur fourmilière pour explorer leur environnement à la recherche de nourriture [14][16], finissent par élaborer des chemins qui s'avèrent fréquemment être les plus courts pour aller de la fourmilière à une source de nourriture intéressante.





**Figure 2.6:** Détermination du plus court chemin par une colonie de fourmis.

- (a) Situation initiale,
- (b) Introduction d'un obstacle,
- (c) Recherche du chemin optimal,
- (d) Prédominance du chemin optimal.

Plus le taux de phéromone à un endroit donné est important, plus une fourmi va avoir tendance à être attirée par cette zone. Les fourmis qui sont arrivées le plus rapidement au nid en passant par la source de nourriture sont celles qui ont emprunté la branche la plus courte du trajet. Il en découle donc que la quantité de phéromones sur ce trajet est plus importante que sur le trajet plus long. De ce fait, le plus court chemin a une probabilité plus grande d'être emprunté par les fourmis que les autres chemins et sera donc, à terme, emprunté par toutes les fourmis.

#### 🚩 Algorithme de colonie de fourmi

**Tant que** le critère d'arrêt n'est pas atteint faire

**Pour** k=1 à m faire

**Choisir** une ville au hasard

**Pour** chaque ville non visitée i faire

**Choisir** une ville j, dans la liste  $J_k^i$  des villes restantes selon .....

$$p_{ij}^k(t) = \begin{cases} \frac{(\tau_{ij}(t))^\alpha (n_{ij})^\beta}{\sum_{r \in J_k^i} (\tau_{ir}(t))^\alpha (n_{ir})^\beta} & \text{si } j \in J_k^i \\ 0 & \text{si } j \notin J_k^i \end{cases}$$

**Fin Pour**

**Déposer** une piste  $\tau_{ij}(t)$  sur le trajet  $T_k(t)$  conformément à

$$\Delta \tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k(t)} & \text{si } (i, j) \in T^k(t) \\ 0 & \text{si } (i, j) \notin T^k(t) \end{cases}$$

**Fin Pour**

Évaporer les pistes selon

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t)$$

**Fin Tant que.**

### **2.2.5.5 la méthode d'optimisation par essaim particulaire**

L'optimisation par essaim de particule est une méthode d'optimisation stochastique, pour les fonctions non-linéaires, basée sur la reproduction d'un comportement social et développée par le Dr .EBERHART et le Dr KENNEDY [9] [11] en 1995. L'origine de cette méthode vient des observations faites lors des simulations informatiques de vols groupés d'oiseaux et de bancs de poissons .Ces simulations ont mis en valeur la capacité des individus d'un groupe en mouvement à conserver une distance optimale entre eux et à suivre un mouvement global par rapport aux mouvements locaux de leur voisinage.

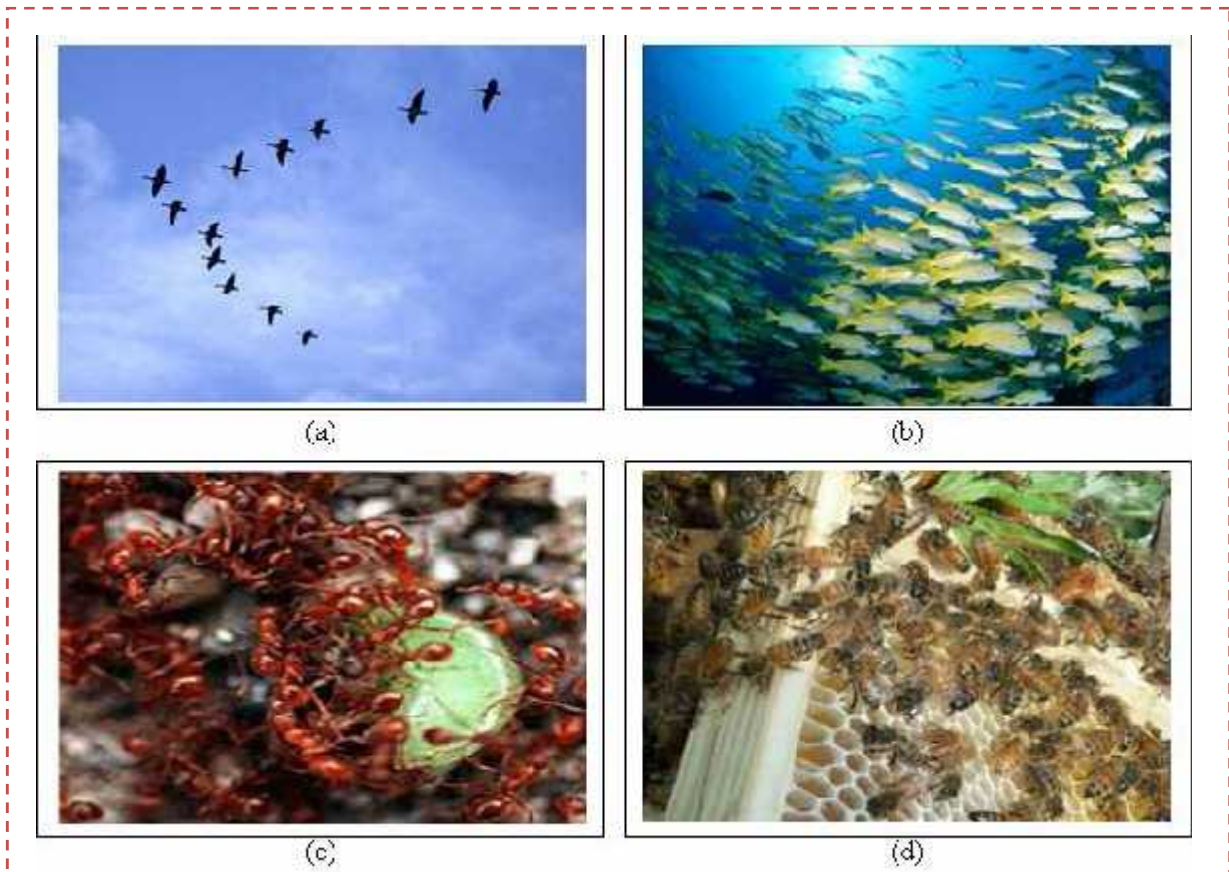
Parmi ces algorithmes, ce chapitre présent un intérêt particulier est apporté à la méthode d'Optimisation par Essaim Particulaire (OEP), qui constitue le sujet principal de ce travail de thèse et qu'an présentera en détails dans ce chapitre

### **2.3 Optimisation par Essaim Particulaire (OEP)**

L'optimisation par essaim de particules est une technique évolutionnaire qui utilise "une population" de solutions candidates pour développer une solution optimale au problème.

Le comportement de l'essaim doit être mesuré par une fonction fitness (aptitude) définie par l'utilisateur (Clerc et al. 2001, Dutot et al. 2002, Ji et al. 2007, Kennedy et al. 1995 et 2001) [11] [8]. L'OEP diffère des autres méthodes de calcul évolutionnaire de façon que les membres de la population appelés "particules", sont dispersées dans l'espace du problème (Kennedy et al. 1995 et 2001).décrit en se plaçant du point de vue d'une Particule (Kennedy et al. 2001, Omran 2004, Van den Bergh 2002, Venter et al. 2002).

Au départ de l'algorithme, un essaim est réparti au hasard dans l'espace de recherche, chaque particule ayant également une vitesse aléatoire [10].



**Figure 2.7:** Groupe de : (a) oiseaux, (b) poissons, (c) fourmis, (d) abeilles.

### 2.3.1 Principe général de la méthode

L'Optimisation par Essaim Particulaire (OEP) est inspirée du comportement social des animaux évoluant en essaim. L'exemple le plus souvent utilisé est le comportement des bancs de poissons. En effet, on peut observer chez ces animaux des dynamiques de déplacement relativement complexes [5], alors qu'individuellement chaque individu a une intelligence limitée et une connaissance seulement locale de sa situation dans l'essaim. Un individu de l'essaim n'a pour connaissance que la position et la vitesse de ses plus proches voisins [2]. Chaque individu utilise donc, non seulement, sa propre mémoire, mais aussi l'information locale sur ses plus proches voisins pour décider de son propre déplacement. Des règles simples, telles que « aller à la même vitesse que les autres » se déplacer dans la même direction, ou encore « rester proche de ses voisins » sont des exemples de comportements qui suffisent à maintenir la cohésion de l'essaim. Le processus de recherche est basé sur deux règles [7]:

- \* Chaque particule est dotée d'une mémoire qui lui permet de mémoriser le meilleur point par lequel elle est déjà passée et elle a tendance à retourner vers ce point.

- \* Chaque particule est informée du meilleur point connu au sein de son voisinage et elle va tendre à aller vers ce point.

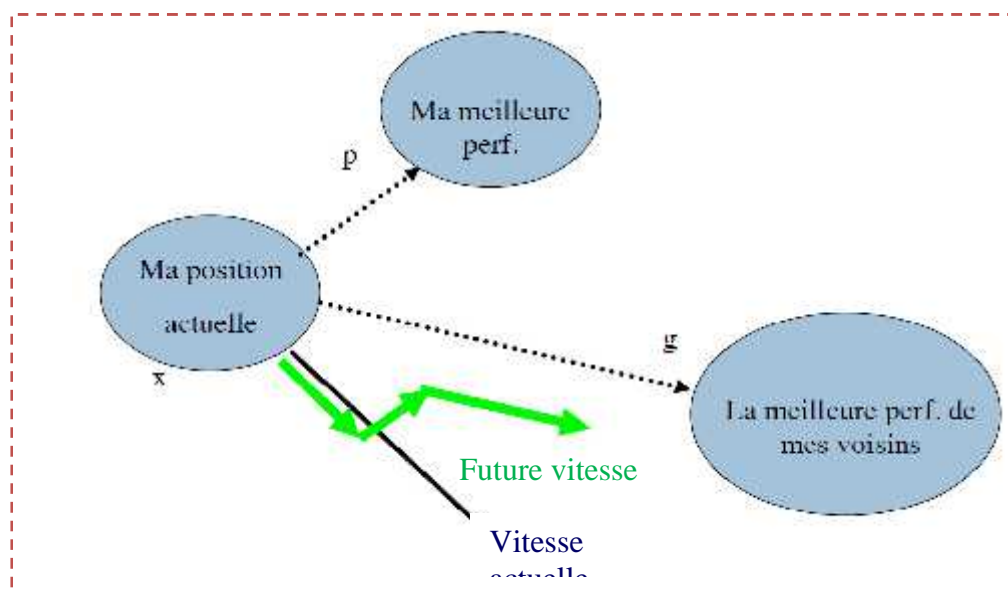
Alors que le déplacement d'une particule est influencé par les trois composantes suivantes:

- Une composante physique : la particule tend à suivre sa direction courante de déplacement.
- Une composante cognitive : la particule tend à se diriger vers le meilleur site par lequel elle est déjà passée.
- Une composante sociale : la particule tend à se fier à l'expérience de ses congénères et, ainsi, à se diriger vers le meilleur site déjà atteint par ses voisins [2].

### 2.3.2 Principales caractéristiques

Ce modèle présente quelques propriétés intéressantes, qui en font un bon outil pour de nombreux problèmes d'optimisation, particulièrement les problèmes fortement non linéaires, continus ou mixtes (certaines variables étant réelles et d'autres entières) :

- il est facile à programmer, quelques lignes de code suffisent dans n'importe quel langage évolué ;
- il est robuste (de mauvais choix de paramètres dégradent les performances, mais n'empêchent pas d'obtenir une solution).



**Figure 2.8:** Schéma de principe du déplacement d'une particule.

Pour réaliser son prochain mouvement, chaque particule combine trois tendances : suivre sa vitesse propre, revenir vers sa meilleure performance, aller vers la meilleure performance de ses congénères [4].

### 2.3.3 Modélisation mathématique de la méthode

Considérons une population composée de  $K$  particules. Les équations formalisant le mouvement des particules sont données par :

$$v(t+1) = w.v_i(t) + c_1.r_1(t).(p_{bi}(t) - x_i(t)) + c_2.r_2(t).(p_g(t) - x_i(t)) \quad (2.1)$$

$$X_i(t+1) = X_i(t) + v_i(t+1) \quad (2.2)$$

Chaque particule  $p_i$  ( $i = 1, 2, \dots, K$ ) dans le swarm (population) est caractérisé par :

- Sa position courante  $x_i(t) \in R^d$ , qui réfère à une solution candidate pour le problème d'optimisation considéré à l'itération  $t$ .

- Sa vitesse  $v_i(t) \in R^d$ .

- La meilleure position  $p_{bi}(t) \in R^d$  identifiée durant sa trajectoire antécédente. Soit  $p_g(t) \in R^d$  la meilleure position globale identifiée dans le processus de recherche pour toutes les particules dans le swarm. La position optimal est mesurée avec une fonction dite *fitness* (fonction objectif) définit suivant le problème d'optimisation. Durant le processus d'optimisation, les particules se déplacent suivant les équations (2.1) et (2.2).

- $r_1$  et  $r_2$  sont des variables aléatoires générés d'une distribution uniforme dans l'intervalle  $[0, 1]$  afin de fournir un poids stochastique aux différentes composantes participant dans la définition de la vitesse de la particule.

- $c_1$  et  $c_2$  sont deux constantes d'accélération régulant les vitesses relatives par rapport aux meilleurs positions locales et globales. Ces paramètres sont considérés comme des facteurs d'échelle utilisés pour déterminer les mouvements relatifs de la meilleure position de la particule ainsi que de la meilleure position globale. Ce sont des facteurs qui déterminent le degré d'influence des positions passées de la particule elle mêmes et celles des autres particules dans le swarm.

- L'inertie  $w$  qui est utilisée comme un compromis entre l'exploration locale et globale du swarm (contrôle la capacité d'exploration de l'algorithme). Des valeurs élevées de ce paramètre permettent une bonne exploration globale, tandis que pour des valeurs minimales une recherche fine est réalisée.

L'équation (2.1) permet le calcul de la vitesse à l'itération  $(t+1)$  pour chaque particule en combinant linéairement la position et la vitesse (à l'itération  $t$ ) et les distances qui séparent la position courante de la particule de son antécédente meilleure position et la meilleure position globale, respectivement. La mise à jour de la position de la particule est réalisée à travers l'équation (2.2). Les équations (2.1) et (2.2) sont itérées jusqu'à ce que la convergence soit atteinte [2].

### 2.3.4 Paramètres importants qu'influencent sur l'OEP

L'algorithme OEP comprend plusieurs paramètres de réglage, Ces paramètres de réglage qui permettent d'agir sur le compromis exploration – exploitation.

L'exploration est la capacité de tester différentes régions de l'espace à la recherche de bonnes solutions candidates. L'exploitation est la capacité de concentrer la recherche autour des solutions prometteuses afin de s'approcher le plus possible de l'optimum.

Parmi ces paramètres on cite :

- a) le nombre de particules de l'essaim, noté  $K$  ;
- b) la topologie et la taille du voisinage d'une particule qui définissent son réseau social ;
- c) l'inertie d'une particule, notée  $w$  ;
- d) les coefficients de confiance, notés  $c_1$  et  $c_2$ , qui pondèrent le comportement conservateur (la tendance à retourner vers la meilleure solution visitée) et le panurgisme (la tendance à suivre le voisinage).

#### Nombre de particules

C'est la quantité de particules allouées à la résolution du problème et qui dépend essentiellement de deux paramètres :

La *taille* de l'espace de recherche et le *rapport* entre les capacités de calcul de la machine et le temps maximum de recherche. Il n'y a pas de règle pour déterminer ce paramètre, faire de nombreux essais permet de se doter de l'expérience nécessaire à l'appréhension de ce paramètre.

#### Topologie de voisinage

Le voisinage constitue la structure du réseau social. à l'intérieur de laquelle les particules communiquent entre elles. Il y a deux grandes méthodes pour définir le voisinage d'une particule, soit un voisinage « géographique », qui doit être recalculé à chaque pas de temps [9], et qui suppose l'existence d'une distance dans l'espace de recherche, soit un voisinage « social » défini une seule fois, et qui est le plus utilisé, pour plusieurs raisons :

- Il est plus simple à programmer.
- Il est moins coûteux en temps de calcul.

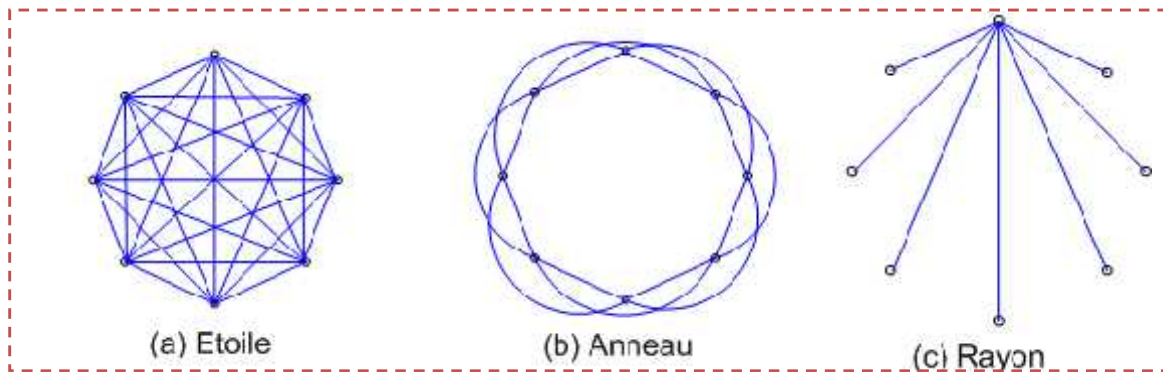
De toute façon, en cas de convergence, un voisinage social tend à devenir un voisinage géographique.

Il existe de nombreuses Topologies dont les suivantes sont les plus utilisées :

– Topologie en étoile (*figure 2.3 (a)*) : le réseau social est complète, chaque particule est attirée vers la meilleure particule notée (gbest) et communique avec les autres.

– Topologie en anneau (*figure 2.3 (b)*) : chaque particule communique avec  $n$  ( $n = 3$ ) voisines immédiates. Chaque particule tend à se déplacer vers la meilleure dans son voisinage local notée ( $l_{best}$ ).

– Topologie en rayon (*figure 2.3 (c)*) : une particule "centrale" est connectée à tous les autres. Seule cette particule centrale ajuste sa position vers la meilleure, si cela provoque une amélioration l'information est propagée aux autres [12] [13].



**Figure 2.9:** Les topologies du voisinage d'un essaim.

#### 🚩 Coefficients de confiance

Les variables de confiance pondèrent les tendances de la particule à vouloir suivre son instinct de conservation ou son panurgisme. Les variables aléatoires  $r_1$  et  $r_2$  peuvent être définis de la façon suivante :

$$\begin{cases} r_1 = r1 \cdot c1 \\ r_2 = r2 \cdot c2 \end{cases}$$

Où  $r_1$  et  $r_2$  suivent une loi de distribution uniforme sur  $[0..1]$  et  $c_1$  et  $c_2$  sont des constantes positives déterminées de façon empirique et suivant la relation  $c_1 + c_2 = 4$ , avec :

$c_1$  : la correction locale.

$c_2$  : la correction globale.

Un choix judicieux de ces deux paramètres permet éventuellement d'avoir une meilleure convergence, ainsi qu'une recherche diversifiée, soit un balayage global de l'espace de recherche.

#### 🚩 Facteur d'inertie

Pour contrôler l'influence de la vitesse obtenue au pas précédent on peut introduire un facteur d'inertie «  $w$  » qui décroît en fonction du temps.

$$\vec{v}_i(t + 1) = w \cdot \vec{v}_i(t) + \dots_1(\vec{x}_{pbesti} - \vec{x}_i(t)) + \dots_2(\vec{x}_{gbesti} - \vec{x}_i(t)) \quad (2.3)$$

$\vec{x}_{pbesti}$  : La position de la meilleure solution par laquelle est passée la  $i^{\text{ème}}$  particule.

$\vec{x}_{gbesti}$  : La position de la meilleure solution connue dans le voisinage de la  $i^{\text{ème}}$  particule

$\dots_1(\vec{x}_{pbesti} - \vec{x}_i)$  : Correspond à la composante cognitive du déplacement,  $c1$  contrôle le comportement cognitif de la particule

$\dots_2(\vec{x}_{gbest} - \vec{x}_i)$  : Correspond à la composante sociale du déplacement,  $c2$  contrôle l'aptitude sociale de la particule.

Un grand facteur d'inertie provoque une grande exploration de l'espace de recherche alors qu'un petit facteur d'inertie concentre la recherche sur un petit espace.

La convergence n'est pas assurée quelques soient les valeurs de  $w$  et également de  $c1$  et  $c2$  (van den Bergh, 2002), les études expérimentales montrent :

$$0.5 \cdot (c1 + c2) - 1 < w < 1 \quad (2.4)$$

Les études ont démontré une meilleure convergence pour  $w \in [0.8, 1.2]$ . Au delà de valeur : 1.2 l'algorithme tend à avoir certaines difficultés à converger [13].

Enfin, il est également possible de faire diminuer le facteur d'inertie au cours du temps, c'est-à-dire faire varier ce paramètre au cours du déroulement de l'algorithme pour assurer une recherche diversifiée. De bons résultats ont été trouvés pour une valeur décroissante linéairement de 0.9 à 0.4.

### Vitesse maximale et coefficient de constriction

Pour éviter que les particules se déplacent trop rapidement d'une région à une autre dans l'espace de recherche, on fixe une vitesse maximale  $V_{max}$ . Ainsi si «  $v_{ij}(t)$  » est la vitesse de la particule  $P_i$  au temps  $t$  dans la dimension  $j$

$$v_{ij}(t) = V_{max} \text{ si } v_{ij}(t) > V_{max} \text{ et } v_{ij}(t) = -V_{max} \text{ si } v_{ij}(t) < -V_{max}$$

$V_{max}$  est généralement dépendant de l'échelle du problème.

$V_{max}$  n'est pas obligatoire (Clerc and Kennedy, 2002) [17]. si on utilise un coefficient de constriction (resserrement) :

$$\tau = \frac{2}{w - 2 + \sqrt{w^2 - 4w}}, w = w_1 + w_2, w > 4 \quad (2.5)$$

$$\vec{v}_i(t + 1) = \tau \cdot \left[ \vec{v}_i(t) + w_1 \cdot r_1 \cdot (\vec{x}_{pbesti} - \vec{x}_i(t)) + w_2 \cdot r_2 \cdot \dots_2(\vec{x}_{gbesti} - \vec{x}_i(t)) \right] \quad (2.6)$$

$w$  : L'utilisation d'un facteur de constriction.



On remarque que l'OEP avec coefficient de constriction est équivalente à l'OEP originale

Avec  $w, c1, c2, w_1, w_2 \Rightarrow \dots v_i = t \cdot w_i \cdot r_i$  Avec  $i=1,2$

### Notes importantes

- Il est à noter que le terme vitesse est ici abusif car les vecteurs  $\vec{v}_i$  ne sont pas homogènes à une vitesse. Il serait plus approprié de parler de « direction de déplacement ». Cependant, pour respecter l'analogie avec le monde animalier, les auteurs ont préféré utiliser le terme vitesse.

- La position des particules ainsi que leur vitesse initiale doivent être initialisées aléatoirement selon une loi uniforme sur  $[0..1]$ .

- L'algorithme doit alors arrêter de s'exécuter tant que l'un des critères de convergence suivants soit atteint :

- Nombre d'itération max a été atteint.
- La variation de la vitesse est proche de 0.
- La niveau de performance ou objectif est atteint.

### 2.3.5 Déroulement de l'algorithme

Chaque particule bouge dans l'espace de recherche, sachant qu'elle possède initialement une vitesse et une petite mémoire lui permettant de se souvenir de sa meilleure performance, en position et en valeur. Donc chaque particule dispose d'un groupe d'informaticiens, historiquement appelé son voisinage. A partir de ces informations, la particule doit décider de son prochain mouvement. Le déroulement des événements qui permettent la prise de cette décision peuvent être classés dans l'ordre suivant :

**Première étape :** on initialise l'essaim de particules dans l'espace de recherche. On peut le faire soit de manière aléatoire (ce qui n'est pas forcément si simple selon la forme de l'espace de recherche), soit de manière régulière. En général, le plus efficace est une combinaison des deux. On initialise également la vitesse de chaque particule.

**Deuxième étape :** évaluer la fonction fitness de chaque particule dans l'espace de recherche, de telle façon que cette fonction soit propre au système à optimiser, cependant, le reste de l'optimisation est indépendant du système physique qu'on veut optimiser.

**Troisième étape :** comparer la fitness de chaque particule avec la fitness précédente ( $f_{best}$ ), si sa valeur est meilleure alors, on remplace  $f_{best}$  par cette valeur, on garde sa position dans l'espace de recherche.

**Quatrième étape :** comparer la fitness évaluée avec la meilleure fitness de toute les particules ( $\mathbf{g}_{best}$ ), si cette valeur est meilleure, alors on la garde comme  $\mathbf{g}_{best}$  tout en gardant son index.

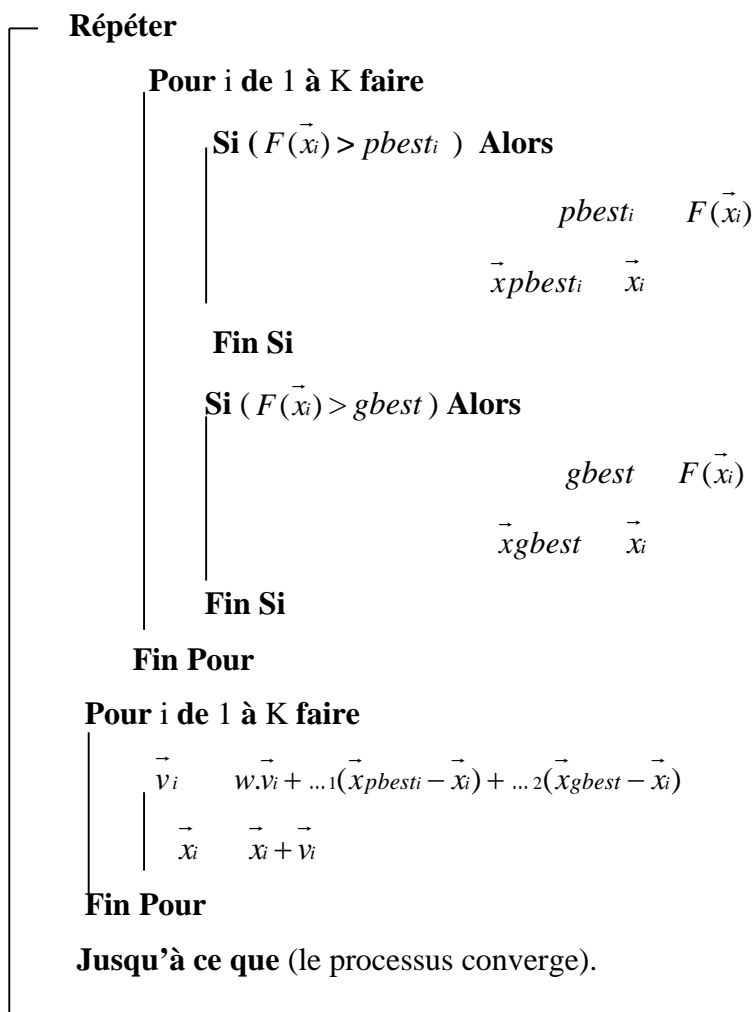
**Cinquième étape :** le renouvellement de la position et de la vitesse de chaque particule se fera selon les équations (2.2) et (2.3)

**Sixième étape :** si le critère d'arrêt n'est pas encore satisfait, on revient à l'étape 2, si non, on arrête le processus.

On peut représenter le déroulement de l'algorithme précédant suivant l'écriture suivante :

❖ **Algorithme avec une topologie en étoile**

[Initialisations] Initialiser aléatoirement la population



Avec :

K : Nombre de particules

$\vec{x}_i$  : Position de la particule Pi

$\vec{v}_i$  : Vitesse de la particule Pi

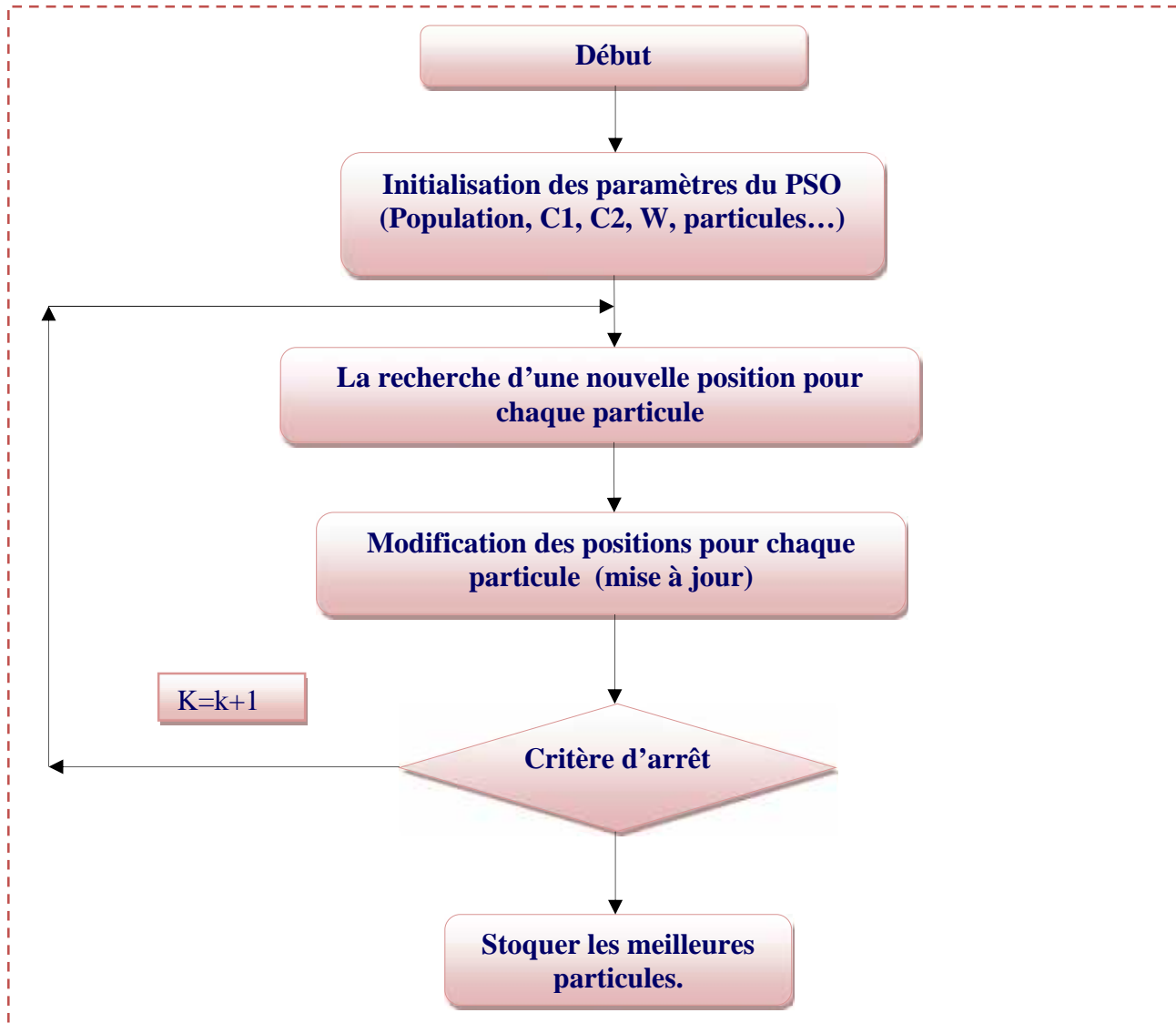
$F(\vec{x}_i(t))$  : Fitness obtenue pour la particule Pi

$p_{best_i}$  : Meilleure fitness obtenue pour la particule  $P_i$

$\vec{x}_{p_{best_i}}$  : Position de la particule  $P_i$  pour la meilleure fitness

$\vec{x}_{g_{best_i}}$  : Position de la particule ayant la meilleure fitness de toutes

$r_1$  et  $r_2$  : valeur aléatoire positive.  $w$  : Paramètre d'inertie.



**Figure 2.10:** Organigramme de l'algorithme de la méthode OEP.

#### 2.4 Exemple simple de la méthode OEP

A la fin de ce chapitre on fait une application de cette méthode pour résoudre une équation à deux variables :

$$f = (y+2)^2 + (x-5)^2 \quad (2.7)$$

Cette fonction admet un optimum global défini au point  $(-2,5)$  tel que  $f(-2,5) = 0$ .

Maintenant, on va résoudre ce problème avec la méthode OEP, et en donnant quatre cas d'exécution où les paramètres de contrôle de la méthode seront changés dans chaque cas, pour voir l'effet et l'importance du réglage de ces paramètres afin d'avoir des résultats plus précis.

• **Le premier cas** : Nombre de particules insuffisant

Initialisation des Paramètres :

L'espace de recherche :  $[-6,6]*[-6,6]$

**Nombre de particule :  $K= 5$**

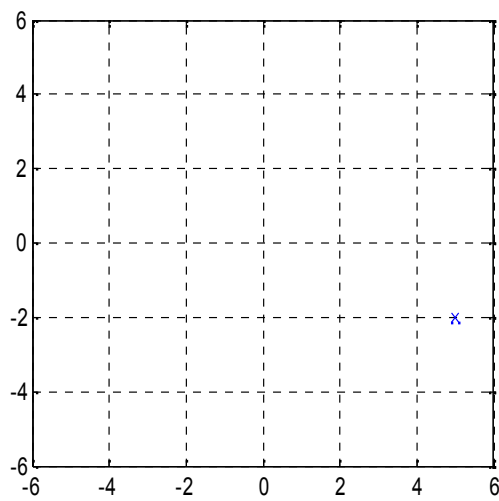
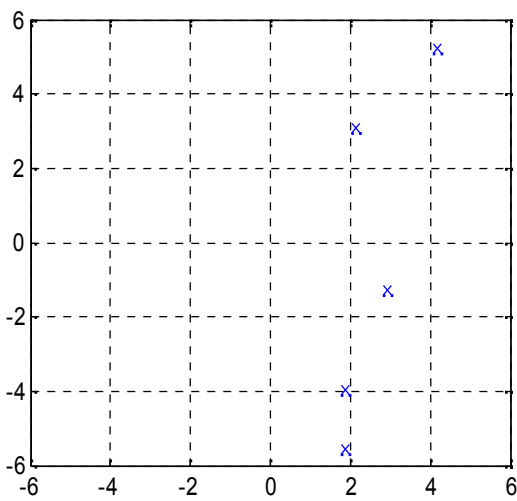
Nombre itération : 50

Paramètre d'inertie  $w = 0.9$

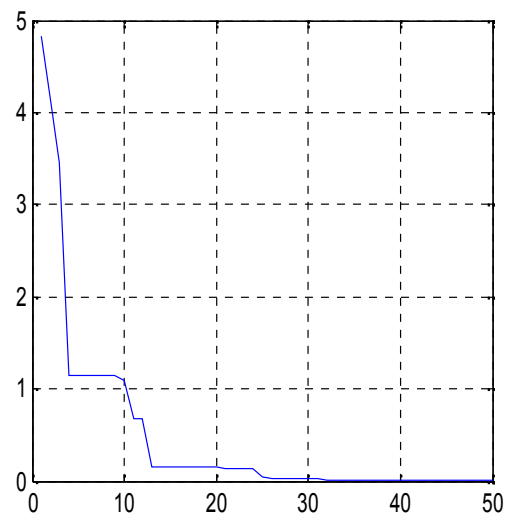
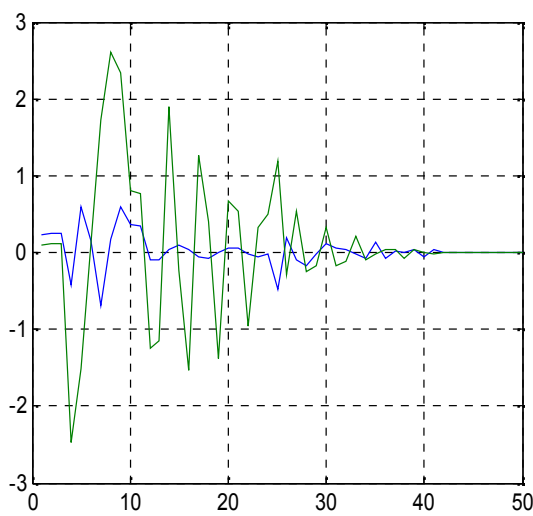
$C1 = 1.5$

$C2 = 1.9$

Les résultats sont représentés suivants :



**a)** Les particules après quelques itérations. **b)** Convergence des particules vers l'optimum.



**c)** Variation des vitesses a travers les itérations **d)** Evolution de la fonction fitness.

**Figure 2.11:** Cas d'un nombre de particules insuffisant.

Avec :

$$\begin{cases} X= 4.9999 \\ Y= -2.0001 \end{cases} , \text{ Erreur}=2.4373 \text{ e-}008.$$

• **Le deuxième cas** : facteur d'inertie important

On prend :

L'espace de recherche :  $[-6,6]*[-6,6]$

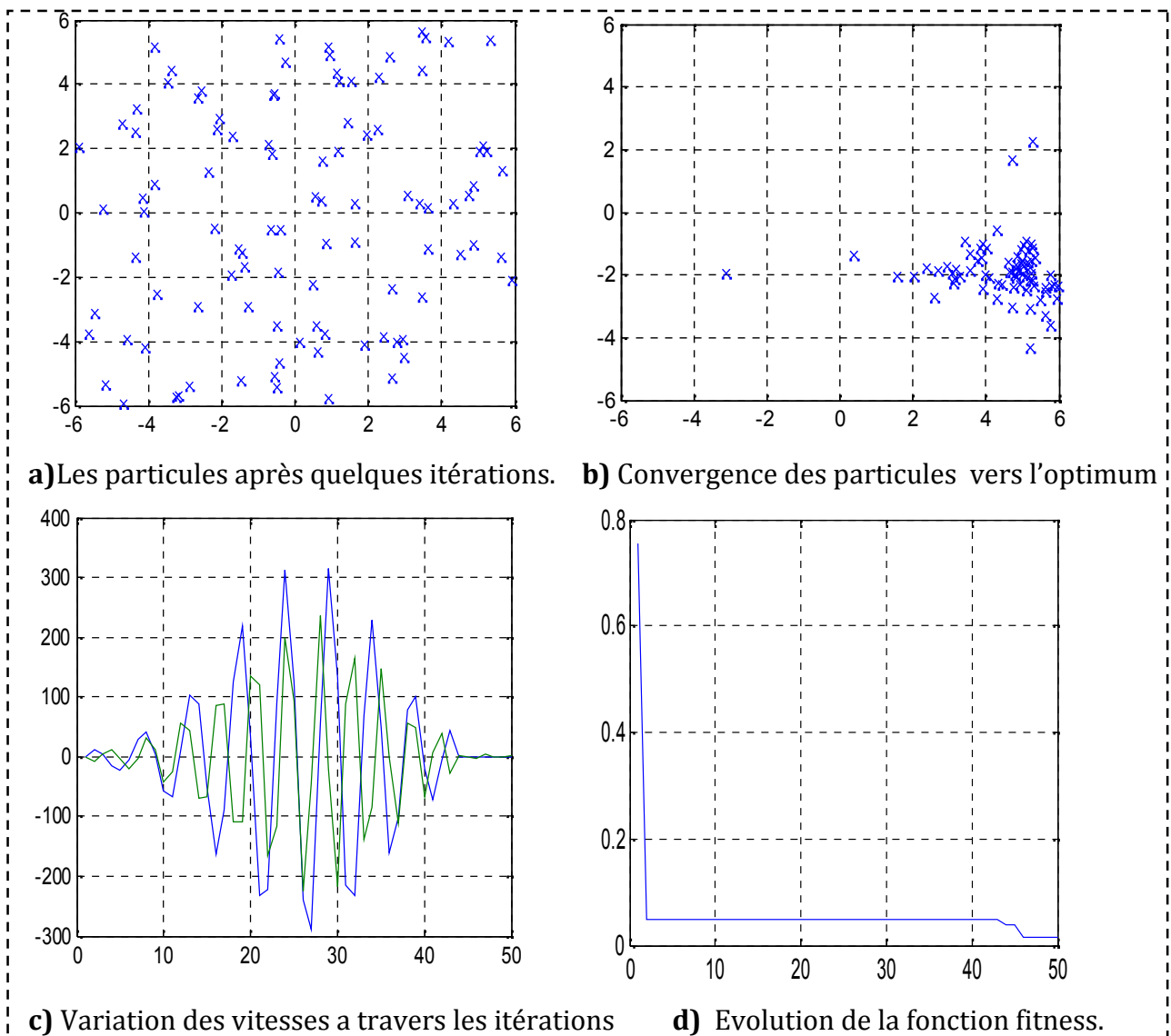
Nombre de particule  $K = 100$

Correction locale  $C1 = 1.5$ , Correction globale  $C2 = 1.9$

**Facteur d'inertie  $w = 1.5$**

Nombre itération : 50

On obtient :



**Figure 2.12** : Cas d'un facteur d'inertie très grand.

Avec :

$$\begin{cases} X=4.9880 ; \\ Y=-1.9775 ; \end{cases} , \text{ Erreur}= 0.0057.$$

• **Le troisième cas** : dimension de l'espace de recherche très grande

On prend :

**L'espace de recherche = [-400,400]\*[-400,400]**

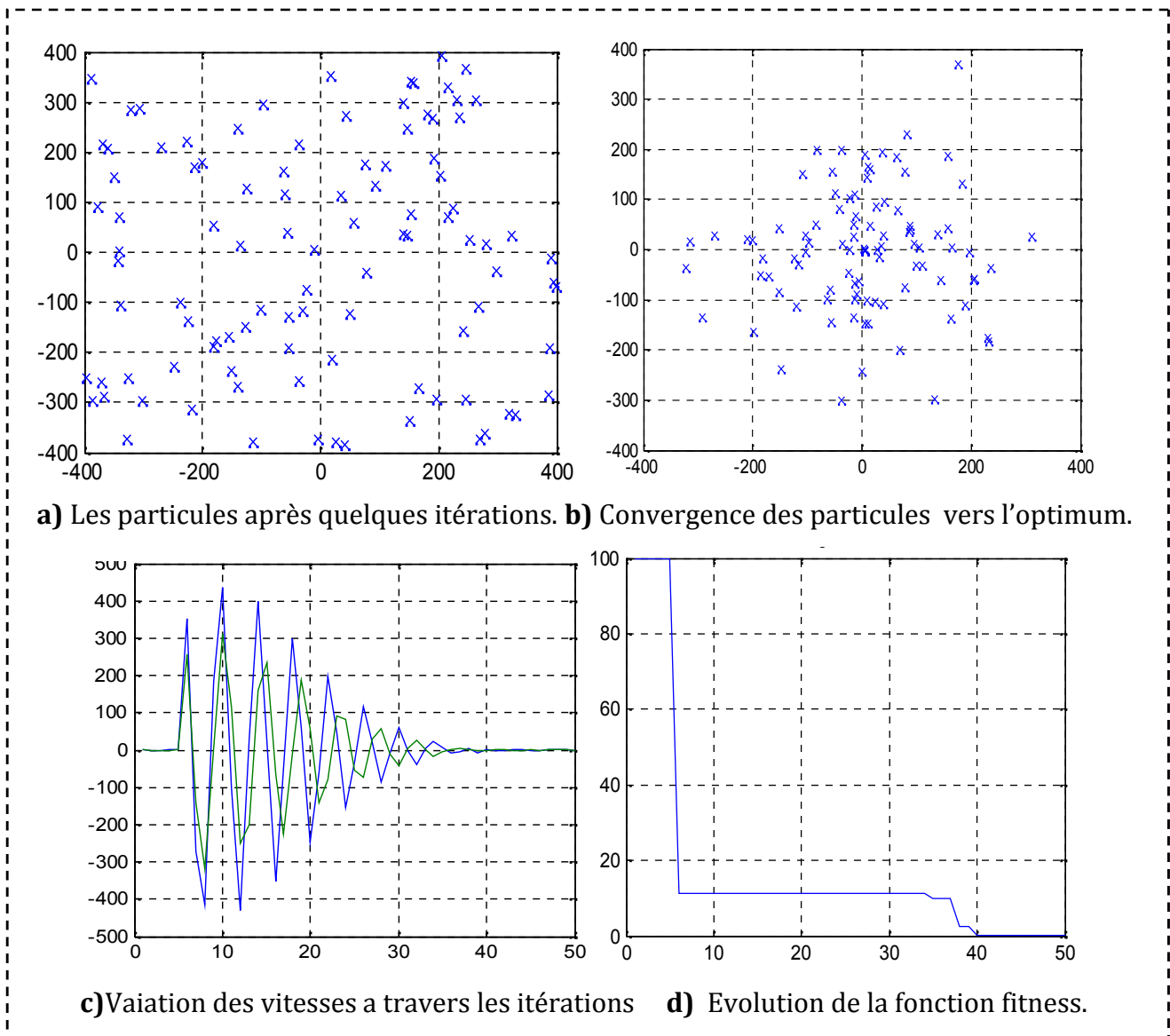
Nombre de particule  $K = 100$

Correction locale  $C1 = 1.5$ , Correction globale  $C2 = 1.9$

Facteur d'inertie  $w = 0.9$

Nombre itération : 50

On obtient :



**a)** Les particules après quelques itérations. **b)** Convergence des particules vers l'optimum.

**c)** Vaiaation des vitesses a travers les itérations **d)** Evolution de la fonction fitness.

**Figure 2.13** : Cas d'un espace de recherche vaste.

Avec :

$$\left\{ \begin{array}{l} X=4.9079 \\ Y=-1.9970 \end{array} \right. , \text{ Erreur= } 0.0085.$$

• **Le quatrième cas** : compromis entre l'ensemble des paramètres

On prend :

L'espace de recherche :  $[-6,6]*[-6,6]$

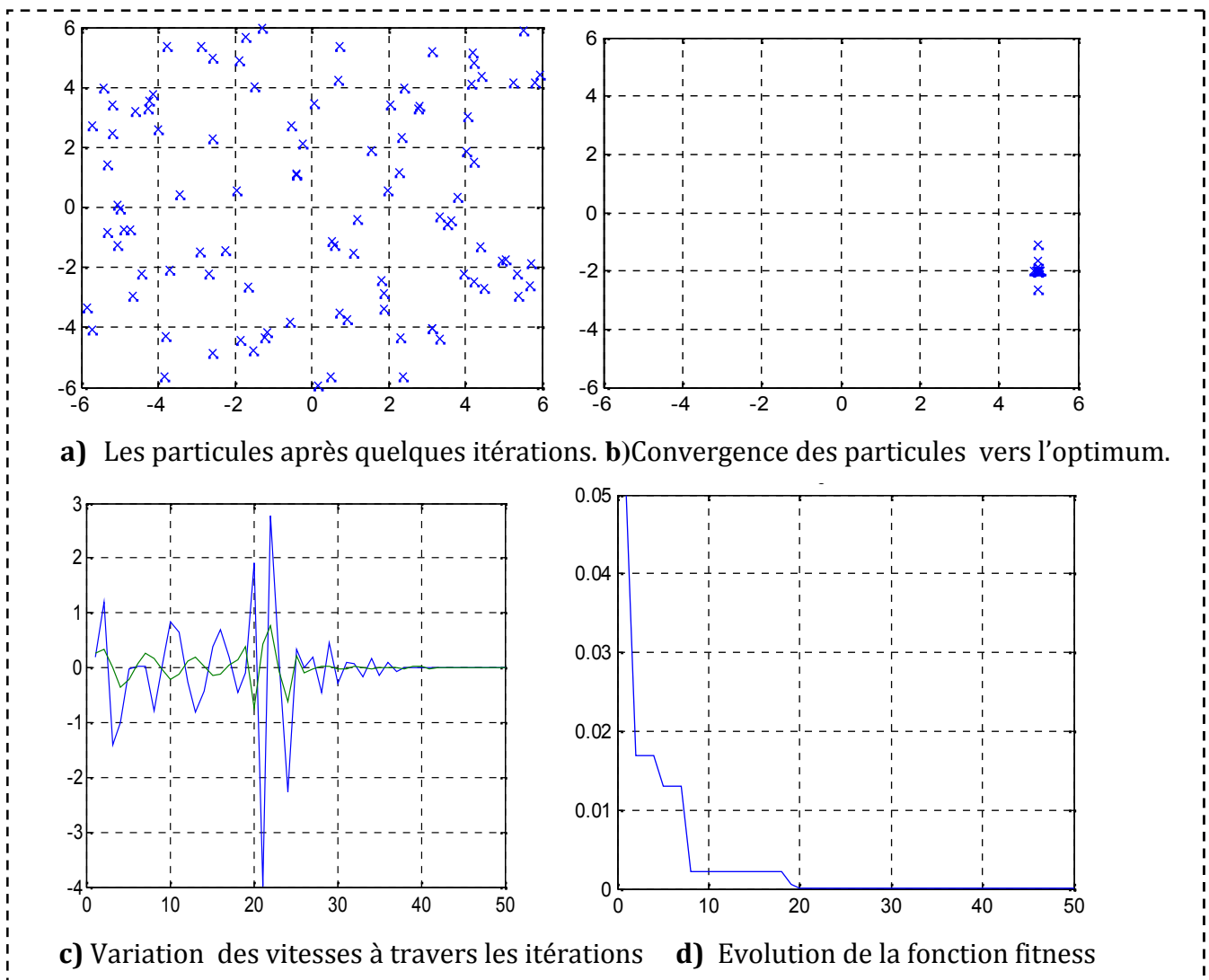
Nombre de particule  $K = 100$

Correction locale  $C1 = 1.5$ , Correction globale  $C2 = 1.9$

Facteur d'inertie  $w = 0.9$

Nombre itération : 50

*On obtient :*



**Figure 2.14:** Cas où les paramètres de contrôle sont assez réglés.

Avec :

$$\left\{ \begin{array}{l} X=5.0000, \\ Y=-2.0000 \end{array} \right. , \text{ Erreur} = 9.7079e-014.$$

## 2.4 Exemple complexe de la méthode OEP

### 2.5

On utilise notre méthode pour résoudre une équation complexe à deux variables :

$$F(x, y) = x^2 + y^2 + 25 \cdot (\sin(x)^2 + \sin(y)^2) \quad (2.8)$$

Cette fonction admet un optimum global défini au point (0,0) tel que  $f(0,0) = 0$ . Maintenant, on va résoudre ce problème avec la méthode OEP, et en donnant quatre cas d'exécution où les paramètres de contrôle de la méthode seront changés dans chaque cas, pour voir l'effet et l'importance du réglage de ces paramètres afin d'avoir des résultats plus précis. A chaque fois on va changer un paramètre et voir son influence sur la méthode.

#### • Le premier cas : influence des coefficients de confiance

Initialisation des Paramètres :

L'espace de recherche :  $[-2\pi, 2\pi]$

Nombre de particule : 40.

Nombre itération : 100.

Paramètre d'inertie  $w = 0.9$ .

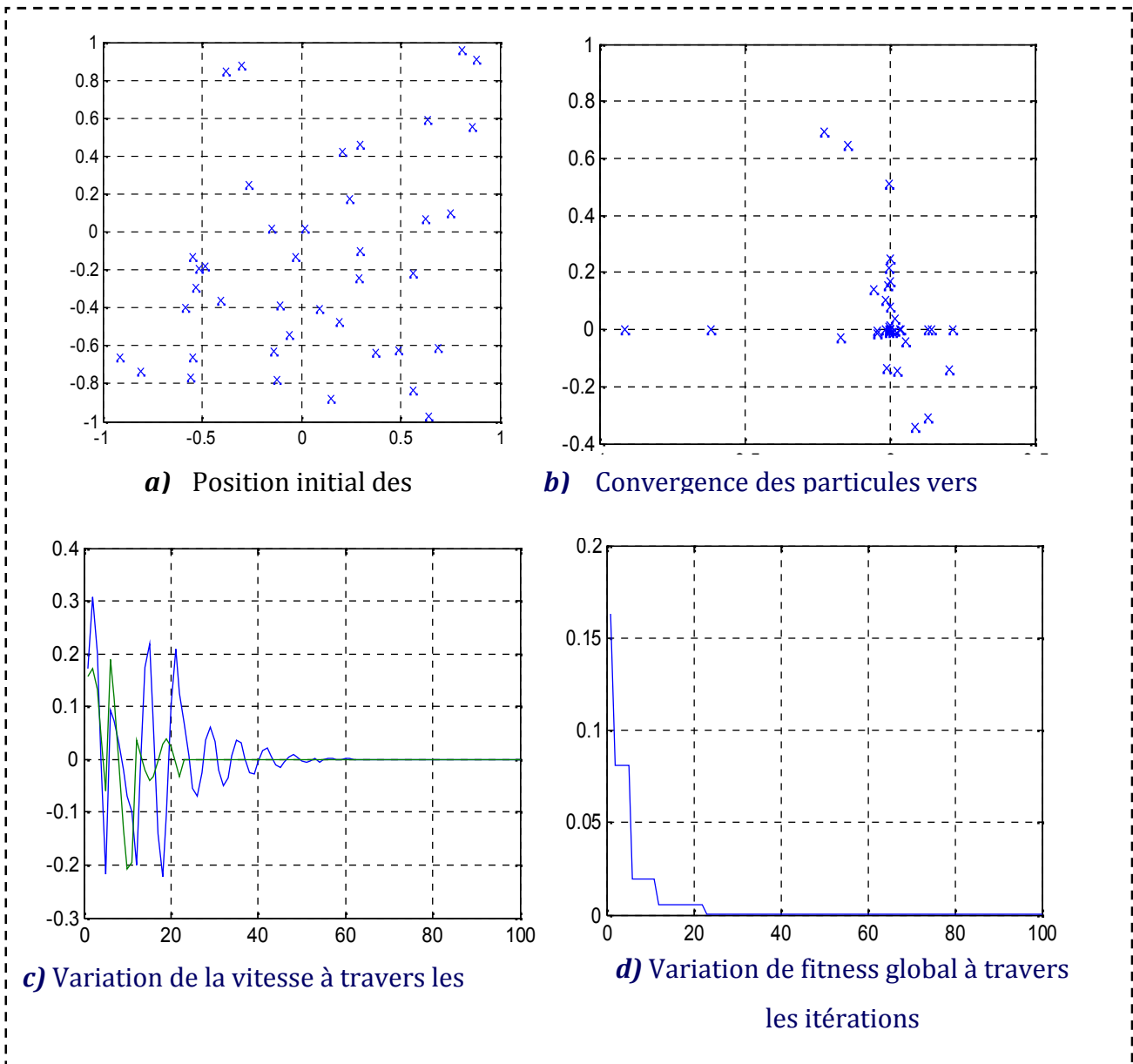
**Les coefficients d'accélération :**

$$c1 = 1.$$

$$c2 = 0.$$



Les résultats sont représentés dans la (figure 2.15) suivante :



**Figure 2.15:** Cas de modification dans les paramètres de confiances.

Avec :

$$\left\{ \begin{array}{l} X=-0.5835, \\ Y=0.0158, \end{array} \right. \quad \text{Erreur} = 8.8599\text{e-}006$$

- **Le deuxième cas** : influence du facteur d'inertie :

On prend:

L'espace de recherche :  $[-2\pi, 2\pi]$

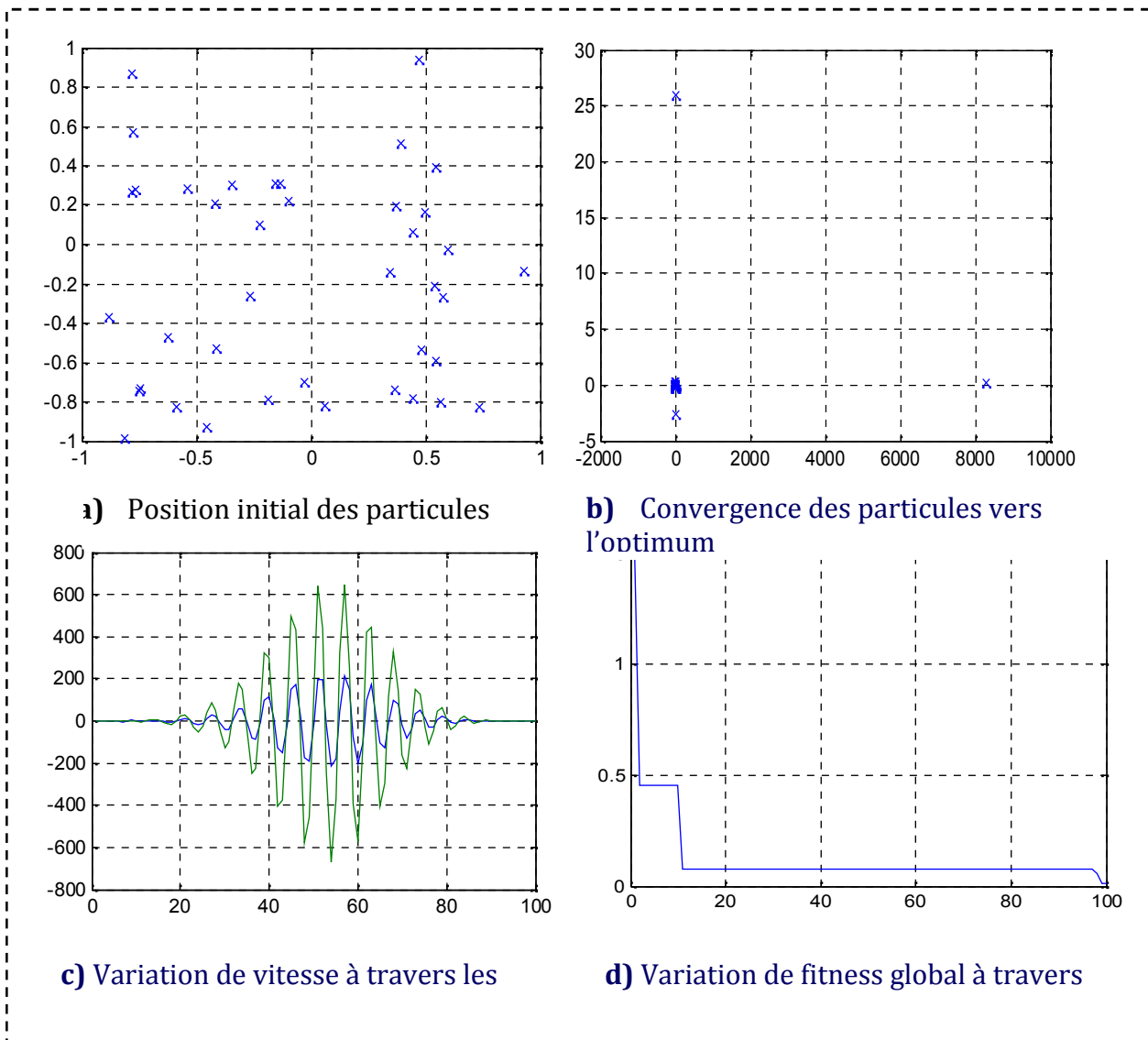
Nombre de particule = 40

Nombre itération = 100

**Facteur d'inertie  $w = 1.5$**

Correction locale  $c1 = 1.5$

Correction globale  $c2 = 1.9$



**Figure 2.16:** Cas d'un facteur d'inertie très grand.

Avec :

$$\begin{cases} X = -0.3124, \\ Y = 0.4840, \quad \text{Erreur} = 8.6274e-016 \end{cases}$$

- **Le troisième cas** : L'influence de l'espace de recherche

On prend :

**L'espace de recherche** :  $[-25\pi, 25\pi]$ ,

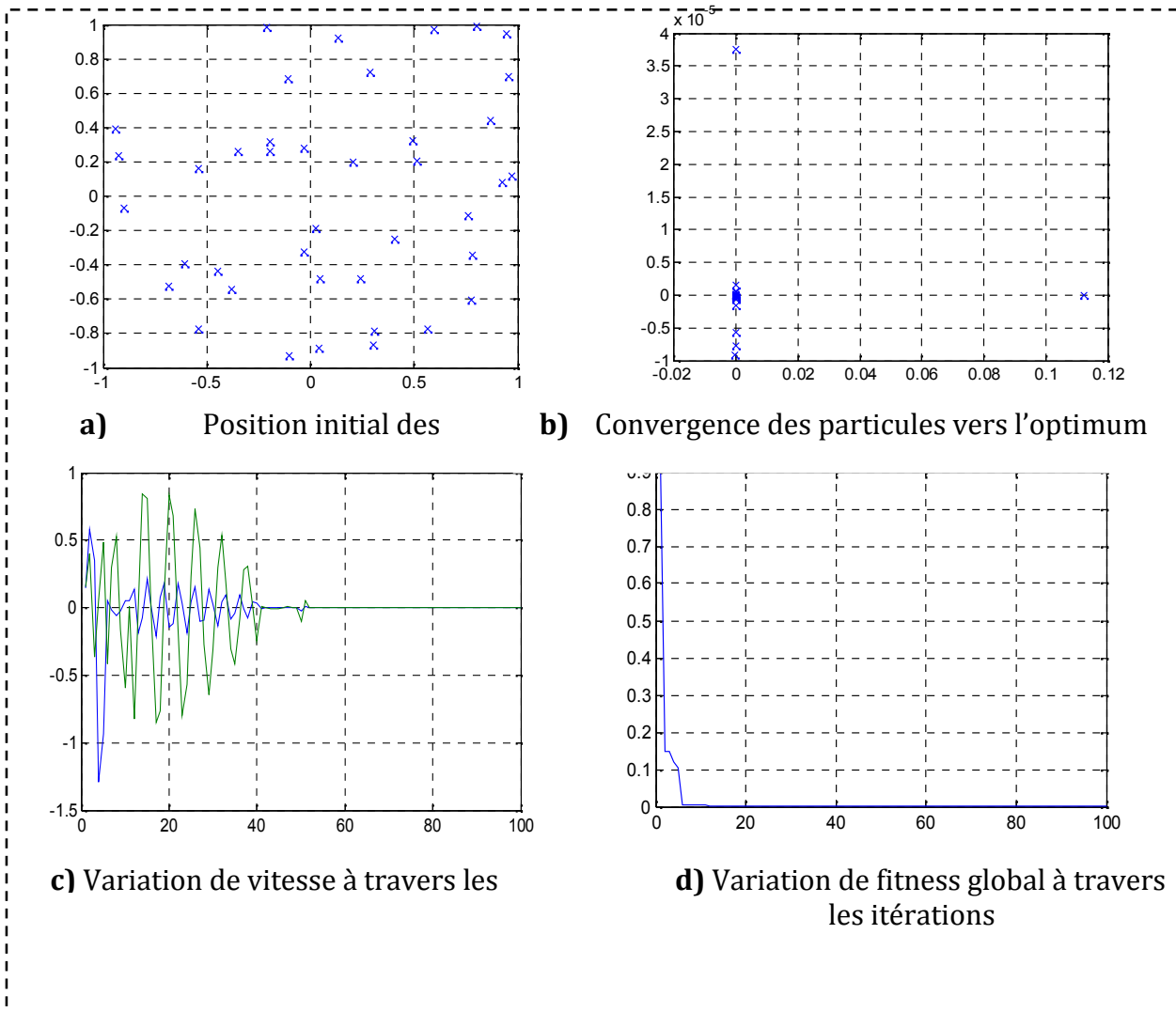
Nombre de particule = 40,

Nombre itération = 100,

Correction locale  $c1 = 1.5$ ,

Correction globale  $c2 = 1.9$

Facteur d'inertie  $w = 0.9$



**Figure 2.17:** L'influence de l'espace de recherche.

Avec :

$$\left\{ \begin{array}{l} X= 0.0317, \\ Y= -0.2204, \end{array} \right. \quad \text{Erreur}= 1.2893e-016$$

- **Le quatrième cas** : compromis entre l'ensemble des paramètres:

L'espace de recherche :  $[-2\pi, 2\pi]$ ,

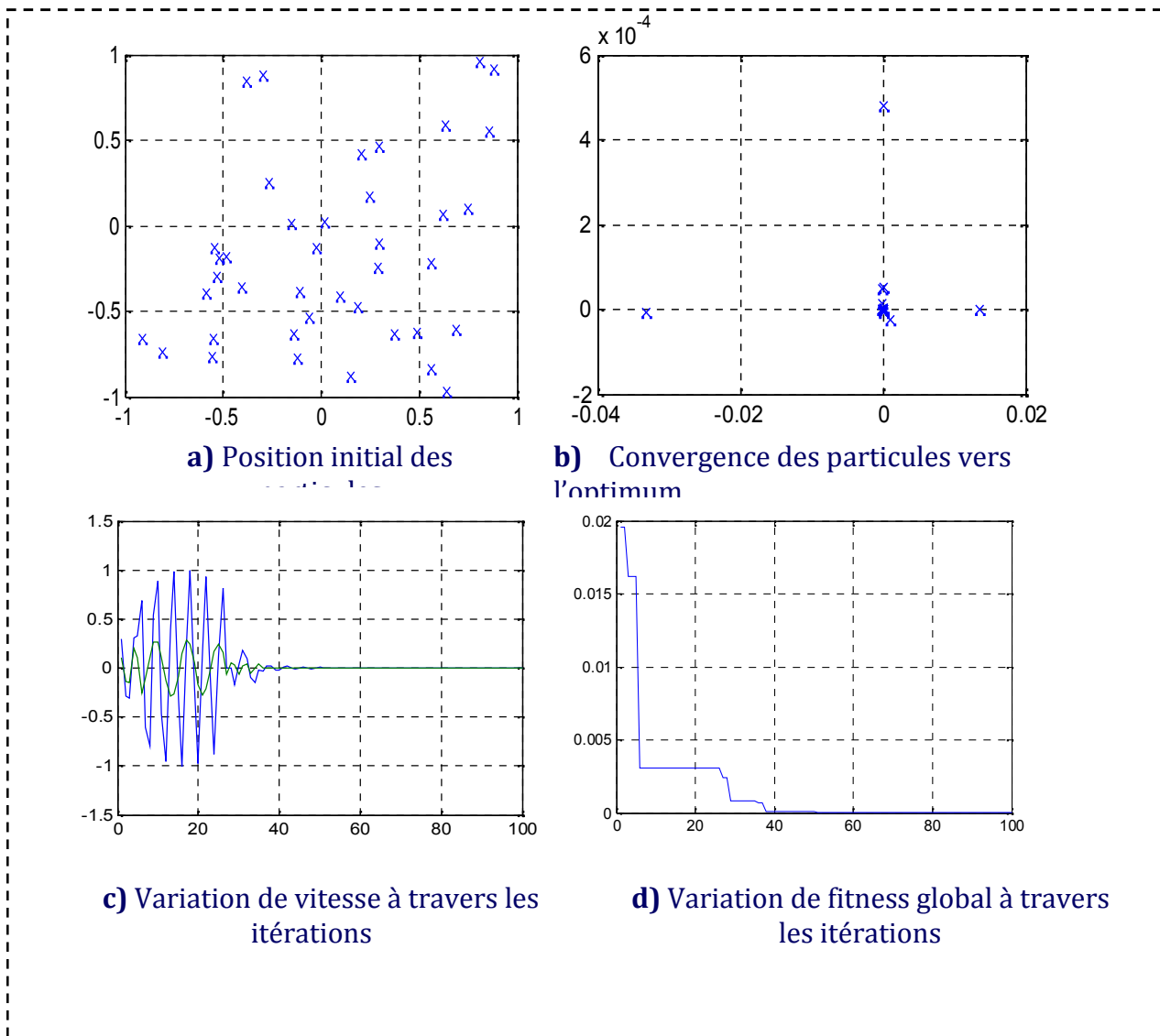
Nombre d'itération = 100.

Nombre de particule = 40.

Facteur d'inertie  $w = 0.9$

Les coefficients d'accélération:  $c1 = 1.5$ ,

$c2=1.9$ .



**Figure 2.18:** Cas où les paramètres de contrôle sont assez réglés.

Avec :

$$\left\{ \begin{array}{l} X= 0.0385, \\ Y= 0.1368, \end{array} \right. \quad \text{Erreur}= 5.2502e-015.$$

## 2.6 Discussion

D'après les figures [*Exemple 1* : (figure 2.11-figure 2.12-figure 2.13 -figure 2.14), *Exemple 2* : (figure 2.15 -figure 2.16 -figure 2.17 -figure 2.18)] et les valeurs de  $x$ ,  $y$  obtenues, on remarque que chaque paramètre a une forte influence sur le comportement de l'algorithme. Alors, le choix des paramètres reste en grande partie empirique, d'ailleurs il faut exécuter le programme plusieurs fois et au fur à mesure on améliore les paramètres pour atteindre l'objectif voulu avec une vitesse de convergence acceptable.

## 2.7 Conclusion

Dans ce chapitre, nous avons présenté une brève introduction sur l'optimisation et les méthodes d'optimisation stochastiques. En général les méthodes déterministes convergent vers l'optimum le plus proche qu'il soit local ou global. à l'opposée, les méthodes stochastiques convergent, grâce à l'intelligence collective, vers l'optimum global. Dans ce chapitre on a essayé de comprendre l'influence des paramètres de réglages (l'inertie, vitesse, population,.....) pour la recherche de la solution désirée d'une part, et d'autre part sur la convergence de l'algorithme. Cette étude va nous servir de base de travail pour l'optimisation de filtre numérique RII.

A decorative border made of a red ribbon, twisted and knotted at the corners, framing the entire page.

# CHAPITRE 3 |

## CHAPITRE III

### Synthèse du filtre numérique RII par la méthode d'optimisation "OEP"

## CHAPITRE 3

### SYNTHESE DE FILTRE NUMERIQUE RII PAR LA METHODE D'OPTIMISATION « OEP »

#### 3.1 Introduction

Dans cette étude on va montrer comment concevoir un filtre numérique de type *IIR* à l'aide d'optimisation par essaim particulaires (OEP). Dans ce cas les paramètres inconnus de filtre numérique *IIR* (les coefficients de filtre) sont considérés comme un vecteur à optimiser par le OEP.

Pour ce la on réalisera un filtre passe bas (LP), un passe-haut (Hp), un passe-bandes (pb) et un stop-Bande (Bs), on utilisant un gabarit idéal pour chaque cas.

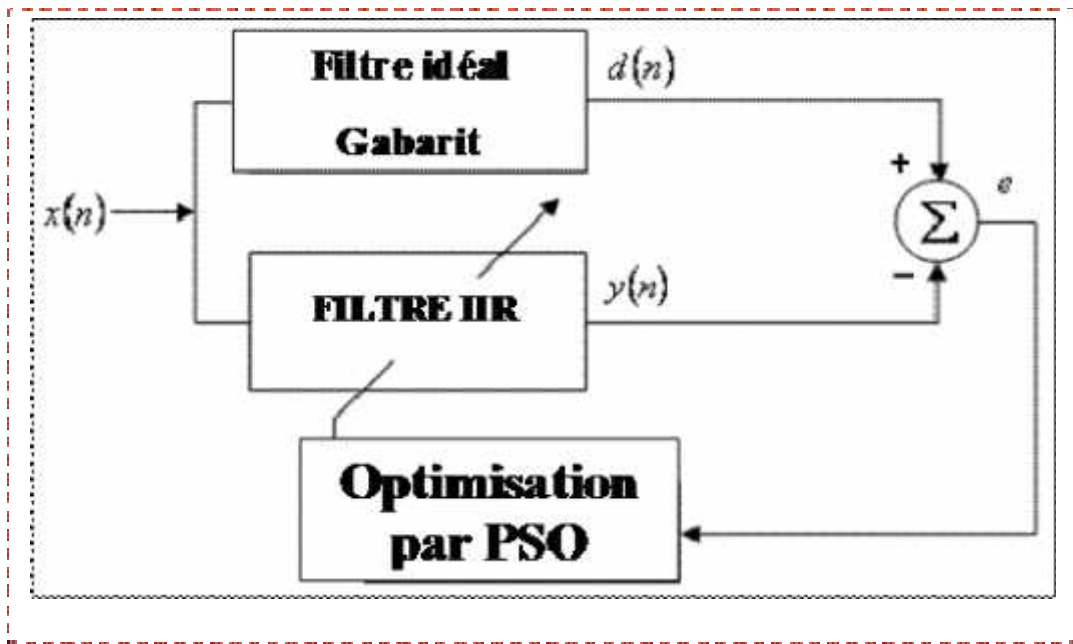
Pour cette raison, on utilisera le langage *MATLAB* pour réaliser la conception des filtres numériques l'utilisation de la méthode d'optimisation OEP.

#### 3.2 La conception de filtre numérique IIR

##### 3.2.1 Description du problème

La conception numérique de filtre IIR basée sur OEP est exécutée comme identification de paramètre comme le montre la figure (3.1). Les paramètres effectifs d'un filtre sont estimés par l'intermédiaire d'un filtre idéal.

Le but de l'optimisation du système est d'ajuster les coefficients du filtre numérique ( $a_i$ ,  $b_i$ ) pour déterminer la fonction de transfert du filtre représentant le filtre idéal approché. En d'autres termes, les paramètres des filtres sont successivement ajustés par du filtre modelant en utilisant les algorithmes d'optimisation jusqu'à l'erreur entre la sortie du filtre et le filtre idéal est réduit au minimum. Le schéma fonctionnel du filtre employant de processus de l'identification IIR de système conçu par l'algorithme de la figure (3.1) [18].



**Figure 3.1** : Schéma d'identification du filtre IIR pour la détermination des coefficients.

Comme critère de minimisation on a utilisé la moyenne de la somme des carrés (MSE) entre la sortie de filtre et la réponse désirée comme suivant : **(la relation 3.1)**

$$MSE = \frac{1}{L} \sum_{n=1}^L (d(n) - y(n))^2 \quad (3.1)$$

$y(n)$  = est La sortie du filtre estimée ;

$d(n)$  = est la sortie du filtre idéal;

$L$  = nombre d'échantillons total.

### 3.2.2 L'application d'OEP aux filtres numérique IIR

La fonction de transfert du filtre numérique IIR représenté par l'équation (3.2):

$$H(Z^{-1}) = \frac{Y_n}{X_n} = \frac{b_0 + b_1 Z^{-1} + b_2 Z^{-2} + \dots + b_m Z^{-m}}{a_0 + a_1 Z^{-1} + a_2 Z^{-2} + \dots + a_n Z^{-n}} \quad (3.2)$$

Le vecteur de coefficients du numérateur et du dénominateur est :

{ $a_0, a_1, a_2, \dots, a_n, b_0, b_1, b_2, \dots, b_m$ }. Ces derniers seront considérer comme les particules de l'OEP à déterminer.

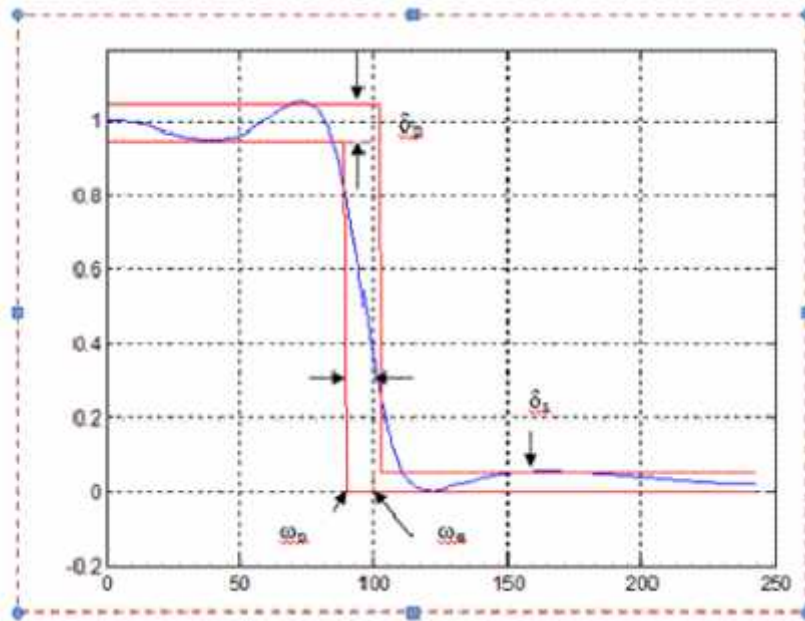
Le but de la fonction fitness a été employé, est de rapprocher efficacement les coefficients de filtre en utilisant les algorithmes, de sorte que la réponse impulsionnelle de filtre soit semblable à celle du filtre idéal [19].



Dans chaque itération, ces particules trouvent une nouvelle position, qui est un nouveau ensemble de coefficients. La fitness des particules est calculée en utilisant les nouveaux coefficients. Cette fitness est employée pour améliorer la recherche dans chaque itération.

Dans l'OEP, la position de chaque particule représente une solution potentielle du problème réel, alors la qualité de la solution dépend de la position des particules.

Les principaux paramètres du gabarit d'un filtre sont représentés dans la figure (3.2) :



**Figure 3.2 :** Les paramètres d'un filtre.

On spécifie les caractéristiques d'un filtre avec un gabarit tel que :

- p: L'ondulation en bande passant.
- a : L'ondulation en bande atténuée.
- p : Dernière fréquence passante.
- a : Première fréquence atténuée.

### 3.2.3 Synthèse de filtre numérique

Pour faire la synthèse de filtre numérique il faut d'abord faire une décomposition de la fonction de transfert en produit (structure cascade) du premier ou du second ordre afin de faciliter la conservation de la stabilité de filtre, dans notre simulation.

Pour notre simulation on va traiter un filtre de 8<sup>ème</sup> ordre et un filtre d'ordre 25 qui ont une fonction de transfert de la forme représentée dans l'équation 3.2.

Où :  $a_k : 1 \leq k < n$  et  $b_k : 0 \leq k < m$ , avec :  $n \leq m$

En prend :

Dans le cas 8<sup>ème</sup> ordre : n = 8, m = 7 et dans le cas de 25<sup>ème</sup> ordre n = 24, m = 23.

Pour le filtre de 8<sup>ème</sup> ordre on décompose le numérateur en trois cellules de 2<sup>ème</sup> ordre et une cellule de 1<sup>er</sup> ordre puis le dénominateur en trois cellules de 2<sup>ème</sup> ordre et deux cellules de 1<sup>er</sup> ordre et pour le filtre de 25<sup>ème</sup> ordre on décompose le numérateur en neuf cellules de 2<sup>ème</sup> ordre et 16cellules de 1<sup>er</sup> ordre puis le dénominateur en neuf cellules de 2<sup>ème</sup> ordre et 14 cellules de 1<sup>er</sup> ordre.

Pour garder la stabilité de ce filtre il faut qu'on suive les conditions de stabilité que sont illustrés dans les équations (1.16) et (1.17).

On déduit la forme suivante :

$$H(Z^{-1}) = \frac{(b_1 Z^2 + b_2 Z^1 + 1)(b_3 Z^2 + b_4 Z^1 + 1)(b_5 Z^2 + b_6 Z^1 + 1) \dots (b_m Z^1 + 1)}{(a_1 Z^2 + a_2 Z^1 + 1)(a_3 Z^2 + a_4 Z^1 + 1)(a_7 Z^1 + 1) \dots (a_n Z^1 + 1)} \quad (3.3)$$

Dans cette étude on va configurer les paramètres du filtre et d'OEP comme suit :

#### ❖ Paramètres reliés au filtre

**Les fréquences :**

#### **Fréquence d'échantillonnage**

**Dans le premier cas :**

fe=20000; % la fréquence d'échantillonnage en 'Hz'.

#### **Fréquences de coupure**

fc1=4000; % la fréquence de coupure en 'Hz'.

fc2=5500; % la fréquence de coupure en 'Hz'.

**Dans le deuxième cas :**

fe =200000; % la fréquence d'échantillonnage en 'Hz'.

#### **Fréquences de coupure**

fc1=40000; % la fréquence de coupure en 'Hz'.

```
fc2=75000; % la fréquence de coupure en 'Hz'.
```

**L'ordre du filtre**

```
N=8; %nombre du filtre
```

### ❖ Les paramètres reliés à l'algorithme OEP

**Nombre de particules**

```
swarm_size = 50; % nombre de particule
```

**Nombre d'itération max**

```
iter_max = 1000; % nombre iteration
```

**parametre d'inertie**

```
W0=0.96; % parametre d'inertie
```

**Les coefficients de correction :**

```
C1 = 1.5; % correction locale
```

```
C2 = 1.7; % correction globale
```

Voir l'organigramme pour la conception de filtre par la méthode OEP :

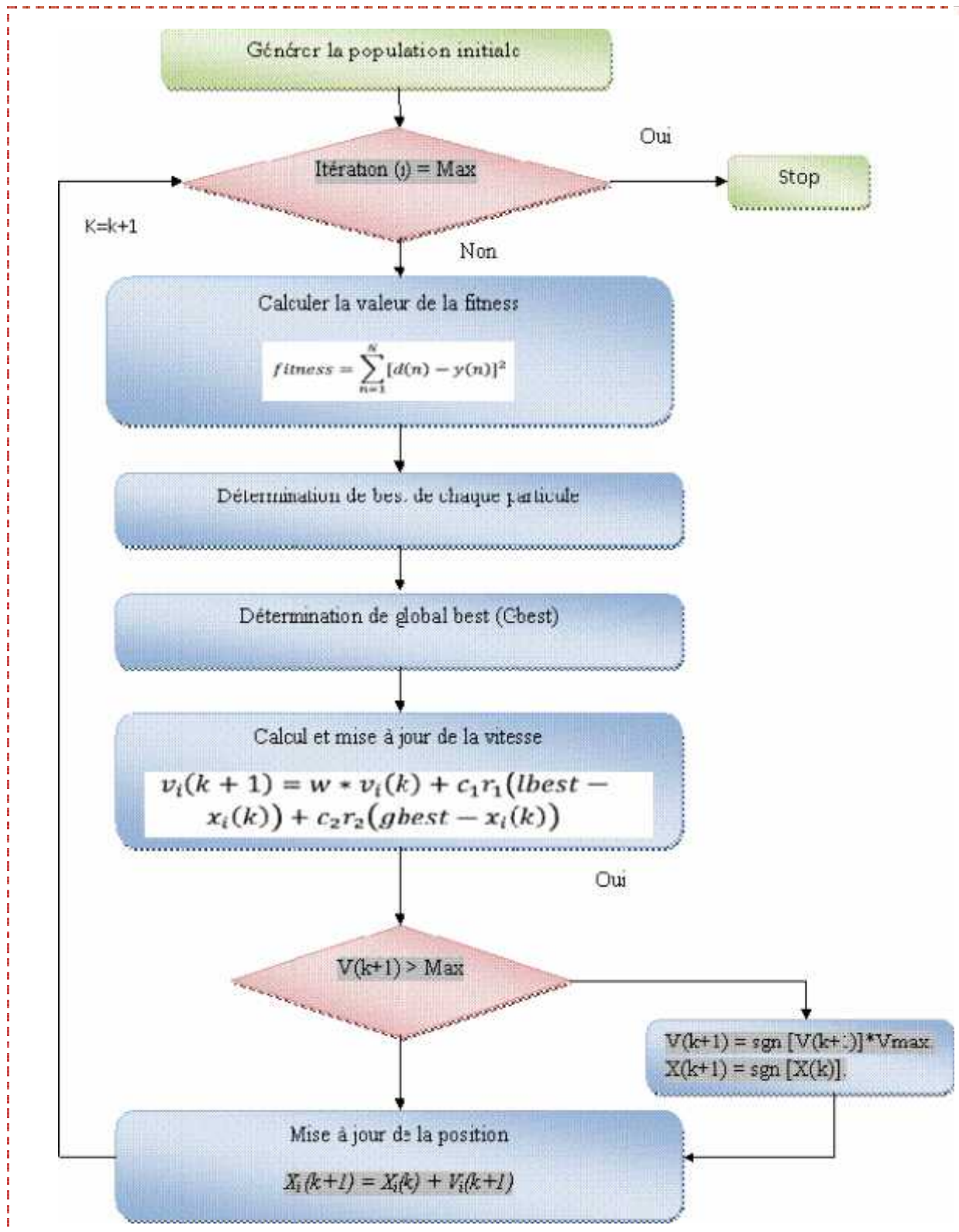
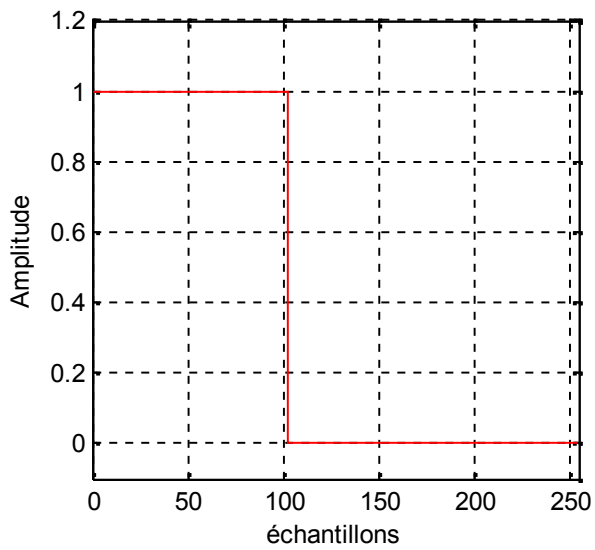


Figure 3.3 : Organigramme d'OEP modifié.

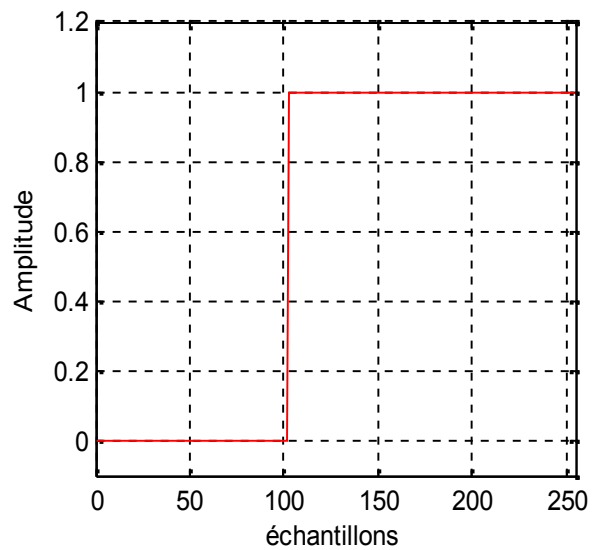
Les gabarits idéaux des quatre types des filtres numériques sont les suivants :

**Gabarit de filtre passe-bas**



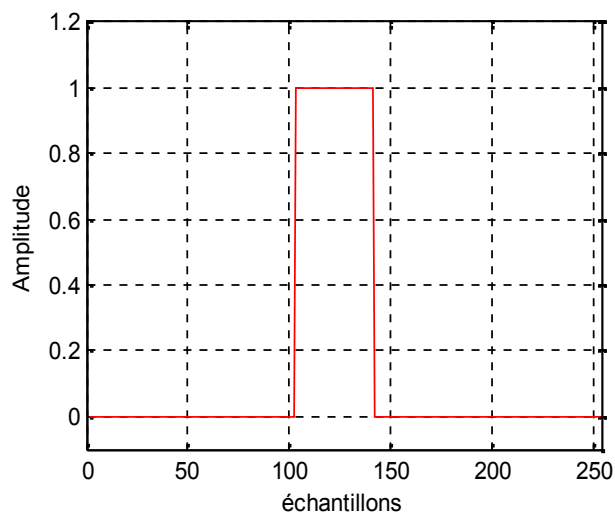
**Figure 3.4 : Gabarit de filtre passe-bas.**

**Gabarit de filtre passe-haut**



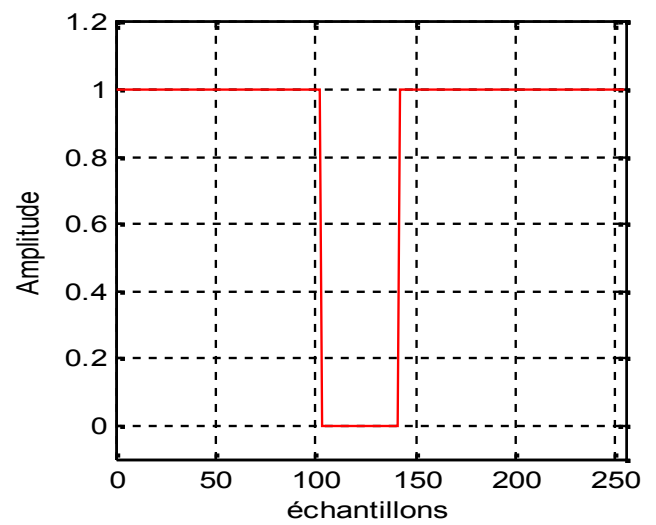
**Figure 3.5 : Gabarit de filtre passe-haut.**

**Gabarit de filtre passe-bande**



**Figure 3.6 : Gabarit de filtre passe-bande.**

**Gabarit de filtre stop-bande**



**Figure 3.7 : Gabarit de filtre top-bande.**

### 3.3 RESULTATS ET INTERPRETATIONS

#### 3.3.1 Simulation n°=1 :

Premièrement on réalisera les quatre types de filtre (passe-bas, passe-haut, passe-bande, stop-bande) d'ordre 8 et après d'ordre 25 et on constatera l'influence d'ordre sur les résultats.

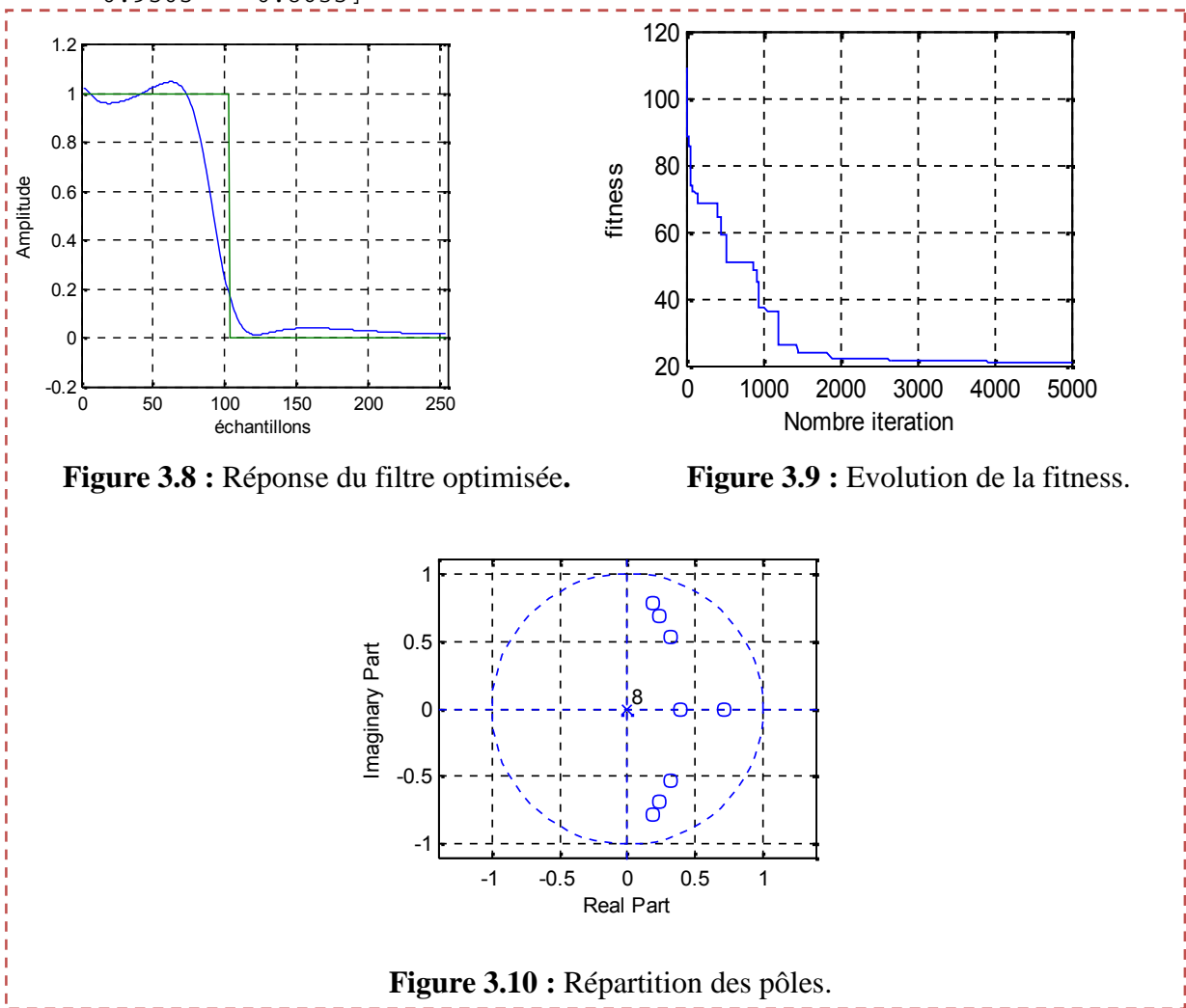
##### ❖ Le cas d'un filtre d'ordre N=8

##### ➤ Cas de filtre passe-bas

🚦 Les coefficients de filtre sont :

$$\checkmark [a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6 \ a_7 \ a_8] = [-0.6252 \quad 0.3748 \quad -0.3593 \quad 0.6407 \quad -0.4722 \quad 0.5278 \quad -0.3866 \quad -0.7056]$$

$$\checkmark [b_1 \ b_2 \ b_3 \ b_4 \ b_5 \ b_6 \ b_7] = [-0.0291 \quad 0.9709 \quad -0.8604 \quad 0.1396 \quad -0.0495 \quad 0.9505 \quad 0.8635]$$



🚦 **Fitness = 21.0808.**

D'après la figure 3.8 on remarque qu'en utilisant un filtre d'ordre 8 on arrive à l'optimiser le gabarit exigé néanmoins avec une convergence de l'algorithme au bout de la 1500<sup>ème</sup> itération, voir figure 3.9.

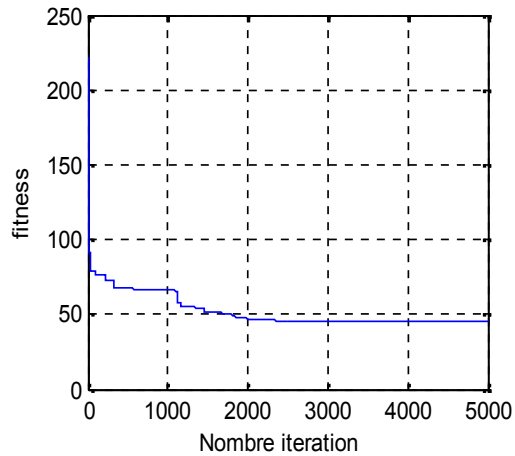
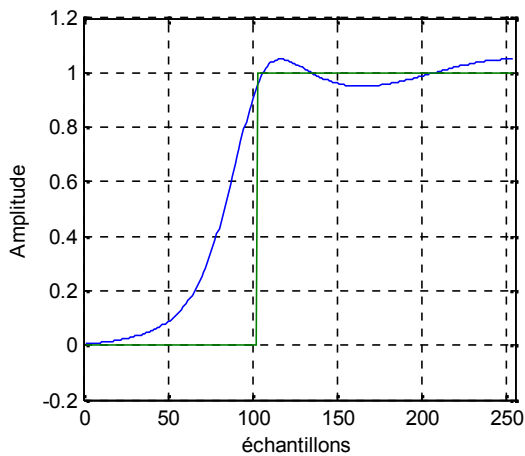
La figure 3.10 montre que les coefficients trouvés imposent un filtre IIR stable car les pôles sont à l'intérieur de cercle unité.

➤ **Cas de filtre passe-haut**

🚩 **Les coefficients de filtre sont :**

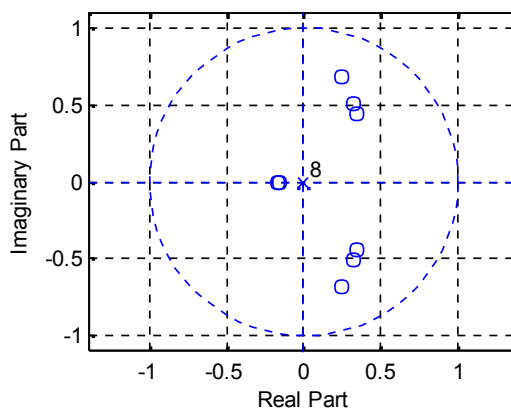
✓  $[a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6 \ a_7 \ a_8] = [-0.6863 \ 0.3137 \ -0.6449 \ 0.3551 \ -0.4850 \ 0.5150 \ 0.1637 \ 0.1530]$

✓  $[b_1 \ b_2 \ b_3 \ b_4 \ b_5 \ b_6 \ b_7] = [-0.7568 \ 0.2432 \ -0.7114 \ 0.2886 \ -0.7944 \ 0.2056 \ -0.9347]$



**Figure 3.11 :** Réponse du filtre optimisée.

**Figure 3.12 :** Evolution de la fitness.



**Figure 3.13 :** Répartition des pôles.

🚩 **Fitness = 45.1927.**

D'après la figure 3.11 on remarque aussi qu'en utilisant un filtre d'ordre 8 on arrive à une fonction de transfert approchée au gabarit exigé avec une convergence de l'algorithme au bout de la 2400<sup>ème</sup> itération, voir figure 3.12.

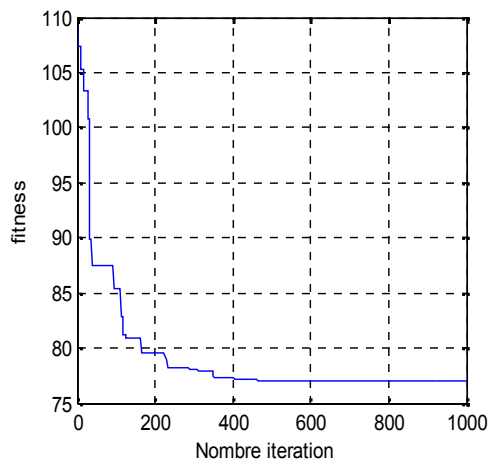
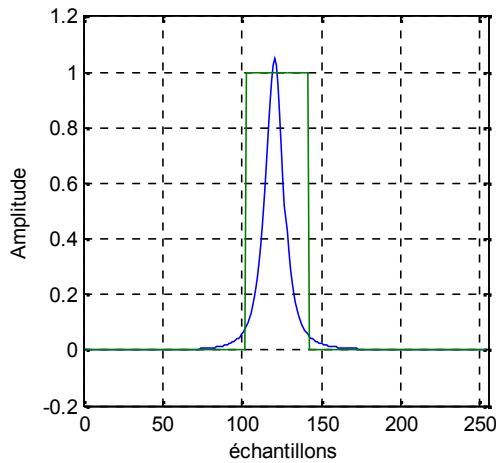
La figure 3.13 montre que les coefficients trouvés imposent un filtre IIR stable car les pôles sont à l'intérieur de cercle unité.

➤ **Cas de filtre passe-bande**

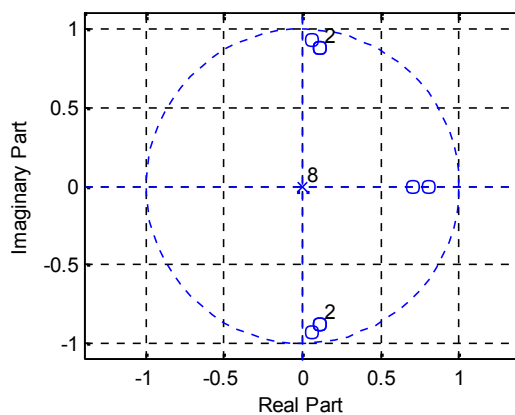
🚦 **Les coefficients de filtre sont :**

$$\checkmark [a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6 \ a_7 \ a_8] = [-0.1980 \ 0.8020 \ -0.1078 \ 0.8922 \ -0.2066 \ 0.7934 \ -0.4966 \ 0.5034]$$

$$\checkmark [b_1 \ b_2 \ b_3 \ b_4 \ b_5 \ b_6 \ b_7] = [-0.6920 \ 0.3080 \ -0.8044 \ 0.1956 \ -0.8934 \ 0.1066 \ -0.4085]$$



**Figure 3.14 :** Réponse du filtre optimisée. **Figure 3.15 :** Evolution de la fitness.



**Figure 3.16 :** Répartition des pôles.

🚦 **Fitness=77.5438.**



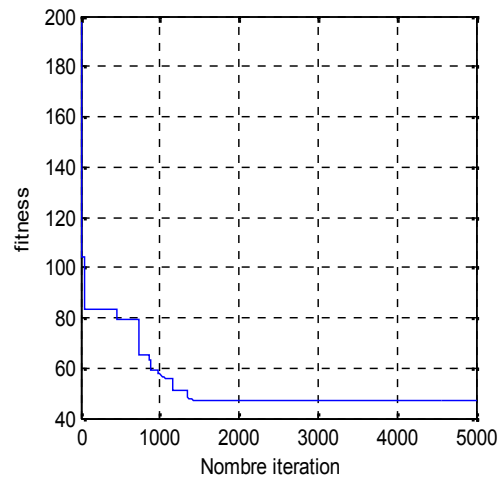
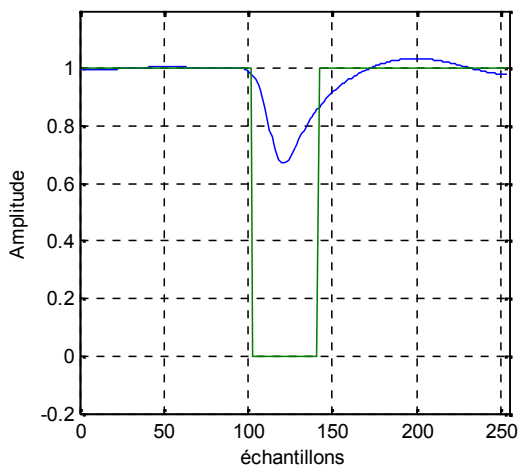
La figure 3.14 montre que la réponse en fréquence de filtre RII n'a pas respecter de manière convenable le gabarit exigé malgré la convergence de l'algorithme, voir figure 3.15.

➤ **Cas de filtre stop-bande**

📌 **Les coefficients de filtre sont :**

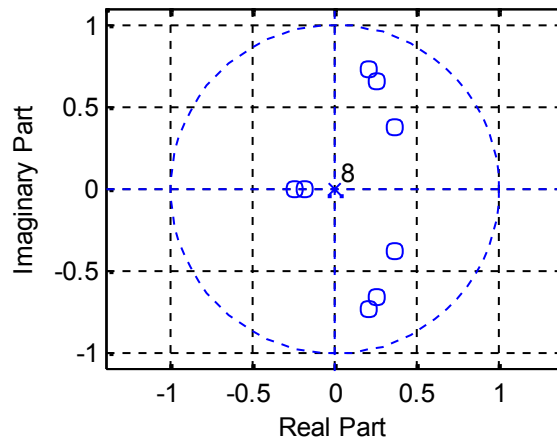
$$\checkmark [a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6 \ a_7 \ a_8] = [-0.7282 \ 0.2718 \ -0.5026 \ 0.4974 \ -0.4194 \ 0.5806 \ 0.2474 \ 0.1802]$$

$$\checkmark [b_1 \ b_2 \ b_3 \ b_4 \ b_5 \ b_6 \ b_7] = [-0.5355 \ 0.4645 \ -0.3573 \ 0.6427 \ -0.7149 \ 0.2851 \ 0.3820]$$



**Figure 3.17 :** Réponse du filtre optimisée.

**Figure 3.18 :** Evolution de la fitness.



**Figure 3.19 :** Répartition des pôles.

📌 **Fitness = 46.9159.**

La figure 3.17 montre que la réponse en fréquence de filtre RII est mauvaise par rapport au celle de gabarit exigé malgré la convergence de l'algorithme, voir figure (3.18).

D'après les résultats précédents on constate qu'on peut réaliser des filtres passe-bas et passe-haut d'ordre 8, mais s'avère insuffisant pour réaliser des filtres stop-bande et passe-bande, ce qui impose un changement de la fonction fitness ou une augmentation de l'ordre.

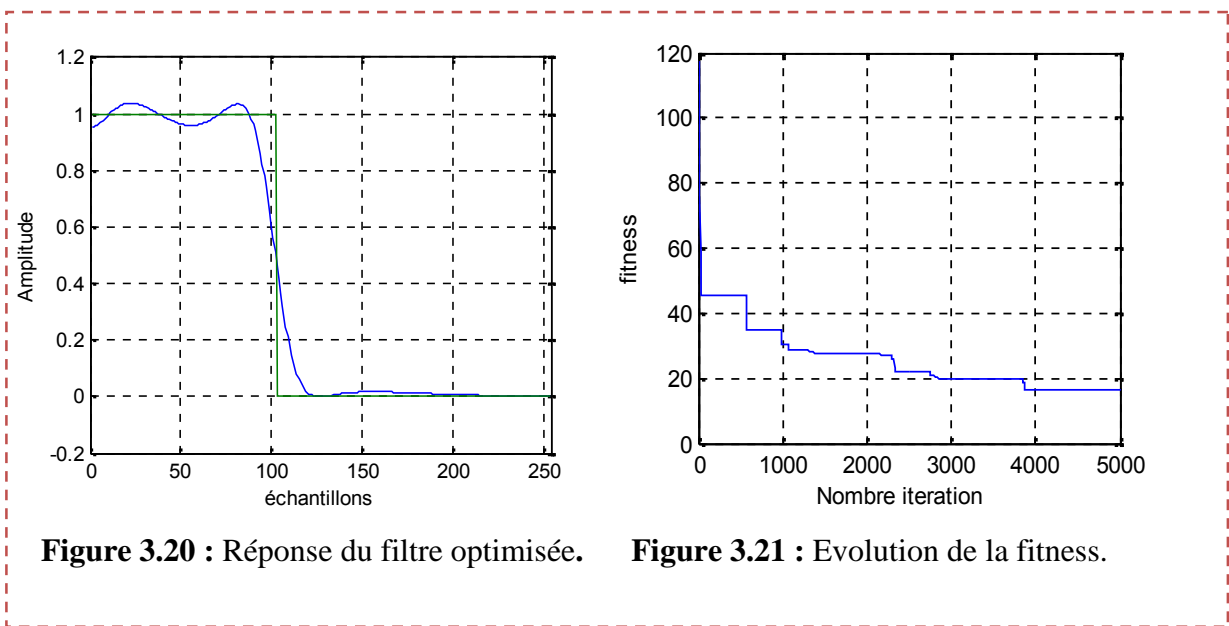
❖ **Le cas d'un filtre d'ordre N=25**

➤ **Cas de filtre passe-bas**

🚩 **Les coefficients de filtre sont :**

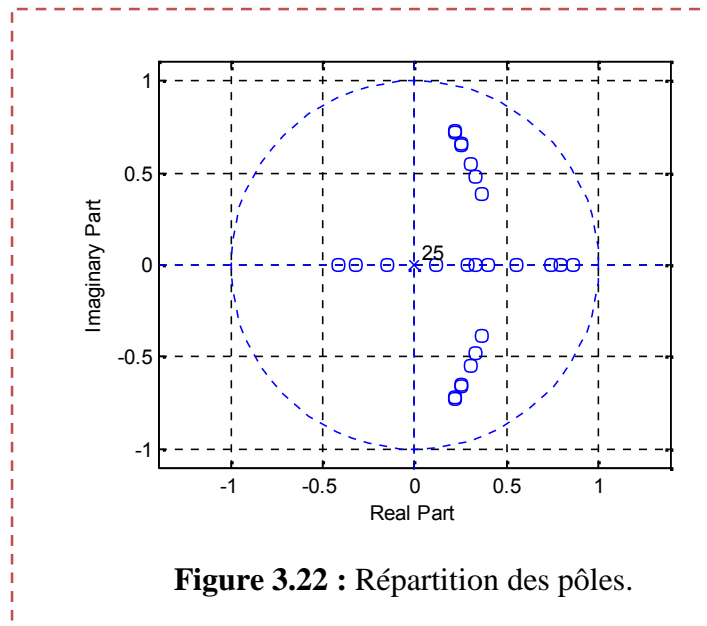
✓ [a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 a15 a16 a17  
a18 a19 a20 a21 a22 a23 a24]= [ -0.4404 0.5596 -0.4986  
0.5014 -0.4295 0.5705 -0.6633 0.3367 -0.6093  
0.3907 -0.5117 0.4883 -0.7228 0.2772 -0.9090  
0.0910 -0.8410 0.1590 -0.4171 -0.2402 -0.4501 -  
0.3568 -0.3969 -0.3267]

✓ [b1 b2 b3 b4 b5 b6 b7 b8 b9 b10 b11 b12 b13 b14 b15 b16 b17  
b18 b19 b20 b21 b22 b23 ]=[-0.9386 0.0614 -0.0076  
0.9924 -0.6746 0.3254 -0.0631 0.9369 -0.8976  
0.1024 -0.6870 0.3130 -0.6581 0.3419 -0.5437  
0.4563 -0.9181 0.0819 0.1304 0.7536 -0.4171  
0.5278 0.6277]



**Figure 3.20 :** Réponse du filtre optimisée.

**Figure 3.21 :** Evolution de la fitness.



**Figure 3.22 : Répartition des pôles.**

 **Fitness=16.2688**

D'après la figure 3.20 on remarque qu'en utilisant un filtre d'ordre 25 on arrive à optimiser le gabarit exigé néanmoins avec une convergence de l'algorithme au bout de la 3800<sup>ème</sup> itération, voir figure 3.21.

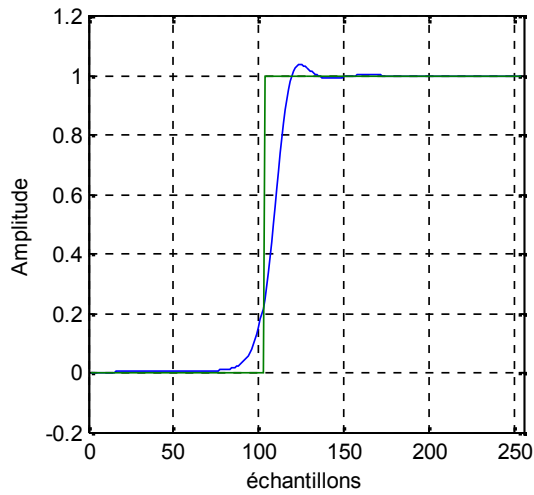
La figure 3.22 montre que les coefficients trouvés du filtre IIR sont acceptables car les pôles sont à l'intérieur du cercle unité.

➤ **Cas de filtre passe-haut**

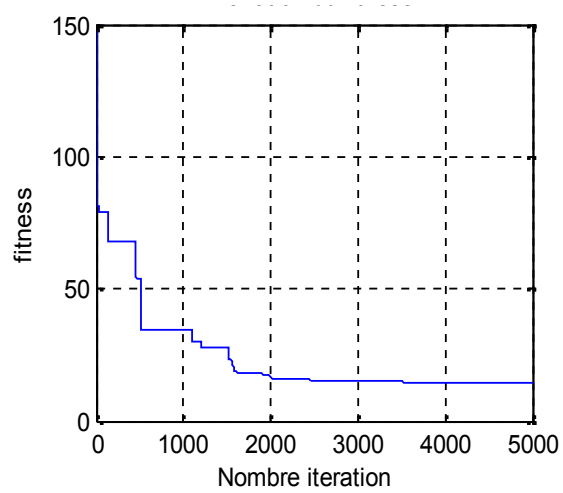
 **Les coefficients de filtre sont :**

✓ [a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 a15 a16 a17  
a18 a19 a20 a21 a22 a23 a24]=[ -0.3577 0.6423 -0.9052  
0.0948 -0.8153 0.1847 -0.8318 0.1682 -0.7369  
0.2631 -0.3394 0.6606 -0.7816 0.2184 -0.3730  
0.6270 -0.3585 0.6415 0.5847 0.1557 0.3089  
0.4124 0.1943 -0.0125]

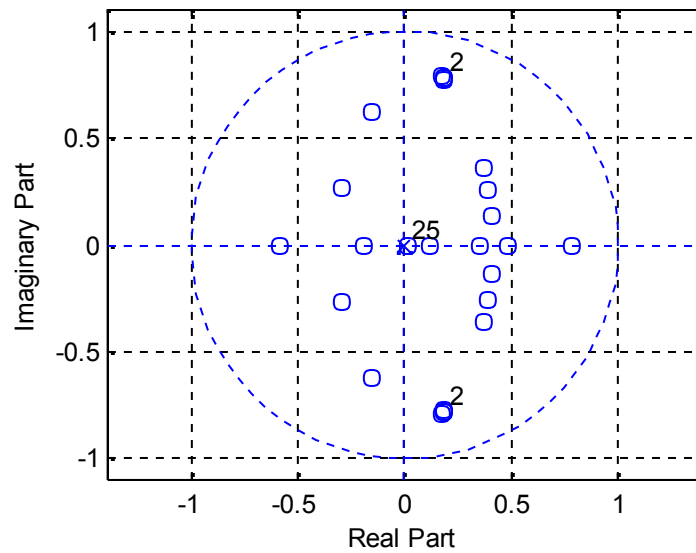
✓ [b1 b2 b3 b4 b5 b6 b7 b8 b9 b10 b11 b12 b13 b14 b15 b16 b17  
b18 b19 b20 b21 b22 b23 ]=[ -0.5553 0.4447 -0.9444  
0.0556 -0.5848 0.4152 -0.6234 0.3766 -0.6384 0.3616  
-0.6529 0.3471 -0.6002 0.3998 -0.6794 0.3206 -0.5917  
0.4083 -0.1547 -0.1325 0.5847 -0.4455 -0.9673].




**Figure 3.23 :** Réponse du filtre optimisée.



**Figure 3.24 :** Evolution de la fitness.



**Figure 3.25 :** Répartition des pôles.

 **Fitness = 14.2177.**

D'après la figure 3.23 on remarque qu'en utilisant un filtre d'ordre 8 on arrive à l'optimiser le gabarit exigé néanmoins avec une convergence de l'algorithme au bout de la 2000<sup>ème</sup> itération, voir figure 3.24.

La figure 3.25 montre que le filtre IIR est stable car les pôles sont à l'intérieur du cercle unité.

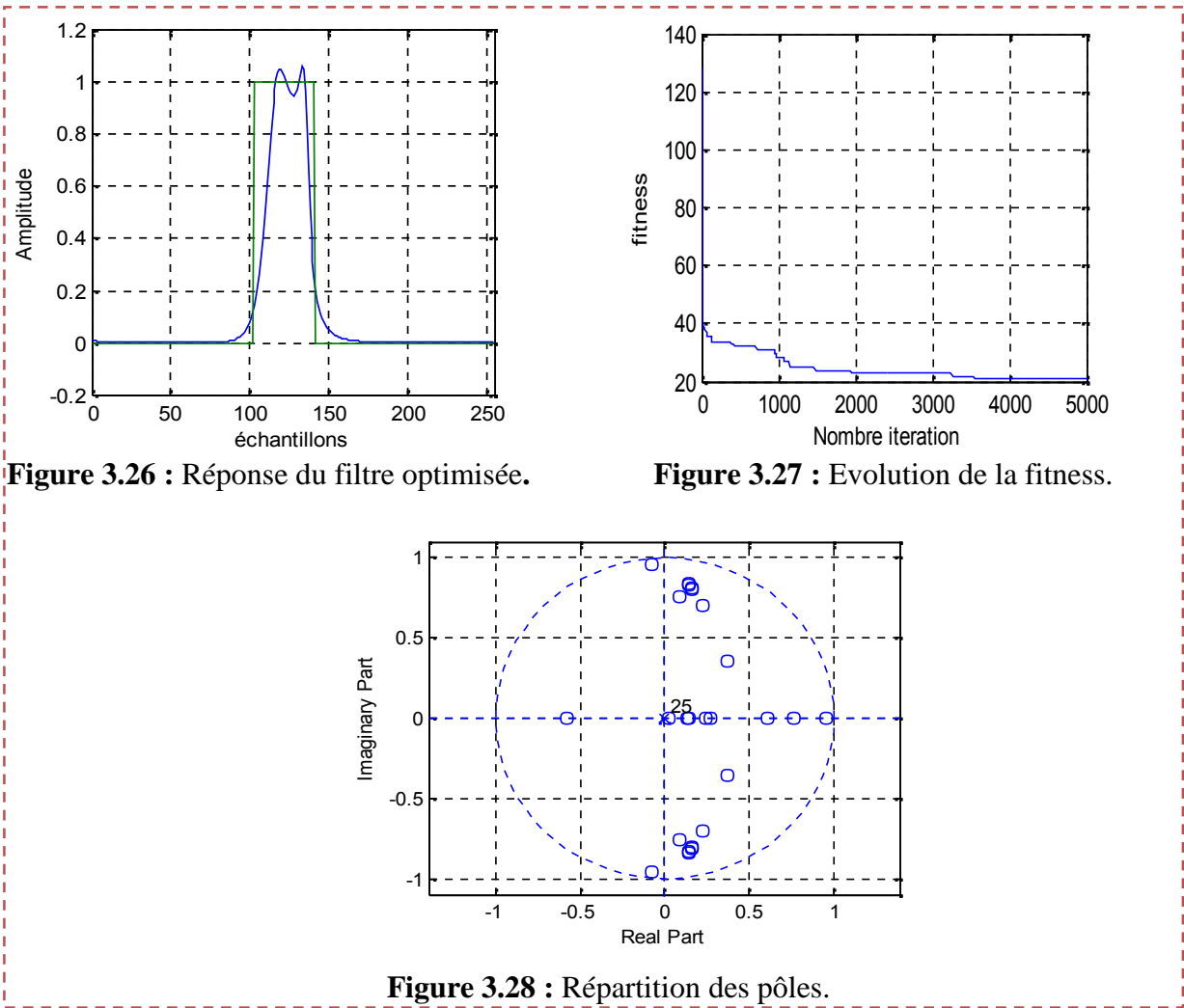
➤ **Cas de filtre passe-bande**

🚩 **Les coefficients de filtre sont :**

✓ [a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 a15 a16 a17 a18 a19 a20 a21 a22 a23 a24]=[ -0.4568 0.5432

✓ -0.9764 0.0236 -0.8509 0.1491 -0.3223 0.6777 -0.3209  
 0.6791 -0.7390 0.2610 -0.8967 0.1033 -0.2796  
 0.7204 -0.2927 0.7073 0.1631 0.9201 -0.1717  
 0.5775 -0.1397 -0.2720]

✓ [b1 b2 b3 b4 b5 b6 b7 b8 b9 b10 b11 b12 b13 b14 b15 b16 b17 b18 b19 b20 b21 b22 b23 ]=[-0.6361 0.3639 -0.5369  
 0.4631 -0.8789 0.1211 -0.7062 0.2938 -0.5481  
 0.4519 -0.6211 0.3789 -0.8702 0.1298 -0.7544  
 0.2456 -0.6179 0.3821 0.5744 0.0785 0.1631  
 0.4622 0.5583]



**Figure 3.26 :** Réponse du filtre optimisée.

**Figure 3.27 :** Evolution de la fitness.

**Figure 3.28 :** Répartition des pôles.

🚩 **Fitness = 20.7893.**

D'après la figure 3.26 on remarque que la réponse impulsionnelle du filtre numérique est approchée vers le gabarit exigé et dans la figure 3.27 on constate une convergence de l'algorithme au bout de la 3500<sup>ième</sup> itération.

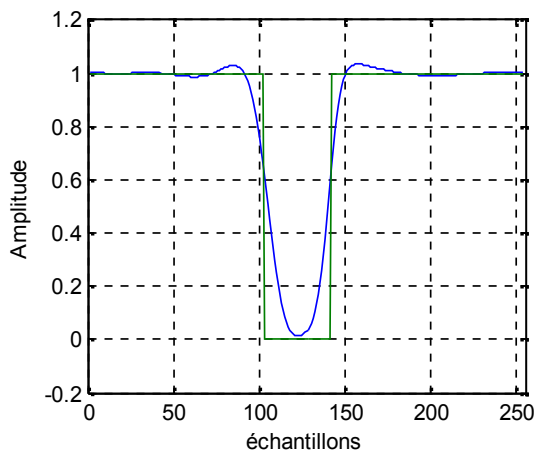
La figure 3.28 montre que le filtre IIR stable car les pôles sont à l'intérieur du cercle unité.

➤ **Cas de filtre stop-bande**

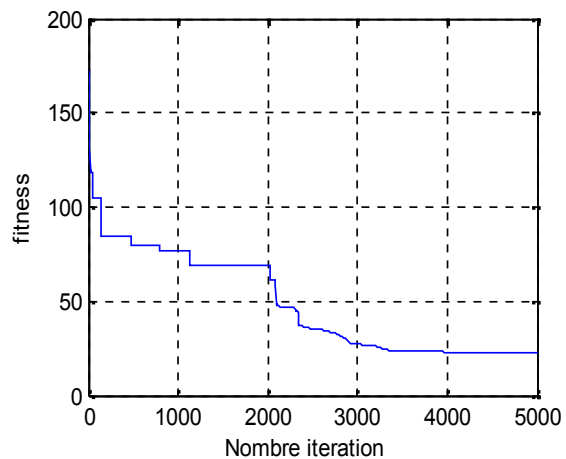
🚩 **Les coefficients de filtre sont :**

✓ [a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 a15 a16 a17  
a18 a19 a20 a21 a22 a23 a24]=[ -0.7654 0.2346 -0.7771  
0.2229 -0.5093 0.4907 -0.3574 0.6426 -0.4626  
0.5374 -0.7717 0.2283 -0.9166 0.0834 -0.4767  
0.5233 -0.4802 0.5198 0.3139 0.7827 -0.2058  
0.4050 0.3558 0.2088]

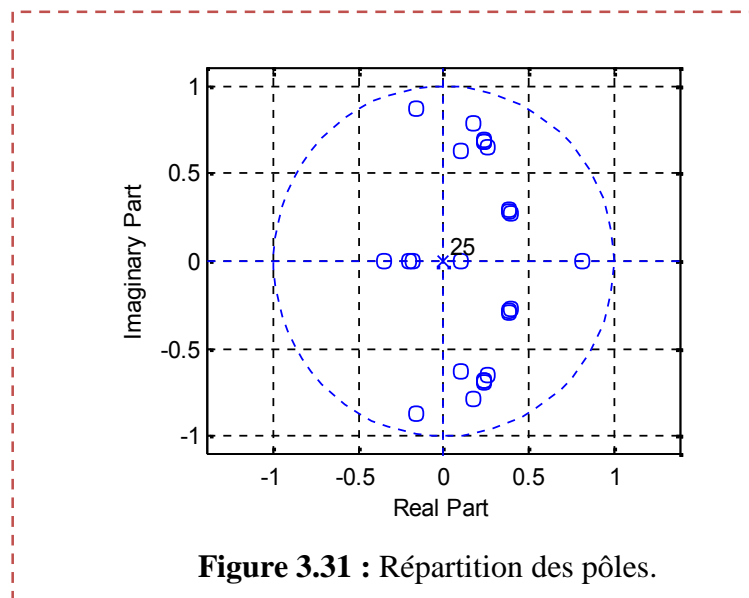
✓ [b1 b2 b3 b4 b5 b6 b7 b8 b9 b10 b11 b12 b13 b14 b15 b16 b17  
b18 b19 b20 b21 b22 b23 ]=[ ]=[ -0.7894 0.2106 -0.6263  
0.3737 -0.1017 0.8983 -0.7426 0.2574 -0.5413  
0.4587 -0.1197 0.8803 -0.4609 0.5391 -0.5189  
0.4811 -0.1239 0.8761 0.1781 -0.8100 0.3139 -0.3032  
-0.0676]



**Figure 3.29 :** Réponse du filtre optimisée.



**Figure 3.30 :** Evolution de la fitness.



**Fitness =22.9790.**

- ✓ La convergence de la fonction fitness induit que l’algorithme d’optimisation a pu suivre les gabarits exigés (voir la figure 3.29).
- ✓ Les pôles sont à l’intérieur du cercle unité dans les deux cas d’ordre 8 et 25, donc les filtres sont stables (voir la figure 3.31).
- ✓ D’après les résultats précédents on constate que l’ordre 25 est suffisant pour réaliser tous les types des filtres exigés.
- ✓ Le filtre d’ordre 25 a donné des résultats meilleurs que le filtre d’ordre 8.

**3.3.2 Simulation n°=2** (cas d’un filtre d’ordre N=25 avec changement de la bande passante et la fréquence d’échantillonnage).

On va réaliser les quatre types de filtre (passe-bas, passe-haut, passe-bande, stop-bande) d’ordre 25 et on constate l’influence de la bande de fréquence sur les résultats.

**3) Bande de fréquence large :**

- ❖  $F_e=200000$  Hz.
- ❖  $F_{c1}=40000$  Hz.
- ❖  $F_{c2}=75000$  Hz.

➤ Cas de filtre passe-bas

✚ Les coefficients de filtre sont :

✓ [a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 a15 a16 a17  
a18 a19 a20 a21 a22 a23 a24]=[ -0.8265 0.1735 -0.9568  
0.0432 -0.5625 0.4375 -0.4618 0.5382 -0.7153  
0.2847 -0.7333 0.2667 -0.5940 0.4060 -0.7468  
0.2532 -0.5065 0.4935 0.2008 0.3113 -0.6049  
0.8699 -0.1709 -0.6851]

✓ [b1 b2 b3 b4 b5 b6 b7 b8 b9 b10 b11 b12 b13 b14 b15 b16 b17  
b18 b19 b20 b21 b22 b23]=[ -0.9563 0.0437 -0.6642  
0.3358 -0.4824 0.5176 -0.5700 0.4300 -0.1184  
0.8816 -0.7330 0.2670 -0.8437 0.1563 -0.8900  
0.1100 -0.1596 0.8404 0.3718 -0.2390 0.2008 -0.0227  
0.2243].

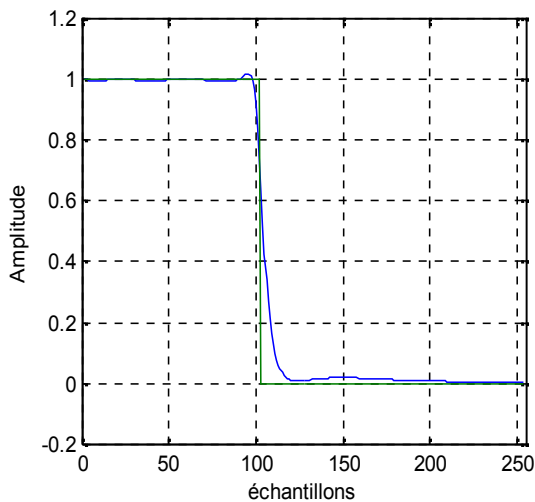


Figure 3.32 : Réponse du filtre optimisée.

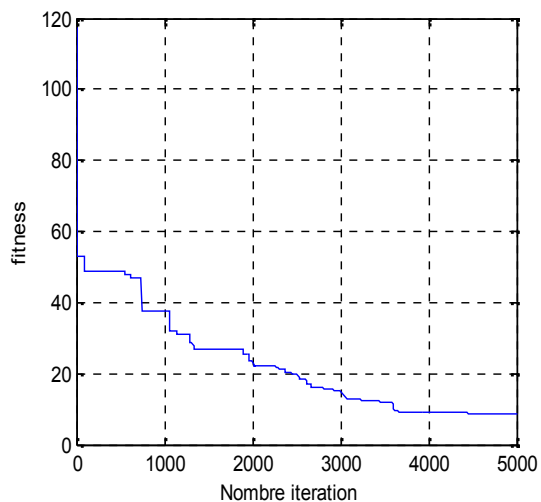
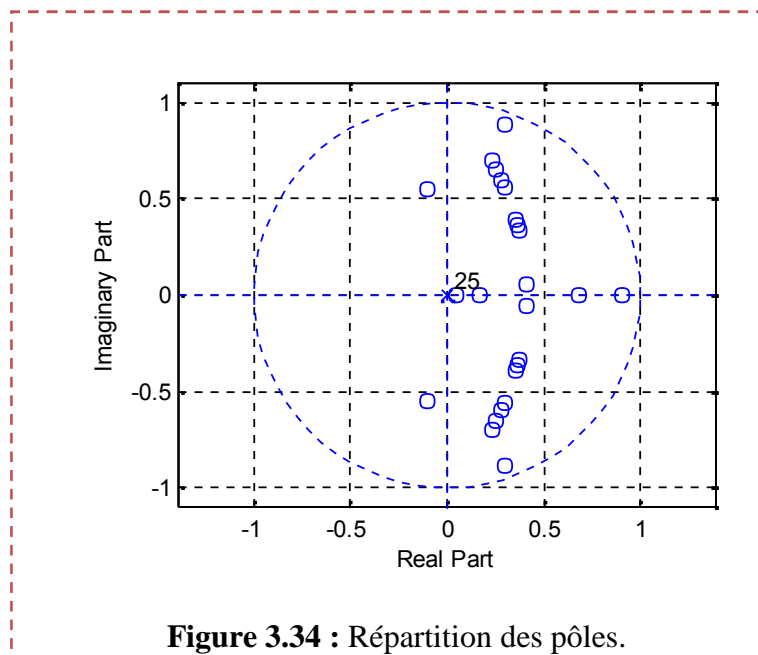


Figure 3.33 : Evolution de la fitness.





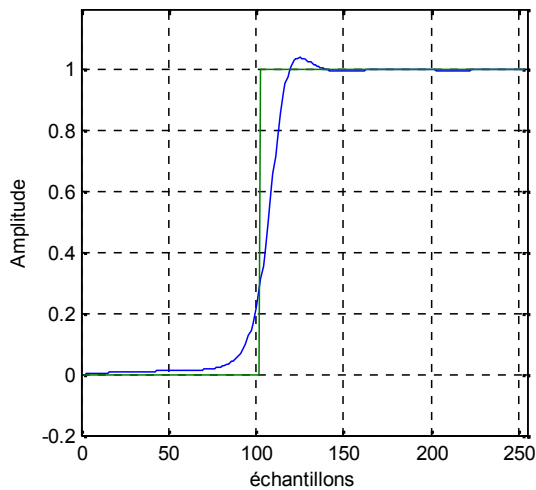
🚩 **Fitness = 8.8441.**

➤ **Cas de filtre passe-haut**

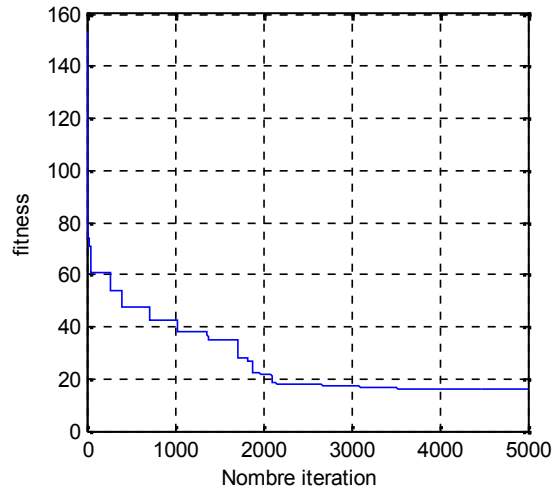
🚩 **Les coefficients de filtre sont :**

✓ [a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 a15 a16 a17  
a18 a19 a20 a21 a22 a23 a24]= [ -0.3492 0.6508 -0.9128  
0.0872 -0.7946 0.2054 -0.3524 0.6476 -0.8051  
0.1949 -0.8210 0.1790 -0.3556 0.6444 -0.6733  
0.3267 -0.7708 0.2292 -0.1102 0.0743 0.1900  
0.2894 -0.1260 -0.1141]

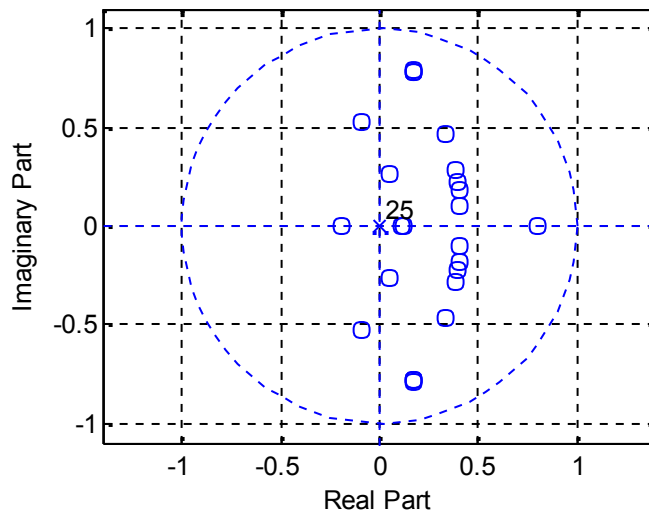
✓ [b1 b2 b3 b4 b5 b6 b7 b8 b9 b10 b11 b12 b13 b14 b15 b16 b17  
b18 b19 b20 b21 b22 b23 ]=[ -0.8108 0.1892 -0.6102  
0.3898 -0.7397 0.2603 -0.6404 0.3596 -0.5281  
0.4719 -0.6262 0.3738 -0.6740 0.3260 -0.7612  
0.2388 -0.5553 0.4447 -0.7527 -0.9984 -0.1102  
-0.3984 0.3856]



**Figure 3.35:** Réponse du filtre optimisée.



**Figure 3.36 :** Evolution de la fitness.



**Figure 3.37 :** Répartition des pôles.

 **Fitness =16.1970**

D'après la figure 3.35 on remarque qu'on utilisant un filtre d'ordre 25 on arrive à l'optimiser le gabarit exigé néanmoins avec une convergence de l'algorithme au bout de la 3500<sup>ème</sup> itération, voir figure 3.36.

La figure 3.37 montre que les coefficients trouvés imposent un filtre IIR stable car les pôles sont à l'intérieur du cercle unité.

➤ Cas de filtre passe-bande

✚ Les coefficients de filtre sont :

✓ [a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 a15 a16 a17  
a18 a19 a20 a21 a22 a23 a24]=[ -0.5747 0.4253 -0.8988  
0.1012 -0.8005 0.1995 -0.9917 0.0083 -0.3388  
0.6612 -0.4460 0.5540 -0.3879 0.6121 -0.8101  
0.1899 -0.8590 0.1410 0.9217 0.5942 0.5818  
0.5016 -0.6133 0.2911]

✓ [b1 b2 b3 b4 b5 b6 b7 b8 b9 b10 b11 b12 b13 b14 b15 b16 b17  
b18 b19 b20 b21 b22 b23 ]=[ -0.6452 0.3548 -0.9330  
0.0670 -0.8679 0.1321 -0.8095 0.1905 -0.9743  
0.0257 -0.7566 0.2434 -0.8044 0.1956 -0.6747  
0.3253 -0.6342 0.3658 -0.4146 0.9540 0.9217 -0.6582  
-0.3529]

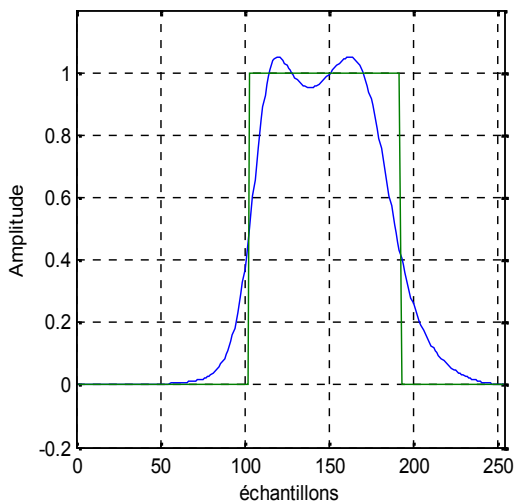


Figure 3.38 : Réponse du filtre optimisée.

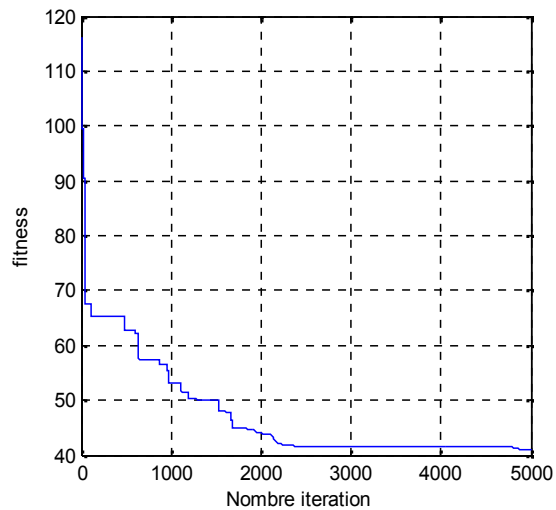
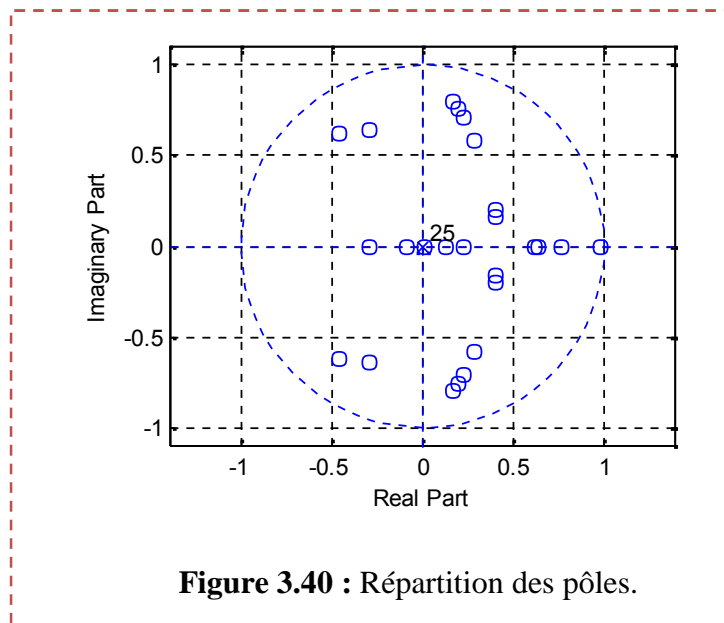


Figure 3.39 : Evolution de la fitness.



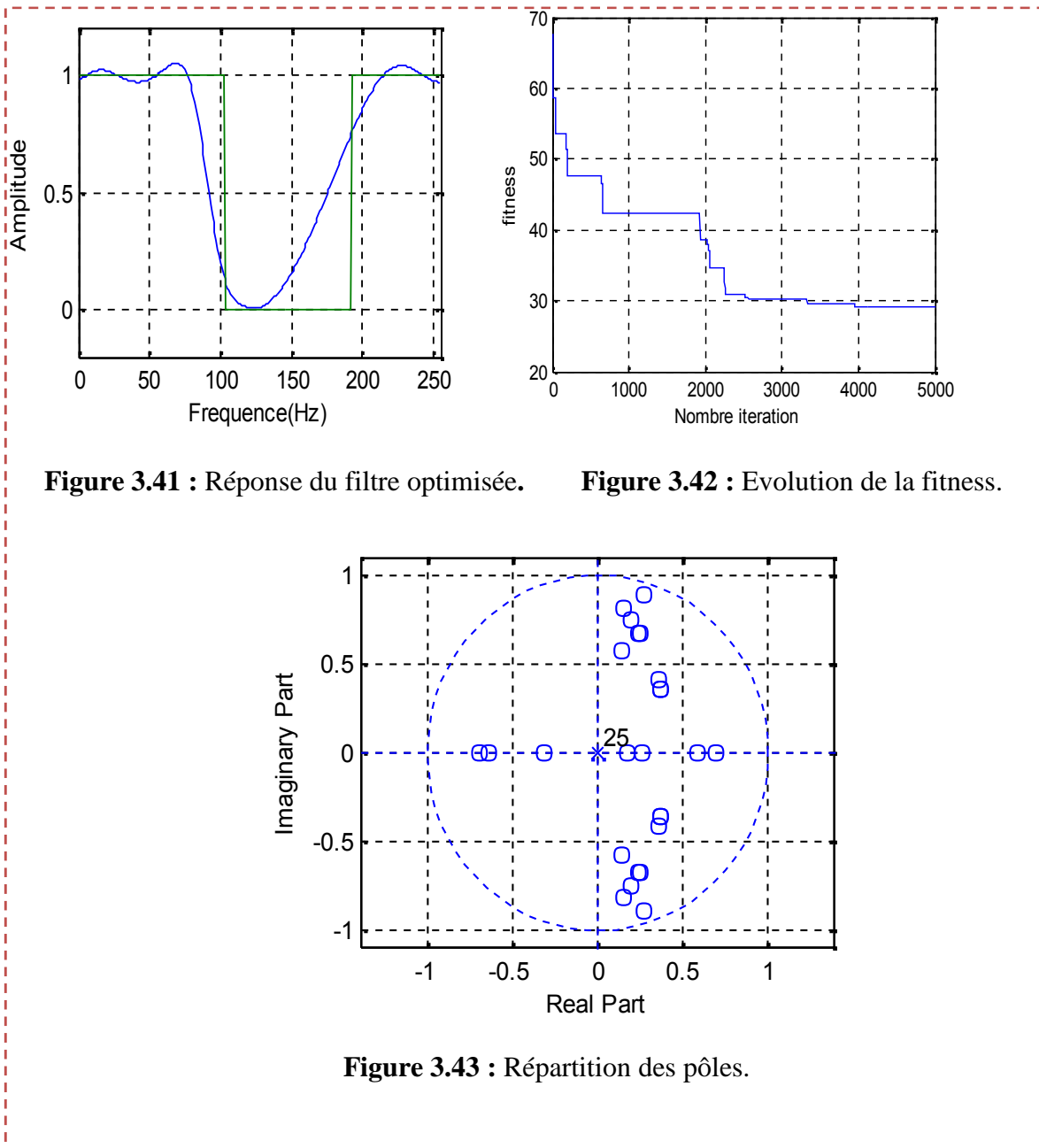
 **Fitness =**

➤ **Cas d'un filtre stop bande**

 **Les coefficients de filtre sont :**

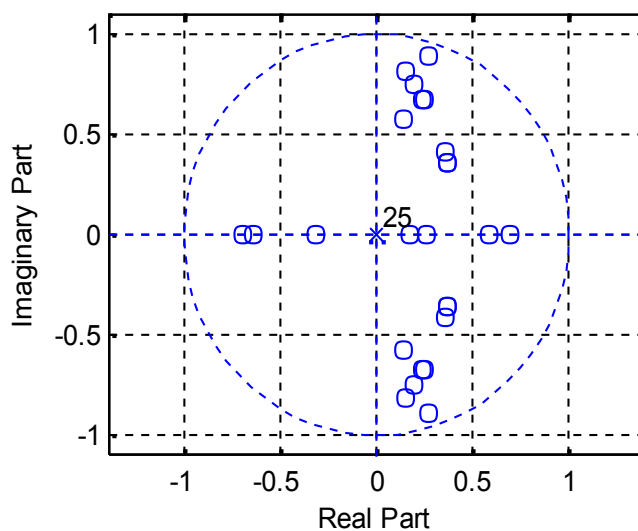
✓ [a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 a15 a16 a17  
a18 a19 a20 a21 a22 a23 a24]=[-0.7373 0.2627 -0.7074  
0.2926 -0.4839 0.5161 -0.3154 0.6846 -0.7390  
0.2610 -0.8758 0.1242 -0.8477 0.1523 -0.4923  
0.5077 -0.3929 0.6071 -0.2931 0.3529 -0.5411  
0.8590 0.6907 0.6412]

✓ [b1 b2 b3 b4 b5 b6 b7 b8 b9 b10 b11 b12 b13 b14 b15 b16 b17  
b18 b19 b20 b21 b22 b23 ]=[ -0.2057 0.7943 -0.6147  
0.3853 -0.6343 0.3657 -0.1782 0.8218 -0.4840  
0.5160 -0.0980 0.9020 -0.7883 0.2117 -0.8813  
0.1187 -0.2647 0.7353 -0.0404 0.5084 -0.2931  
-0.1406 -0.1556]




**Figure 3.41 :** Réponse du filtre optimisée.

**Figure 3.42 :** Evolution de la fitness.



**Figure 3.43 :** Répartition des pôles.

 **Fitness =29.1135**

D'après la figure 3.41 on remarque qu'on utilisant un filtre d'ordre 25 on arrive à l'optimiser le gabarit exigé néanmoins avec une convergence de l'algorithme au bout de la 3500<sup>ème</sup> itération, voir figure 3.42.

Les filtres sont stables parce que les pôles sont à l'intérieur du cercle unité voir la figure 3.43.

D'après les figures (3.33); (3.36); (3.39) et (3.42) on remarque aussi que une augmentation de la bande passante induit une augmentation de temps de convergence.

#### 4) La Bande de fréquence petite :

- ❖  $F_e=20000$  Hz.
- ❖  $F_{c1}=4000$  Hz.
- ❖  $F_{c2}=5500$  Hz.

On prend les résultats de la première manipulation avec le filtre numérique d'ordre 25.

- **Cas de filtre passe-bas** :(voir la figure (3.20),(3.21) ;(3.22)).
- **Cas de filtre passe-haut** :(voir la figure (3.23) ;(3.24) ;(3.25)).
- **Cas de filtre passe-bande** :(voir la figure (3.26) ;(3.27) ;(3.28)).
- **Cas de filtre stop-bande** :(voir la figure (3.29) ;(3.30);(3.31)).

#### On constat que :

✓ Tous les filtres d'ordre 25 nous donnent de bons résultats même avec changement au niveau des fréquences de coupure  $f_{c1}$ ,  $f_{c2}$  et la fréquence d'échantillonnage, voir figures (3.32); (3.35) (3.38) et (3.41), mais on constate que les résultats sont meilleurs dans le cas des bandes passantes qui sont petites surtout dans les cas de filtre passe-bande et stop-bande.

### 3.4 Conclusion

Les résultats obtenus par ces simulations, envisagent nettement l'efficacité de l'algorithme de l'OEP pour la détermination des coefficients des filtres numérique du type RII ( $a_i, b_i$ ) en passant par un problème d'optimisation.

On remarque bien, sur les courbes d'ordre 25 sont nettement meilleures que se du filtre d'ordre 8, donc l'ordre a une grande importance dans la convergence et la précision de l'OEP. De même que si le nombre d'itérations augmente on peut obtenir de résultat meilleur.

De même le nombre d'itération doit être suffisamment important, pour confirmer la convergence de l'algorithme, pour cette raison là il faut choisi un bon compromis entre le nombre d'itération et la précision, qui sera un choix judicieux.

Dans le cas où  $N = 25$  on a réalisé les quatre types de filtre numérique RII avec la même fonction fitness, ce qui confirme la souplesse de l'OEP dans la conception des filtres numérique.

## CONCLUSION GENERALE

L'étude présentée dans ce mémoire porte sur, l'application de la méthode d'optimisation par essaim de particules inspiré de comportement social des animaux dans la synthèse et l'optimisation des paramètres des filtres numériques de type RII .

D'après les résultats qu'on a trouvés l'OEP a pu optimiser de manière satisfaisante plusieurs types de filtre numérique RII (Passe- Bas (PB), Passe- Haut(PH), Passe- Band (PB), et Stop- Band (SB)) on utilisant un minimum de connaissance de ces derniers à sa voir les conditions de stabilité d'un filtre et son gabarit. On a remarqué aussi que l'ordre de filtre est un paramètre important pour satisfaire un gabarit exigé.

En plus, les résultats qu'on a trouvés montrent la possibilité d'utiliser les méthodes d'optimisation (heuristiques et méta-heuristiques) dans les domaines d'électroniques et de télécommunication.

En plus des avantages issus des méthodes traditionnelles d'optimisation, l'OEP possède les qualités suivantes:

➤ Contrairement aux autres Algorithmes heuristiques, ce dernier a une grande flexibilité de contrôle qui permet de balancer entre l'exploration globale et locale de l'espace de recherche en jouant seulement sur le paramètre d'inertie  $w$  et les coefficients de confiance  $c1$  et  $c2$ .

➤ L'OEP utilise des règles de transition stochastique, et non pas déterministes. Cela permet d'obtenir une recherche dans des domaines considérés compliqués et incertains. Cet avantage confère à l'OEP une grande robustesse et une flexibilité qui dépassent celles des méthodes conventionnelles d'optimisation à ca voir l'optimisation pour la détermination de l'ordre de numérateur et dénominateur de filtre numérique RII.



## REFERENCES

- [1] – *BENOUARED .A.* « Synthèse de Filtres Numériques en Précision Finie, Utilisant les Algorithmes Génétiques » Mémoire de Magister en Electrotechnique ; Spécialité :Signaux et Systèmes -Université de SAAD DAHLED de BLIDA-**2009**.
- [2] – *BENOUARED .A. ET KOUIDER ELOUAHED B.* « SYNTHÈSE DU DIAGRAMME DE RAYONNEMENT D’UN RESEAU D’ANTENNES PAR LA METHODE : PARTICLE SWARM OPTIMIZATION (**PSO**) » MEMOIRE D’INGENIEURE EN ELECTROTECHNIQUE ; SPECIALITE : COMMUNICATION: UNIVERSITE DE SAAD DAHLED DE BLIDA- **2010**.
- [3] –*BENOUARED .A.* « Conception d’un Filtre Numérique Par Algorithme Génétique Hiérarchique »Mémoire d’ingénieure en Electrotechnique; Spécialité Communication/ contrôle : Université de SAAD DAHLED de BLIDA- **2004**.
- [4] – *M.DOUDOU* : “cours de transmission numérique de signal TNS de 4eme année”. Département d’aéronautique de BLIDA.
- [5] – *M. BAZI Smail* - «Contribution à la Commande Robuste d’une Machine Asynchrone par la Technique **PSO**» Mémoire de Magister en Electrotechnique Université de BATNA-**2004**.
- [6] – *Amir NAKIB*- « Conception de méta-heuristiques d’optimisation pour la segmentation d’images. Application à des images biomédicales » THÈSE DE DOCTORAT université de PARIS- **2007**.
- [7] – *M. Clerc, ed. Hermes* ” L’optimisation par essais particuliers” -**2005**.
- [8]- *Suganthan,P.* “Particle Swarm Optimizer with Neighborhood Operator.IEEE Congress on Evolutionary Computation, volume III” -**1999**.
- [9] - *Kennedy, J.* “Small Worlds and Mega-Minds: Effects of Neighborhood Topology on Particle Swarm Performance. In IEEE Congress on Evolutionary Computation, volume III” -**1999**.
- [10] -*Jin-Kao Hao, Philippe Galinier, Michel Habib* “Méta-heuristiques pour l’optimisation combinatoire et l’affectation sous contraintes. Revue d’intelligence artificielle”-**1999**.

- [11] - *Kennedy, J. and Eberhart, R.* "Particle Swarm Optimization. Of IEEE International Conference on Neural Network, volume IV" -**1995.**
- [12] - *Van den Bergh, F.* "An Analysis of Particle Swarm Optimizers. PhD thesis, Department of Computer Science, University of Pretoria" -**2002.**
- [13] - *Clerc, M. and Kennedy, J.* "The Particle Swarm: Explosion, Stability, and Convergence in a Multi-Dimensional Complex Space. In Proceedings of the IEEE Transactions on Evolutionary Computation, volume VI " -**2002.**
- [14] - *Éric Taillard.* " Principes d'implémentation des méta-heuristiques, in chapitre 2 de *J. Teghem, M. Pirlot (dir.)* : " Optimisation approchée en recherche opérationnelle. Hermès, p. 57-79" -**2002.**
- [15] -*Christian Blum, Andrea Roli* : "Meta-heuristics in Combinatorial Optimization: Overview and Conceptual Comparison. ACM Computing Survey, Vol. 35"-**Sept. 2003.**
- [16] - *Marco Domingo and Gianni Di Caro*: "Ant Algorithms for Discrete Optimization. Université Libre de Bruxelles"-**2004.**
- [17] - *Vincent Bachelet, Zouhir Hafidi, Philippe Preux, El-Ghazali Talbi* :  
" La coopération des méta-heuristiques" -**2009.**
- [18] - *Bipul Luitel, Ganesh K* : "Particle swarm optimization with quantum infusion for system identification" Engineering Applications of Artificial Intelligence USA-**2010.**
- [19] - *Louis MO* : "Particle Swarm Optimization with Quantum Infusion for the Design of Digital Filters" IEEE Swarm Intelligence Symposium USA, -**September 2008.**