



Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique

MÉMOIRE DE MASTER

Sciences et Technologies
Filière : Automatique
Spécialité : Automatique et informatique industrielle

Réf. :

Présenté et soutenu par :
Achouri Souhila

Le : dimanche 27 septembre 2020

Etude de l'Identification d'un Système d'Ordre Fractionnaire

Jury :

Pr.	Benchaabane Fateh	Pr	Université de Biskra	President
Dr.	Mihi Assia	MCB	Université de Biskra	Examineur
Dr.	Megherbi Hassina	MCA	Université de Biskra	Rapporteur

Année universitaire : 2019- 2020

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement Supérieur et de la recherche scientifique



Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : Automatique
Option : Automatique et Informatique industrielle

Mémoire de Fin d'Etudes
En vue de l'obtention du diplôme:

MASTER

Thème

Etude de l'Identification d'un Système
d'Ordre Fractionnaire

Présenté par :

Achouri souhila

Avis favorable de l'encadreur :

Megherbi Hassina

signature

Avis favorable du Président du Jury

Nom Prénom

Signature

Cachet et signature

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement Supérieur et de la Recherche scientifique



Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : Automatique

Option : Automatique et Informatique industrielle

Thème :

Etude de l'identification d'un système d'ordre fractionnaire

Proposé par : Megherbi Hassina

Dirigé par : Megherbi Hassina

RESUMES (Français et Arabe)

Dans ce travail, nous avons exploré l'utilisation d'une technique d'optimisation pour déterminer les paramètres optimaux du correcteur PID et $PI^{\alpha}D^{\mu}$ fractionnaire, à savoir α , μ , K_p , K_i et K_d , qui minimisent une fonction objective définie en terme de l'erreur de régulation. On a utiliser un algorithme d'optimisation par Essaims particulaires (PSO). Une comparaison entre les performances obtenus par la correction PID et $PI^{\alpha}D^{\mu}$ fractionnaire nous démontre la supériorité de ce dernier, ainsi que les performances du contrôleur fractionnaire. L'efficacité de l'algorithme PSO est illustrés par une application d' une régulation de vitesse d'un moteur a courant continu.

يركز هذا العمل على تحديد نظام ذو ترتيب كسري من خلال طريقة التحسين بأسراب الجسيمات (Particle Swarm Optimization - PSO). الخوارزمية المستخدمة هي خوارزمية تحسين PSO بمعامل عطالة متناقص خطياً و آلية حبس للسرعة فقط. تعمل هذه الخوارزمية على تحسين الأوامر الكسرية ومعاملات نظام ذي بنية معروفة. كانت نتائج المحاكاة التي تم الحصول عليها مرضية للغاية وأظهرت فعالية التقنية المستخدمة.

DEDICACES

A ma très chère mère

Quoi que je fasse ou je dis, je ne saurai point te remercier comme il se doit.

A mon très cher père.

Tu as toujours a mes cotés pour me soutenir et m'encourager

A ma sœur AYA.

T'est une personne tout simplement merveilleuse, tu es ma meilleure amie.

A mes frère YUCEF et IDRIS.

Vous êtes l'espoir dans ma vie.

A mon bébé d'amour ABDELMALEK ANES.

A la mémoire de quel qu'un qui me vraiment manques dans cette vie, mon grand- père

MEROUANE RABAH.

Sans oublier l'amie de la vie RAYEN.

A toute ma famille la famille de mon père, la famille de ma mère.

A tous ce qui me souhaite le succès et le bonheur.

REMERCIEMENTS

La première personne que je tiens à remercier est mon encadreur Mme Megherbi Hassina, pour l'orientation, la confiance, la patience qui ont constitué un apport considérable sans lequel ce travail n'aurait pas pu être mené au bon port, qu'elle trouve dans ce travail un hommage vivant à sa haute personnalité.

A tous ceux qui m'ont enseigné durant les 5 ans de l'université un très grand merci.

Liste des Tableaux.....	v
Liste des Figures.....	vi
Liste des Abréviations.....	vii
INTRODUCTION GENERALE.....	1
I Chapitre I : Concepts de base d'un système d'ordre fractionnaire	1
I.1 Introduction :	1
I.1 Les opérateurs d'ordre fractionnaire.....	1
I.2 Quelques propriétés de la dérivée d'ordre fractionnaire :	3
I.3 La transformation de Laplace des opérateurs d'ordre fractionnaire.....	4
I.4 Méthodes d'approximation des opérateurs d'ordre fractionnaire.....	5
I.5 Représentation des systèmes d'ordre fractionnaire	7
I.6 Identification des Systèmes Linéaires d'Ordre fractionnaire	9
I.7 Identification fréquentielle des systèmes d'ordre fractionnaire :	9
I.8 Identification temporelle des systèmes d'ordre fractionnaire :.....	11
I.9 Conclusion :	16
II Chapitre II L'optimisation et l'algorithme d'optimisation par essais particuliers	17
II.1 Introduction	17
II.2 Définition de l'optimisation.....	18
II.3 Types de problèmes d'optimisation.....	19
II.4 Classification des méthodes d'optimisation	20
II.5 Principe de méthodes méta-heuristiques	23
II.6 Optimisation par Essaim Particulaire (PSO):	24
II.7 Principe de fonctionnement de PSO :.....	25
II.8 Algorithme de base de PSO :.....	31
II.9 Configuration des paramètres	32
II.10 Amélioration des algorithmes PSO.....	33
II.11 Avantages et inconvénient de l'algorithme PSO :	40

II.12	Conclusion	40
III	Chapitre III : Algorithme PSO pour l'identification d'un système fractionnaire.....	42
III.1	Introduction.....	42
III.2	Le système d'ordre fractionnaire à identifier.....	42
III.3	Principe d'identification d'un système fractionnaire à base d'un algorithme PSO.....	42
III.4	L'algorithme de particule d'essaim (PSO)	43
III.5	Simulation et discussions :.....	47
III.6	Conclusion :	53
	Conclusion générale	54
	Perspectives.....	55
	Références bibliographique.....	56

Liste des Tableaux

Tableau 4-1. Les paramètres du modèle optimisés par l’algorithme PSO proposé dans les 10 exécutions.....	49
--	----

Liste des Figures

Figure I.1 Classification des systèmes linéaires invariants dans le temps.	8
Figure I.2 Principe de la méthode à erreur de sortie	12
Figure I.3 Décomposition modale d'une fonction de transfert.	14
Figure I.4 Modèle à erreur d'équation.	15
Figure II.1 Les optima locaux vs. L'optimum global.	18
Figure II.2 Classification des problèmes d'optimisation.	20
Figure II.3 Classification des méthodes d'optimisation.	23
Figure II.4 Les fonctions principales des méthodes d'optimisation méta-heuristiques.	24
Figure II.5 Structure schématique d'une particule dans l'algorithme PSO	24
Figure II.6 schéma de principe du comportement des particules d'un essaim	25
Figure II.7 Exemple de modèle de simulation du comportement de particule	26
Figure II.8 Organigramme général de la méthode PSO	29
Figure II.9 Schéma d'itération des particules	30
Figure II.10 L'algorithme PSO de base.	31
Figure II.11 Topologies du voisinage les plus utilisées.	32
Figure II.12 Confinement de position par absorption.	33
Figure II.13 Confinement de position par réflexion.....	33
Figure II.14 Confinement de position par retour aléatoire.....	34
Figure II.15 Calcul de la dimension particulaire moyenne pour N particules [54].....	39
Figure III.1 Schéma de principe d'identification paramétrique par l'algorithme PSO.	43
Figure III.2. Stratégie de confinement de vitesse.....	45
Figure III.3. Organigramme du PSO utilisé.	46
Figure III.4 courbes de convergence de l'algorithme PSO avec des essais initiaux différentes.....	48
Figure III.5 La réponse indicielle du modèle identifié et du système réel.	50
Figure III.6 La réponse du modèle identifié et du système réel à un signal d'entrée sinusoïdal.	51
Figure III.7 La réponse du modèle identifié et du système réel à un signal d'entrée rampe....	51
Figure III.8. Signal d'entrée PRBS.	52
Figure III.9 La réponse du modèle identifié et du système réel à un signal d'entrée PRBS....	52

Liste des Abréviations

PSO: Particle Swarm Optimization

PRBS : Pseudo Random Binairy Sequence

LIT : Linear Invariant Time

FIPS : Fully Informed Particle Swarm

DE: Differential evolutionnairy

DEUS : DE Updating Strategy

RUS : Random Updating Strategy

PSOUS: PSO Updating Strategy

EPU-PSO : Efficient population utilization

CLPSO : Comprehensive Learning Particle Swarm Optimization

UPSO: Unified Particle Swarm Optimization

FCPSO : Fast Convergence Particle Swarm Optimization en anglais)

PMD : Particle Mean Dimension

Résumé

Ce travail porte essentiellement sur l'identification d'un système d'ordre fractionnaire par la méthode d'optimisation méta-heuristique des essaims particulaires (Particle Swarm Optimization –PSO en anglais). L'algorithme utilisé est un algorithme d'optimisation des essaims particulaires à poids d'inertie décroissant linéairement avec un mécanisme de confinement sur la vitesse seulement. Cet algorithme optimise les ordres fractionnaires et les coefficients d'un système à structure connue. Les résultats de simulation obtenus étaient très satisfaisants et ils ont montré l'efficacité de la technique utilisée.

ملخص

يركز هذا العمل على تحديد نظام ذو ترتيب كسري من خلال طريقة التحسين بأسراب الجسيمات (Particle Swarm Optimization - PSO). الخوارزمية المستخدمة هي خوارزمية تحسين PSO بمعامل عطالة متناقص خطياً و آلية حبس للسرعة فقط. تعمل هذه الخوارزمية على تحسين الأوامر الكسرية ومعاملات نظام ذي بنية معروفة. كانت نتائج المحاكاة التي تم الحصول عليها مرضية للغاية وأظهرت فعالية التقنية المستخدمة.

INTRODUCTION GENERALE

Durant les dernières décennies, le domaine du contrôle et d'automatisation industriel a connu une multiplicité d'effort de recherche en développement technologique et théorique [1-7]. Cela est dû à cause de la concurrence et des besoins de plus en plus exigeants de point de vue qualité et performances. Le progrès technologique est une conséquence du grand saut qualitatif qu'a connu le domaine informatique au niveau de logiciel ou en matériel.

L'une des théories qui peut être considérée aussi bien ancienne que nouvelle et qui connaît actuellement une grande popularité parmi les chercheurs dans les sciences fondamentales, ingénierie et en particulier en automatique, est celle du Calcul Fractionnaire qui étend la dérivation et l'intégration aux ordres fractionnaires. Au début c'était presque un jeu d'esprit pour certains mathématiciens de renommée, qui voulaient généraliser la notion de différentiation d'ordre entier par des opérateurs d'ordre fractionnaire, permettant le calcul de la dérivée d'ordre réel ou complexe d'une fonction différentiable [8].

De nombreuses définitions ont été alors données sur la dérivation et l'intégrale d'ordre fractionnaire [9-11], leur application dans le domaine des systèmes de commande automatique remonte au début des années soixante [12]. Mais, ce n'est que dans les dernières décennies que les contrôleurs basés sur le calcul d'ordre fractionnaire gagnent de plus en plus d'intérêts [13]. De nombreux procédés dans les systèmes de physique et d'ingénierie peuvent être commandés et modélisés plus précisément par des dérivés/intégrales fractionnaires que par les dérivées ou intégrales d'ordre entier traditionnel [14]. Miller et Ross, [15], ont mentionné que presque tous les domaines de la science et de l'ingénierie ont une application des dérivés fractionnaires [16].

Le problème d'identifier un système à partir de données réelles devient difficile lorsque l'ordre fractionnaire est introduit. Pour un modèle entier, une fois l'ordre maximal du système identifier est fixé, les paramètres du modèle peuvent être optimisés directement, alors, que l'identification des systèmes d'ordre fractionnaire nécessite le choix du nombre d'opérateurs fractionnaire, la puissance fractionnaire de chaque opérateur et les coefficients des opérateurs [17].

Aujourd'hui, l'optimisation joue un rôle important en automatique. Elle est une discipline mathématique qui concerne la découverte de minimale et de maximale de fonctions ou de processus. La maximisation d'une fonction f équivaut à la minimisation de l'opposé de cette

fonction, l'optimisation, se compose de différentes techniques, est utilisé dans différents domaines tels que la connaissance de l'ordinateur et l'IA et a également un grand effet sur l'industrie et les entreprises.

Les méthodes d'optimisation méta-heuristiques sont des méthodes approchées qui traitent des problèmes d'optimisation difficile qui peuvent aider à résoudre le problème de l'optimisation des paramètres d'un système d'ordre fractionnaire. La méthode d'optimisation des essaims particulaires (PSO : Particle Swarm Optimisation) est une méthode d'optimisation méta-heuristique simple fondée sur le comportement coopératif de groupes d'individus [18].

Dans ce travail cette méthode est utilisée pour optimiser les ordres fractionnaires et les coefficients des opérateurs fractionnaires dont le nombre est supposé connu.

Ce travail est structuré en 4 chapitres :

- Le premier chapitre présente une introduction aux notions de base théorique du calcul fractionnaire, ainsi que les principales propriétés des opérateurs d'ordre fractionnaire.
- L'objet du deuxième chapitre est l'utilisation et l'exploitation du concept de la dérivation d'ordre fractionnaire dans un contexte d'identification paramétrique des systèmes fractionnaires. Au cours de laquelle certaines méthodes ont été expliquées brièvement dans l'identification temporelle et fréquentielle.
- le troisième chapitre présente des généralités sur l'optimisation et les méthodes d'optimisation ainsi que les détails sur l'algorithme utilisé qui est un algorithme d'optimisation des essaims particulaires.
- le dernière chapitre est réservé à la simulation de l'identification d'un système d'ordre fractionnaire avec l'algorithme PSO, et les principaux résultats réalisés

I Chapitre I: Concepts de base d'un système d'ordre fractionnaire

I.1 Introduction :

Le calcul fractionnaire est une généralisation du calcul régulier (ordre entier). Il est considéré comme un sujet ancien et encore nouveau. Durant les dernières décennies, le calcul fractionnaire s'est développé et a pris de l'importance aussi bien dans le domaine des mathématiques que dans des applications scientifiques. Bien que le calcul d'ordre fractionnaire soit un sujet ancien qui a apparu il y a trois siècles, le concept des opérateurs d'ordre fractionnaire a été développé principalement au 19ème siècle par Riemann et Liouville dont le but était d'étendre la dérivation et l'intégration classique en employant non seulement des ordres entiers mais également des ordres non entiers [19]. Beaucoup de travaux de recherche scientifique ont montré l'importance de ces systèmes d'ordre fractionnaire et leur intérêt dans différents domaines scientifiques. Les applications concernent des domaines très variés des sciences physiques, tels que la mécanique, l'électricité, la chimie, la biologie, l'économie, la modélisation, l'identification aussi bien dans le domaine temporel que fréquentiel et notamment la commande des processus, la mécatronique et la robotique [20].

I.1 Les opérateurs d'ordre fractionnaire

Le calcul fractionnaire est une généralisation de l'intégration et de la différentiation à l'opérateur fondamental d'ordre non entier (α) noté ${}_{t_0}D_t^\alpha$; où t_0 et t sont des limites de l'opération. L'opérateur intégro-différentiel continue est défini comme [21] :

$${}_{t_0}D_t^\alpha = \begin{cases} \frac{d^\alpha}{dt^\alpha} & \alpha > 0 \\ 1 & \alpha = 0 \\ \int_{t_0}^t (dt)^{-\alpha} & \alpha < 0 \end{cases} \quad (\text{I.1})$$

Où $\alpha \in \mathbb{R}$ est l'ordre de l'opération. Il existe plusieurs définitions mathématiques pour l'intégration et la dérivation d'ordre fractionnaire. Les plus connues sont la définition de

Grundwald-Leitnikov, la définition de Riemann-Liouville et définition de Caputo. Ces définitions ne donnent pas toujours des résultats identiques mais elles sont équivalentes pour un large panel de fonctions.

I.1.1 Définition de Grundwald-Leitnikov

Soit $\alpha \in \mathbb{R}^+$; $t_0 \in \mathbb{R}$ et f une fonction localement intégrable définie sur $[t_0, +\infty]$

La définition de Grundwald-Leitnikov, notée G-L, de la dérivée d'ordre fractionnaire α de la fonction f de borne inférieure t_0 est donnée par [21] :

$${}^{GL}D_t^\alpha f(t) = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{k=0}^{\lfloor \frac{t-t_0}{h} \rfloor} (-1)^k \binom{\alpha}{k} f(t - kh) \quad (I.2)$$

Où $\lfloor . \rfloor$ dénote la partie entière, h est le temps d'échantillonnage et les coefficients $\binom{\alpha}{k}$ sont données par :

$$\binom{\alpha}{k} = \frac{\Gamma(\alpha + 1)}{\Gamma(k + 1)\Gamma(\alpha - k + 1)} \quad (I.3)$$

Où $\Gamma(.)$ est la fonction gamma d'Euler défini par :

$$\Gamma(\alpha) = \int_0^\infty y^{\alpha-1} e^{-y} dy, \alpha > 0 \quad (I.4)$$

Avec $\Gamma(\alpha + 1) = \alpha\Gamma(\alpha)$ et $\Gamma(1/2) = \sqrt{\pi}$

La définition de l'intégration de G-L d'ordre fractionnaire est formulée comme suit :

$${}^{GL}I_t^\alpha f(t) = {}^{GL}D_t^{-\alpha} f(t) = \lim_{h \rightarrow 0} h^\alpha \sum_{k=0}^{\lfloor \frac{t-t_0}{h} \rfloor} (-1)^k \binom{\alpha}{k} f(t - kh) \quad (I.5)$$

I.1.2 Définition de Riemann-Liouville

I.1.2.1 Définition 1 :

L'intégrale d'ordre fractionnaire α de la fonction f de borne inférieure t_0 selon la définition de Riemann-Liouville est donnée par [21] :

$${}^{RL}I_t^\alpha f(t) = \frac{1}{\Gamma(\alpha)} \int_{t_0}^t (t - \tau)^{\alpha-1} f(\tau) d\tau \quad (I.6)$$

Avec $t \geq t_0$.

1.1.2.2 Définition2 :

La dérivée d'ordre fractionnaire α de la fonction f de borne inférieure t_0 selon la définition de Riemann-Liouville est définie par [21] :

$${}^{RL}D_t^\alpha f(t) = \frac{1}{\Gamma(n - \alpha)} \frac{d^n}{dt^n} \int_{t_0}^t (t - \tau)^{n-\alpha-1} f(\tau) d\tau \quad (1.7)$$

Où le nombre n est un entier tel que $n-1 < \alpha < n$.

1.1.3 Définition de Caputo :

La dérivée d'ordre fractionnaire α de la fonction f de borne inférieure t_0 introduite par Caputo est définie par [3]:

$${}^C D_t^\alpha f(t) = I^{n-\alpha} D^n f(t) = \frac{1}{\Gamma(n - \alpha)} \int_{t_0}^t \frac{f^{(n)}(\tau)}{(t - \tau)^{\alpha-n+1}} d\tau \quad (1.8)$$

Avec n est un entier positif vérifiant l'inégalité $(n-1) < \alpha < n$. Cette définition peut être formulée également en fonction de la définition de Riemann-Liouville comme suit :

$${}^{RL}D_t^\alpha f(t) = {}^C D_t^\alpha f(t) + \sum_{k=0}^{n-1} \frac{t^{k-\alpha}}{\Gamma(n - \alpha + 1)} f^{(k)}(0^+) \quad (1.9)$$

Ce qui peut être écrit autrement par :

Remarque : Dans ce qui suit, pour simplifier l'écriture on notera D^α pour ${}^{RL}D_t^\alpha$ et I^α pour ${}^C I_t^\alpha$.

I.2 Quelques propriétés de la dérivée d'ordre fractionnaire :

Les propriétés fondamentales des opérateurs d'ordre fractionnaire sont les suivantes [21] :

- Si $f(t)$ est une fonction analytique de t , alors sa dérivée d'ordre fractionnaire $D^\alpha f(z)$ est une fonction analytique de t et α .
- Pour $\alpha = n$, tel que n un entier, l'opération $D^\alpha f(z)$ donne le même résultat que la différentiation classique d'ordre entier n .
- Pour $\alpha = 0$, l'opération $D^0 f(z)$ est l'opérateur d'identité :

$$D^0 f(z) = f(z)$$

- La différentiation et l'intégration d'ordres fractionnaire sont des opérations linéaires

- $D^m\{af(z) + bg(z)\} = aD^m f(z) + bD^m g(z)$

I.3 La transformation de Laplace des opérateurs d'ordre fractionnaire

Les définitions des opérateurs de dérivation et d'intégrale d'ordre fractionnaire définies antérieurement dans le domaine temporelle se basent sur des formalismes un peu compliqués. Dans le domaine de Laplace, elles sont caractérisées par une simplicité remarquable.

I.3.1 La transformation de Laplace de l'intégral d'ordre fractionnaire

D'après l'intégrale d'ordre fractionnaire de Riemann-Liouville d'ordre $\alpha > 0$, définie par l'équation (I.5); qui peut être écrite comme un produit de convolution de fonction $g(t) = \frac{1}{\Gamma(\alpha)} t^{\alpha-1}$ [21] et $f(t)$:

$$I^\alpha f(t) = D^{-\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t - \tau)^{\alpha-1} f(\tau) d\tau = \frac{1}{\Gamma(\alpha)} t^{\alpha-1} * f(t) \quad (I.10)$$

La transformée de Laplace de la fonction $t^{\alpha-1}$ est donnée comme [21]:

$$G(s) = L\{t^{\alpha-1}\} = \Gamma(\alpha) s^{-\alpha} \quad (I.11)$$

Donc la transformée de Laplace de l'intégrale d'ordre fractionnaire de Riemann-Liouville est [19]:

$$L\{I^\alpha(f(t))\} = s^{-\alpha} F(s) \quad (I.12)$$

De la même façon la transformée de Laplace de l'intégrale d'ordre fractionnaire défini par Gröndwald-Leitnikov ou par Caputo est aussi donnée par l'équation (I.15).

I.3.2 Transformée de Laplace de la dérivée d'ordre fractionnaire

I.3.2.1 Définition de Riemann-Liouville :

La transformée de Laplace de la dérivée d'ordre fractionnaire défini par Riemann-Liouville est donnée par :

$$L\{D^\alpha f(t)\} = s^\alpha F(s) - \sum_{k=0}^{n-1} s^k [D^{k-\alpha+1} f(t)]_{t=0} \quad (I.13)$$

avec $(n-1) < \alpha < n$; mais l'applicabilité pratique de la transformée de Laplace de la dérivée de Riemann-Liouville est limitée à cause de l'absence d'interprétation physique des valeurs limites des dérivées d'ordre fractionnaire pour $t = 0$ [21].

I.3.2.2 Définition de Caputo :

La transformée de Laplace de la dérivée d'ordre fractionnaire défini par Caputo est donnée par :

$$L\{D^\alpha f(t)\} = s^\alpha F(s) - \sum_{k=0}^{n-1} s^{\alpha-k-1} f^k(0) \quad (I.14)$$

Avec $\alpha > 0$. L'avantage principal de la définition de Caputo par rapport à celle de Riemann-Liouville est qu'elle permet de considérer des conditions initiales conventionnelles faciles à interpréter. En plus, la dérivée de Caputo d'une constante est zéro, alors que la dérivée de Riemann-Liouville d'une constante est une fonction non bornée à $t = 0$, [21].

1.3.2.3 Définition de Grundwald-leitnikov :

La transformée de Laplace de la dérivée d'ordre fractionnaire défini par Grundwald-Leitnikov est donnée par :

$$L\{D^\alpha f(t)\} = s^\alpha F(s) \quad \alpha > 0 \quad (I.15)$$

I.4 Méthodes d'approximation des opérateurs d'ordre fractionnaire

En général, une approximation rationnelle de l'opérateur d'ordre fractionnaire $H(s) = s^\alpha$ peut être obtenue en utilisant différentes méthodes [22], comme la méthode de Carlson, la méthode de Matsudo, la méthode d'Oustaloup et la méthode de Charef.

1.4.1 Méthode de Carlson

La méthode proposée par Carlson [23] est basée sur l'hypothèse suivante [8] :

$$(H(s))^{\frac{1}{\alpha}} - G(s) = 0 \Rightarrow H(s) = (G(s))^\alpha \quad (I.16)$$

En définissant $\alpha = \frac{1}{q}$, $m = q/2$, une fonction rationnelle d'approximation itérative de $(G(s))^\alpha = s^\alpha$ peut être donnée par :

$$H_i(s) = H_{i-1}(s) \frac{(q-m)(H_i(s))^2 + (q+m)G(s)}{(q+m)(H_i(s))^2 + (q-m)G(s)} \quad (I.17)$$

Où la valeur initiale est $H_0(s) = 1$.

1.4.2 Méthode de Matsudo

La méthode d'approximation proposée par [24], pour un ensemble de points sélectionnés W_i , $i=0,1, 2, \dots, M$, où M sont des fréquences réparties dans une bande $[W_0, W_M]$, prend la forme [8] :

$$H(s) = a_0 + \frac{s - W_0}{a_1 + \frac{s - W_1}{a_2 + \frac{s - W_2}{a_3 + \dots}}} \quad (I.18)$$

I.4.3 Méthode d'Oustaloup

La méthode d'Oustaloup [25] utilise une distribution récursive de zéros et pôles d'ordre entier, répartis dans une bande de fréquence limitée. Ainsi, l'approximation de l'opérateur s^α dans une bande de fréquence $[W_b, W_h]$, est donnée par [8] :

$$G(s) = S^\alpha \approx \hat{G}(s) = C_0 \prod_{i=0}^N \frac{1 + \frac{s}{Z_i}}{1 + \frac{s}{P_i}} \quad (I.19)$$

Avec les Z_i et P_i sont les pôles et les zéros de l'approximation. N est le nombre de pôles d'approximation.

$$\text{Si } \alpha > 0 : \begin{cases} Z_1 = W_b \sqrt{\eta} \\ P_i = Z_i \lambda, \quad i = 1, 2, \dots, N \\ Z_{i+1} = P_i \eta, \quad i = 1, 2, \dots, N \\ W_h = P_N \sqrt{N} \end{cases}, \text{ Si } \alpha < 0 : \begin{cases} P_1 = W_b \sqrt{N} \\ Z_1 = P_i \lambda, \quad i = 1, 2, \dots, N \\ P_{i+1} = Z_i \eta \quad i = 1, 2, \dots, N \\ W_h = Z_N \sqrt{N} \end{cases} \quad (I.23)$$

$$N = \text{integer} \left\{ \log \left(\frac{W_h}{W_b} \right) / \log(\alpha \eta) \right\} \quad (I.20)$$

$$\alpha = \left(\frac{W_h}{W_b} \right)^{\frac{|\lambda|}{N}}, \quad \eta = \left(\frac{W_h}{W_b} \right)^{\frac{1-|\alpha|}{N}} \quad (I.21)$$

I.4.4 Méthode de Charef :

Cette méthode proposée ressemble à la méthode d'Oustaloup. Elle permet l'approximation d'une fonction de la forme [22] :

$$H(s) = \frac{1}{\left(1 + \left(\frac{s}{p_t} \right) \right)^\alpha} \quad (I.22)$$

Avec $0 < \alpha < 1$, en une quotient polynomiale factorisée qui s'écrit comme suit [19] :

$$H(s) = \frac{1}{\left(1 + \left(\frac{s}{p_t} \right) \right)^\alpha} = \lim_N \frac{\prod_{i=0}^{N-1} \left(1 + \frac{s}{z_i} \right)}{\prod_{i=0}^N \left(1 + \frac{s}{p_i} \right)} \quad (I.23)$$

Où $N+1$ est le nombre total des singularités qui peut être déterminé par la bande fréquentielle d'approximation. L'équation (I.27) peut être tronquée à un nombre fini N , et l'approximation devient [19] :

$$H(s) = \frac{1}{\left(1 + \left(\frac{s}{p_t}\right)\right)^\alpha} \approx \frac{\prod_{i=0}^{N-1} \left(1 + \frac{s}{z_i}\right)}{\prod_{i=0}^N \left(1 + \frac{s}{p_i}\right)} \quad (I.24)$$

Les pôles et les zéros de la fonction de singularités peuvent être obtenus comme suit [19] :

$$P_0 = P_T \sqrt{b}, P_i = P_0 (a \cdot b)^i, \quad Z_i = a \cdot P_0 (a \cdot b)^i \quad (I.25)$$

Avec

$$a = 10^{\left[\frac{y}{10(1-a)}\right]}, b = 10^{\left[\frac{y}{10a}\right]}, a \cdot b = 10^{\left[\frac{y}{10a(1-a)}\right]}. \quad (I.26)$$

$$N = \text{Integer} \left[\log \left(\frac{W_{max}}{P_0} \right) / \log(ab) \right] + 1 \quad (I.27)$$

I.5 Représentation des systèmes d'ordre fractionnaire

Le diagramme de classification des systèmes Linéaires Invariants dans Temps (LIT) est donné dans la figure (I.1).

Le comportement d'un système d'ordre fractionnaire est le plus souvent décrit par l'équation différentielle ou des fonctions de transferts contenant des opérateurs d'ordre fractionnaire [19].

I.5.1 Equation différentielle généralisé :

Dans le cas mono-variable, un modèle mathématique est généralement représenté par une équation différentielle d'ordre fractionnaire suivante:

$$y(t) + \sum_{i=1}^n a_i D^{\alpha_i} y(t) = \sum_{j=0}^m b_j D^{\beta_j} u(t) \quad (I.28)$$

où $u(t)$ et $y(t)$ désignent respectivement l'entrée et la sortie du système, les coefficients a_i et b_j sont des réels, les ordres de dérivation sont ordonnés pour des raisons d'identifiabilité : $0 < \alpha_1 < \alpha_2 < \dots < \alpha_n$ et $0 \leq \beta_0 < \beta_1 < \dots < \beta_m$ et n et m sont des nombres entiers. Comme dans le cas d'une équation différentielle à dérivées entières, les ordres de dérivation doivent vérifier la contrainte $\alpha_n > \beta_m$ pour que le système soit strictement propre [26].

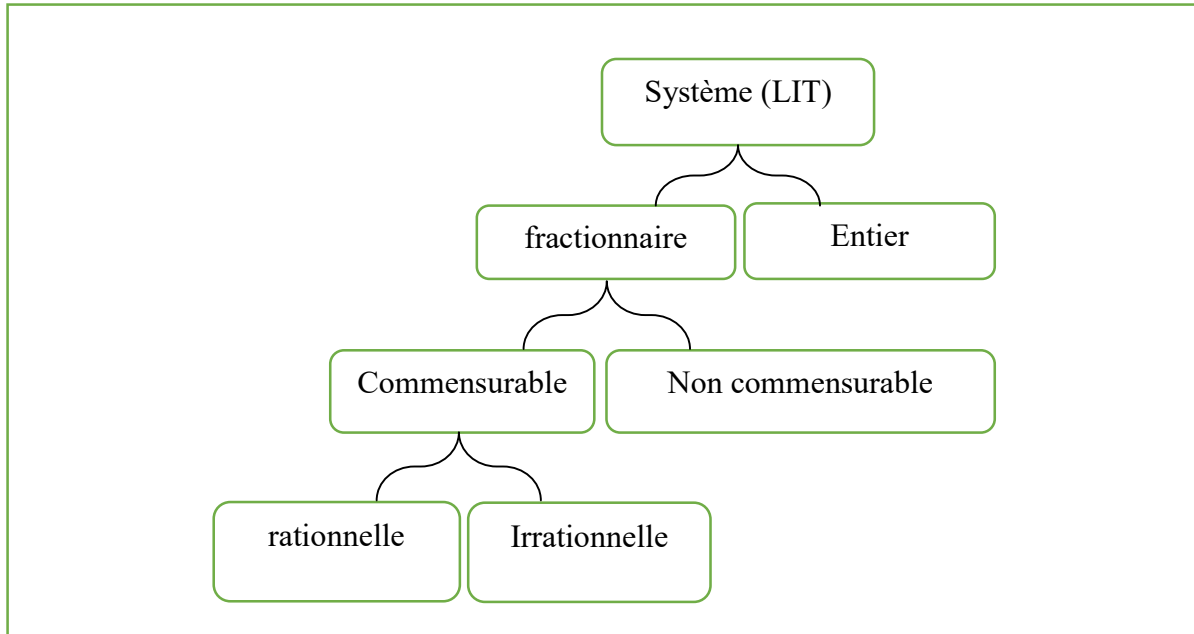


Figure I.1 Classification des systèmes linéaires invariants dans le temps.

Un système d'ordre fractionnaire est dit d'ordre commensurable lorsque tous les ordres de dérivation $\alpha_1, \alpha_2, \dots, \alpha_n$ et $\beta_0, \beta_1, \dots, \beta_m$ sont multiples du même nombre non entier α tel que $0 < \alpha < 1$. Alors l'équation (I.32) peut se mettre sous la forme [27]:

$$y(t) + \sum_{i=1}^n a_i D^{i\alpha} y(t) = \sum_{j=0}^m b_j D^{j\alpha} u(t) \quad (I.29)$$

Dans ce cas pour vérifier la contrainte $\alpha n > \beta m$ on doit avoir $n > m$ [19].

I.5.2 Fonction de Transfert d'Ordre Fractionnaire

La transformée de Laplace de l'équation différentielle de l'équation (I.32) permet de déterminer la fonction de transfert du système d'ordre fractionnaire comme suit [27] :

$$F(s) = \frac{Y(s)}{U(s)} = \frac{\sum_{i=0}^m b_j s^{\beta_j}}{1 + \sum_{i=1}^n a_i s^{\alpha_i}} \quad (I.30)$$

Dans le cas des systèmes fractionnaires d'ordre commensurable α de l'équation (I.29), cette fonction de transfert s'écrit sous la forme [19]:

$$F(s) = \frac{\sum_{i=0}^m b_j s^{j\alpha}}{1 + \sum_{i=1}^n a_i s^{i\alpha}} \quad (I.31)$$

Si l'ordre commensurable α est l'inverse d'un nombre entier Q , $F(s)$ de l'équation (I.28) s'écrit comme suit [19]:

$$F(s) = \frac{\sum_{j=0}^m b_j s^{j/Q}}{1 + \sum_{i=1}^n a_i s^{i/Q}} \quad (I.32)$$

I.6 Identification des Systèmes Linéaires d'Ordre fractionnaire

L'identification a pour but la détermination, généralement à partir de mesures physiques, des paramètres numériques d'un modèle mathématique, de telle sorte que dans le domaine de fonctionnement pour lequel il a été établi, ce modèle présente un comportement aussi proche que possible de celui du système réel [8].

Dans les dernières décennies, l'identification des systèmes d'ordre fractionnaire a gagné beaucoup d'intérêts de la communauté scientifique. Malgré que ce type de modèle pose un problème d'identification plus difficile exigeant non seulement l'estimation des coefficients du modèle mais également la détermination de ses ordres fractionnaires, l'identification des systèmes d'ordre fractionnaire est devenue un thème de recherche très actif. En plus, l'évaluation numérique de la dérivation d'ordre fractionnaire est la manipulation des matrices de données de dimension élevée conduit aux problèmes des singularités matricielles et par conséquent au biais d'estimation dû aux erreurs de calcul [19].

Deux classes d'identification des systèmes d'ordre fractionnaire ont été développées ; la première exige la connaissance a priori des ordres fractionnaires et estime seulement les paramètres du modèle mais la seconde estime les paramètres ainsi que les ordres fractionnaires des dérivés du modèle à identifier en même temps [19].

Dans la première classe des méthodes inspirées de quelques méthodes classiques fréquentielles d'identification des systèmes d'ordre entier a été proposé. Récemment, de nouvelles méthodes temporelles et fréquentielles qui forment la deuxième classe ont été développées. On peut citer entre autre les méthodes basé sur les variables instrumentales, l'algorithme des moindres carrés récursives, les méthodes d'optimisation méta heuristiques comme l'algorithme génétique, l'algorithme des essais particuliers.

I.7 Identification fréquentielle des systèmes d'ordre fractionnaire :

L'identification des systèmes d'ordre fractionnaire dans le domaine fréquentielle consiste à estimer les paramètres du modèle qui sont les coefficients du numérateur et le dénominateur

de la fonction de transfert en minimisant la moyenne des erreurs quadratiques entre la sortie du modèle et le système à identifier sur un ensemble de fréquence de mesure. Plusieurs méthodes d'identification sont développées pour les systèmes d'ordre fractionnel en se basant sur des données de la réponse fréquentielle (amplitude et phase) sont citées et décrites brièvement dans ce qui suit :

1.7.1 Méthode de Levy pour les modèles d'ordre fractionnaire

La méthode de Levy a été proposée pour la première fois pour identifier les coefficients de la fonction de transfert d'ordre entier avec la connaissance a priori des ordres du numérateur et du dénominateur. Cette méthode a été adaptée par la suite pour les modèles d'ordre fractionnaire mais nécessitant la connaissance de l'ordre commensurable [28].

1.7.2 Méthode de Vinagre

La méthode de Vinagre est une amélioration de la méthode de Levy. Elle introduit des poids en fonction de la fréquence. Ces poids ont pour effet d'améliorer la qualité d'approximation aux basses fréquences [28].

1.7.3 Méthode itérative de Sanathanan et Korner

La méthode itérative de Sanathanan et Korner est une autre méthode améliorée de la méthode de Levy qui utilise des poids qui dépendent de la fréquence et de l'itération. Elle permet alors de compenser le faible effet des données de basse fréquence dans le modèle final identifié [28].

1.7.4 Méthode itérative de Lawrence et Rogers

La méthode de Lawrence et Roger a été proposée pour éviter la résolution du système si de nouvelles données sont obtenues. C'est une méthode itérative et récursive qui consiste à mettre à jour les paramètres du modèle après chaque prise en compte d'une nouvelle donnée fréquentielle. Pour assurer la convergence, cette méthode nécessite un très grand nombre de données [29].

1.7.5 Méthode itérative de Hartley et Lorenzo

La méthode Hartley et Lorenzo permet d'obtenir un modèle d'ordre fractionnaire sous la forme :

$$G(s) = C_n s^{n\alpha} + \dots + C_2 s^{2\alpha} + C_1 s^\alpha + C_0 \quad (I.33)$$

Où bien :

$$G(s) = \frac{1}{C_n S^{n\alpha} + \dots + C_2 S^{2\alpha} + C_1 S^\alpha + C_0} \quad (I.34)$$

Où n est l'ordre du polynôme et α est l'ordre commensurable et tous deux doivent être choisis par l'utilisateur. Les coefficients C_0, C_1, \dots, C_n du modèle d'équation (II.1) sont obtenus par la résolution des équations suivantes :

$$\begin{bmatrix} G(jw_1) \\ G(jw_2) \\ \vdots \\ G(jw_{n+1}) \end{bmatrix} = \begin{bmatrix} 1 & (jw_1)^\alpha & (jw_1)^{2\alpha} & \dots & (jw_1)^{n\alpha} \\ 1 & (jw_2)^\alpha & (jw_2)^{2\alpha} & \dots & (jw_2)^{n\alpha} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & (jw_{n+1})^\alpha & (jw_{n+1})^{2\alpha} & \dots & (jw_{n+1})^{n\alpha} \end{bmatrix} \begin{bmatrix} C_0 \\ C_1 \\ \vdots \\ C_n \end{bmatrix} \quad (I.35)$$

Où w_1, w_2, \dots, w_{n+1} sont les fréquences d'échantillonnage. Pour le cas du modèle de l'équation (II.2) le côté gauche de l'équation (II.3) sera l'inverse du comportement fréquentiel.

I.8 Identification temporelle des systèmes d'ordre fractionnaire :

L'objectif de l'identification dans le domaine temporel est d'obtenir les paramètres d'un modèle donné à partir des données temporelles [19]. Pour l'identification des systèmes d'ordre fractionnaire le modèle utilisé est de la forme [27] :

$$\sum_{i=0}^n a_i D^{\alpha_i} y(t) = \sum_{j=0}^m b_j D^{\beta_j} u(t) \quad (I.36)$$

Où $u(t)$ est l'entrée, $y(t)$ est la sortie et les ordres de différentiation α_i ($0 \leq i \leq n$) et β_j ($0 \leq j \leq m$) sont des nombres réels positifs tels que $\alpha_0 < \alpha_1 < \dots < \alpha_n$; $\beta_0 < \beta_1 < \dots < \beta_m$ et $\beta_m < \alpha_n$

La fonction de transfert de ce modèle est donnée comme [27] :

$$G(s) = \frac{b_m S^{\beta_m} + b_{m-1} S^{\beta_{m-1}} + \dots + b_0 S^{\beta_0}}{a_n S^{\alpha_n} + a_{n-1} S^{\alpha_{n-1}} + \dots + a_0 S^{\alpha_0}} \quad (I.37)$$

Plusieurs méthodes d'identification sont disponibles, notamment les méthodes de l'erreur de sortie [19].

I.8.1 Méthodes à erreur de sortie

Les méthodes à erreur de sortie regroupent trois approches d'identification par modèle d'ordre fractionnaire. Leur principe est illustré dans la figure (II.1) [19]. Ces méthodes se différencient par le mode d'utilisation du modèle à identifier. La première approche utilise la forme développée d'une fonction de transfert d'ordre fractionnaire [23]. La seconde est basée sur une décomposition modale [24] et enfin, la dernière approche repose sur une décomposition en fonctions orthogonales d'ordre fractionnaire [25].

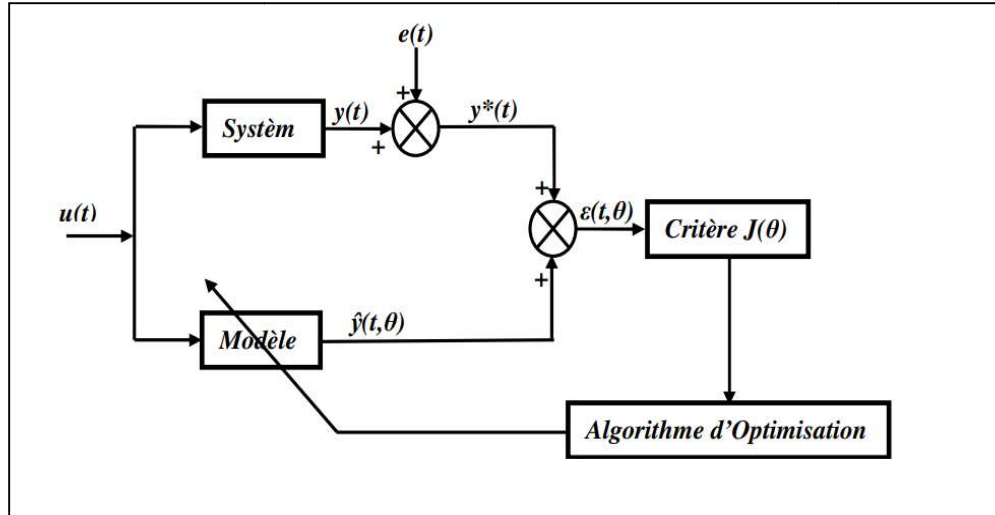


Figure I.2 Principe de la méthode à erreur de sortie

Après discrétisation des données d'entrée/sortie, Le critère à minimiser généralement utilisée qui dépend de l'erreur de sortie $\varepsilon(t_k, \theta)$, est donnée par [19] :

$$J(\theta) = \frac{1}{k} \sum_{k=0}^{k-1} \varepsilon^2(t_k, \theta) \quad (I.38)$$

avec $\varepsilon^2(t_k, \theta) = y^*(t_k) - \hat{y}(t_k, \theta)$

Où $y^*(t_k)$ est la sortie du système bruité, $\hat{y}(t_k, \theta)$ est la sortie du modèle.

1.8.1.1 Méthode Basée sur la Forme Développée du Modèle [19] :

Cette approche consiste à estimer les paramètres du modèle d'ordre fractionnaire dont la fonction de transfert est donnée par [30] :

$$G(s) = \frac{\sum_{i=0}^m b_i s^{\beta_i}}{1 + \sum_{j=1}^n a_j s^{\alpha_j}} \quad (I.39)$$

Un algorithme de programmation non linéaire est nécessaire pour identifier les paramètres du système θ (qui sont les coefficients a_j et b_i et les ordres fractionnaires α_j et β_i), car la sortie estimée $\hat{y}(t_k, \theta)$ est non linéaire par rapport à ces paramètres. Cet algorithme peut être l'algorithme de *Marquardt* qui est utilisé pour déterminer itérativement les paramètres optimales, soit [19] :

$$\hat{\theta}_{i+1} = \hat{\theta}_i - \left\{ [j + \zeta I]^{-1} j \right\}_{\theta = \hat{\theta}_i} \quad (I.40)$$

$$\text{Où } \begin{cases} j = -2 \sum_{k=0}^{k-1} \varepsilon(t_k) S(t_k, \theta) : \textit{gradient} \\ \dot{j} \approx 2 \sum_{k=0}^{k-1} S(t_k, \theta) S^T(t_k, \theta) : \textit{Hessien} \\ S(t_k, \theta) = \frac{\partial y(t_k, \theta)}{\partial \theta} : \textit{fonction de sensibilité de la sortie} \\ \zeta : \textit{paramètre de marquardt} \end{cases}$$

les fonctions de sensibilité de la sortie sont obtenues comme suit [19] :

$$\frac{\partial \hat{y}(t, \theta)}{\partial b_i} = \frac{S^{\beta_i}}{1 + \sum_{j=1}^{m_A} a_j S^{\alpha_j}} u(t); \quad \frac{\partial \hat{y}(t, \theta)}{\partial a_j} = \frac{-\sum_{i=1}^{m_A} b_i S^{\beta_i + \alpha_j}}{\left(1 + \sum_{j=1}^{m_A} a_j S^{\alpha_j}\right)^2} u(t) \quad (\text{I.41})$$

$$\frac{\partial \hat{y}(t, \theta)}{\partial \beta_i} = \frac{b_i \ln(S) S^{\beta_i}}{1 + \sum_{j=1}^{m_A} a_j S^{\alpha_j}} u(t); \quad \frac{\partial \hat{y}(t, \theta)}{\partial \alpha_j} = \frac{-a_j \ln(\sum_{i=1}^{m_A} b_i S^{\beta_i + \alpha_j})}{\left(1 + \sum_{j=1}^{m_A} a_j S^{\alpha_j}\right)^2} u(t) \quad (\text{I.42})$$

Les deux premières fonctions de sensibilité équation (II.9) peuvent être calculées facilement. Par contre, les deux autres, équation (II.10) sont plus problématiques en raison de la présence du terme $\ln(s)$; alors elles sont calculées numériquement plutôt qu'analytiquement.

1.8.1.2 Méthode de la Décomposition Modale d'une Fonction de Transfert [31] :

Cette méthode se base sur la décomposition modale de la fonction de transfert d'ordre fractionnaire commensurable qui est donnée comme suit [19] :

$$G(s) = \sum_{l=1}^L \sum_{q=1}^{\alpha_l} \frac{A_{l,q}}{(s^\alpha - s_l)^q} \quad (\text{I.43})$$

Où s_l (pour $l = 1, \dots, L$) représentent les pôles en s^α de multiplicité α_l . Quand ces pôles en s^α sont distincts la décomposition modale de la fonction de transfert, Figure I.3, sera [19] :

$$G(s) = \sum_{l=1}^L \frac{A_l}{(s^\alpha - s_l)} \quad (\text{I.44})$$

$\theta^T = [A_1, s_1, \dots, A_L, s_L, \alpha]^T$ est le vecteur de paramètres qui sera estimé par minimisation de la norme quadratique de l'erreur de sortie $\varepsilon(t_k, \theta) = y^*(t_k) - \hat{y}(t_k, \theta)$. En utilisant l'approche d'optimisation de *Marquadt*, alors les fonctions de sensibilité sont données par [19] :

$$\frac{\partial \hat{y}(t, \theta)}{\partial A_L} = \frac{1}{(s^\alpha - s_l)} u(t) \quad (I.45)$$

$$\frac{\partial \hat{y}(t, \theta)}{\partial S_L} = \frac{A_l}{(s^\alpha - s_l)^2} u(t) \quad (I.46)$$

$$\frac{\partial \hat{y}(t, \theta)}{\partial \alpha} = \sum_{i=0}^L \frac{A_l s^\alpha \ln(s)}{(s^\alpha - s_l)^2} u(t) \quad (I.47)$$

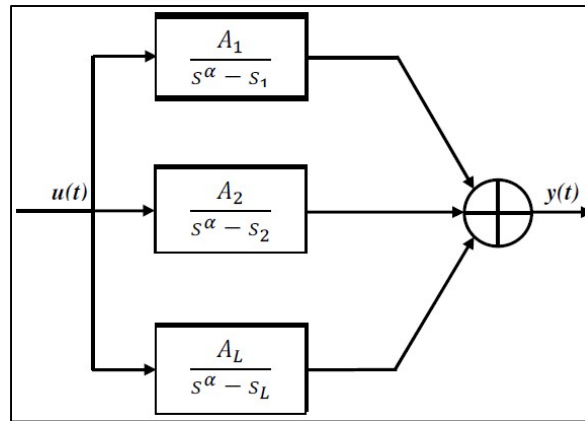


Figure I.3 Décomposition modale d'une fonction de transfert.

1.8.1.3 Méthode de décomposition en fonctions orthogonales

Avec cette méthode la fonction de transfert du système d'ordre fractionnaire à identifier est décomposée en fonctions orthogonales comme suit :

$$G(s) = \sum_{m=m_0}^M g_m G_m(s) \quad (I.48)$$

Où $\theta = [g_{m_0}, \dots, g_M]$ est le vecteur de paramètres, et $G_m(s)$ sont les fonctions orthogonales. Trois formes de fonctions orthogonales peuvent être utilisées : Les fonctions de Laguerre, Les fonctions de Kautz et les fonctions issues de la Base Orthogonale Généralisée (BOG). Pour un système stable ces fonctions orthogonales sont des fonctions connues. Alors, seuls les coefficients g_m seront identifiés par minimisation de l'erreur quadratique de sortie.

1.8.2 Méthodes à erreur d'équation :

L'objectif de cette approche consiste à estimer les coefficients $(a_1, \dots, a_l, b_0, \dots, b_m)$ d'un modèle d'ordre fractionnaire défini par une équation différentielle généralisée de la forme [8] :

$$\sum_{l=0}^L a_l D^{n_{al}} y(t) = \sum_{m=0}^M b_m D^{n_{bm}} u(t) + e(t) \quad (I.49)$$

où $u(t)$ est l'entrée, $y(t)$ est la sortie, $e(t)$ représente un bruit stochastique, $n_{al}, n_{bm} \in \mathfrak{R}, l \in [0 \dots L], m \in [0 \dots M], a_0 = 1$. L'objectif de l'identification est l'estimation paramétrique des coefficients $a_1, \dots, a_L, b_1, \dots, b_m$ en se basant sur des méthodes à erreur d'équation avec une connaissance a priori des ordres de différentiation n_{aL}, n_{b_m} .

Une fois les ordres de dérivation fixés, l'objectif principal consiste à estimer le vecteur des paramètres $\theta = [b_0, b_1, \dots, b_m, a_1, a_2, \dots, a_L]$, à temps continu à partir de K couples d'échantillons des signaux d'entrée/sortie

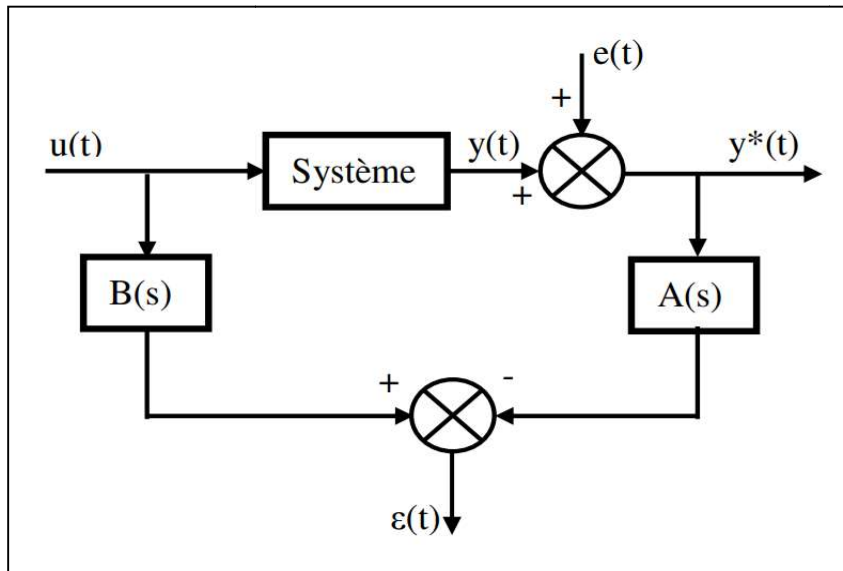


Figure I.4 Modèle à erreur d'équation.

La minimisation de l'erreur d'équation ε est donnée par:

$$\varepsilon(t, \theta) = y^*(t) - \varphi^*(t)^T \theta \quad (I.50)$$

$$\varphi^*(t)^T = [u^{(\beta_0)}(t), \dots, u^{(\beta_m)}(t), -y^{*(\alpha_1)}(t), \dots, -y^{*(\alpha_n)}(t)] \quad (I.51)$$

$$A(s) = 1 + \sum_{j=1}^n a_j s^{\alpha_j} \quad (I.52)$$

$$B(s) = \sum_{i=0}^m b_i s^{\beta_i} \quad (I.53)$$

Dans ce type de problème d'identification, l'estimation paramétrique des coefficients du modèle se décompose en trois phases bien distinctes. La première consiste à discrétiser le modèle d'ordre fractionnaire puis mettre le modèle discret ainsi obtenu sous une forme linéaire avec de nouveaux paramètres par un changement de variables. La deuxième phase consiste à estimer les nouveaux paramètres (paramètres du modèle discrétisé) par une méthode à erreur de prédiction en utilisant une de deux techniques d'optimisation des moindres carrés linéaires : la technique non récursive ou récursive. Enfin, dans la troisième phase, les paramètres du modèles originales à temps continu (coefficients a_i et b_j) seront calculés à partir des paramètres estimés dans la deuxième phase [8].

I.9 Conclusion :

Dans ce chapitre, on a présenté les trois définitions du calcul fractionnaire, la transformation de Laplace des opérateurs différentiels et quelques propriétés du système d'ordre fractionnaire. Ensuite, on a présenté quelques méthodes d'approximation suivis des représentations d'un système d'ordre fractionnaire. On a présenté aussi l'objectif de l'identification et les différentes méthodes d'identification des systèmes d'ordre fractionnaire. Selon le comportement du système d'ordre fractionnaire disponible (comportement fréquentielle -fréquence, amplitude et phase- ou temporelle -échantillons de couples entrée/sortie-), plusieurs méthodes d'identification peuvent être utilisées. La plus part d'entre eux sont des extensions des méthodes utilisées dans l'identification des systèmes d'ordre entier. Les méthodes d'identification présentées se diffèrent de la forme de la représentation du système d'ordre fractionnaire à identifié : à structure connue/inconnue ou commensurable/non-commensurable.

II Chapitre II L'optimisation et l'algorithme d'optimisation par essais particulières



II.1 Introduction

L'algorithme d'optimisation fait partie des outils mathématiques de recherche de solution et d'aide de décision, largement utilisé en informatique, disponible aux professionnels qui souhaitent mettre en place une stratégie efficace pour l'amélioration de leurs résultats. Ce type d'algorithmes est particulièrement recommandé pour la conception de solutions sur mesure destinées à des projets délicats dont les problématiques sont aussi nombreuses que complexes. L'algorithme d'optimisation est adapté à tous les secteurs d'activité humaines que ce soit scientifique, sociale, économique, technique, ...etc. Pour répondre au besoin de résoudre des problèmes complexes et/ou à grande échelle rencontrés dans ces secteurs, plusieurs méthodes d'optimisations inspirées de la nature appelées méthodes méta-heuristiques ont été proposées dans la littérature, et ont prouvé leur efficacité dans différentes applications [32], par exemple l'optimisation d'un trajet, de la forme d'un objet, du rendement d'un appareil, du choix d'investissements, ... etc. Ces méthodes se basent généralement sur des phénomènes physiques, biologiques, socio-psychologiques et font appel au hasard. Parmi ces méthodes il ya l'algorithme de recuit simulé, la recherche tabou, l'algorithme génétique, l'algorithme évolutionnaire différentielle, l'algorithme de colonies de fourmi. Nous nous intéressons dans ce chapitre à un de ces méthodes, qui est l'optimisation par essaim particulière (PSO). L'algorithme PSO est un algorithme méta-heuristique initialement proposé par Kennedy et Eberhart [33].

II.2 Définition de l'optimisation

L'optimisation est une branche de la mathématique et de l'informatique en tant que discipline. Elle consiste en la recherche du minimum (ou du maximum) d'une certaine quantité, appelée coût ou objectif. On supposera que le coût dépend de N variables réelles, rassemblées en un vecteur $x = (x_1, \dots, x_N) \in \mathbb{R}^N$, et qui fournissent une valeur $J(x)$ où J est une fonction de \mathbb{R}^N dans \mathbb{R}^N . En général, les variables x_1, \dots, x_N ne seront pas autorisées à prendre n'importe quelle valeur, mais devront satisfaire des contraintes que l'on représentera par un sous ensemble $K \subset \mathbb{R}^N$. On écrira les problèmes d'optimisation sous la forme générale suivante :

$$(P) \quad \inf_{x \in K} J(x) \quad (II.1)$$

On dit que problème (P) admet une solution s'il existe un choix de variables $x_0 \in K$ tel que :

$$\forall x \in K \quad J(x_0) \leq J(x) \quad (II.2)$$

On dit alors que x_0 est un minimiseur (ou point de minimum ou encore point optimal) de J sur K , et que $J(x_0)$ est un minimum de J sur K [34].

La fonction J est appelée la fonction objectif ou fonction coût. Le domaine K est appelé l'espace de solution ou l'espace de recherche et les éléments de K sont appelés les solutions candidates.

Dans l'espace de recherche, il peut exister des solutions intermédiaires, qui sont également des optimums, mais uniquement pour un sous-espace restreint de l'espace de recherche : on parle alors d'optimums locaux. Cette notion est illustrée sur la Figure II.1.

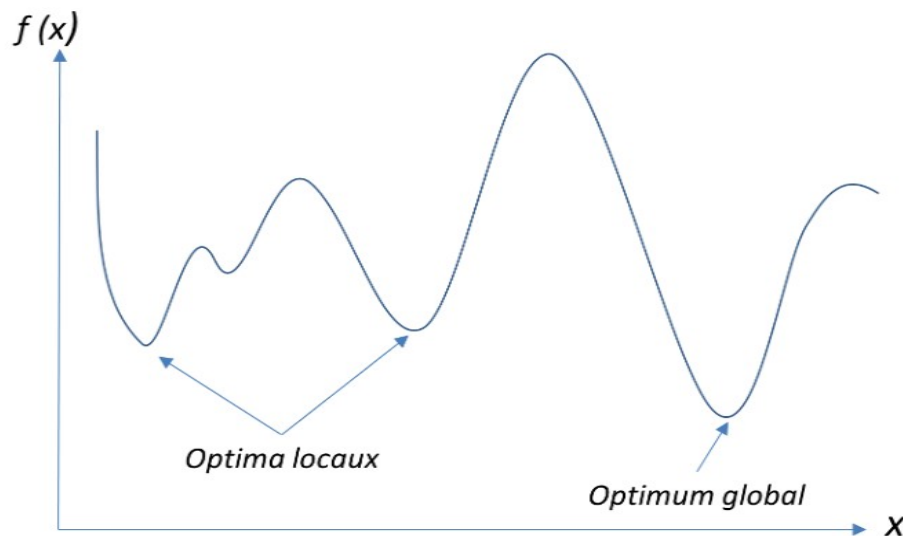


Figure II.1 Les optima locaux vs. L'optimum global.

II.3 Types de problèmes d'optimisation

Il existe plusieurs classes de problème d'optimisation [35], dont nous citons les plus connues dans ce qui suit, Figure II.2.

II.3.1 Optimisation Mono-variable et Multi-variable :

Les problèmes d'optimisation mono-variable désignent habituellement tous les problèmes dont la fonction objectif ne dépend que d'une seule variable. Les problèmes d'optimisation multi-variable rassemblent les problèmes dont la fonction objective est définie par plus d'une variable.

II.3.2 Optimisation Uni-modale et Multimodale

Un problème dont la fonction objectif ne compte qu'un seul optimum est appelé problème uni-modale. Dans ce cas, un optimum local est aussi un optimum global. A l'inverse, lorsqu'une fonction admet plusieurs optima locaux, elle est dite multimodale.

II.3.3 Optimisation Mono-Objectif et Multi-Objectif

Les problèmes d'optimisation mono-objectif regroupent les problèmes n'ayant qu'un objectif unique à optimiser. Les problèmes d'optimisation multi-objectif possèdent plus d'un objectif à optimiser.

II.3.4 Optimisation continue et discrète

A la différence de l'optimisation continue qui recherche une solution dans un ensemble infini d'objets (ex : l'ensemble des nombres réels), l'optimisation discrète regroupe les problèmes dont la solution recherchée est une combinaison d'éléments parmi un ensemble fini d'objets.

II.3.5 Optimisation avec et sans contraintes

Les problèmes d'optimisation sans contraintes sont des problèmes pour lesquels toute solutions appartenant à l'espace de recherche est considérée comme une solution faisable et acceptable. Les problèmes d'optimisation avec contraintes sont des problèmes dont une solution appartenant à l'espace de recherche ne peut être considérée comme solution faisable que si elle satisfait l'ensemble des contraintes prédéfinies.

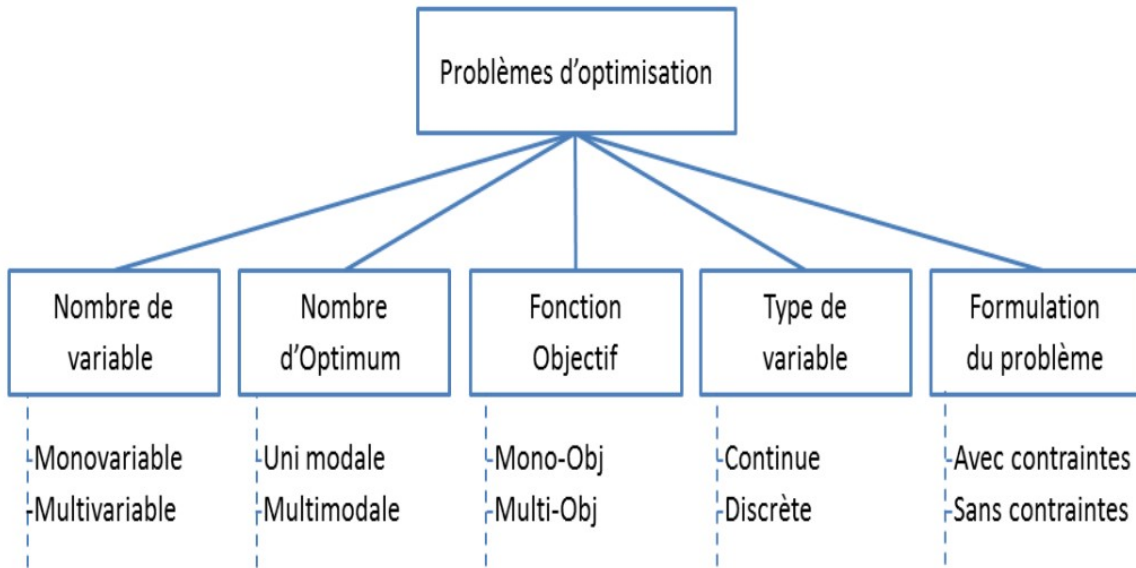


Figure II.2 Classification des problèmes d'optimisation.

II.4 Classification des méthodes d'optimisation

Malgré l'abondance de méthodes d'optimisation proposées dans la littérature, Elles peuvent être groupées en deux grandes classes. Les méthodes exactes (ou déterministes) et des méthodes approchées (non-déterministes).

II.4.1 Les méthodes exactes

Ces méthodes ont tout d'abord été introduites pour résoudre de manière exacte des problèmes particuliers comme par exemple les problèmes continus et linéaires sous contraintes linéaires (algorithme du simplexe de Dantzig). Ces méthodes ont aussi été élargies aux cas discrets et mixtes mais uniquement dans le cas linéaire. La principale qualité des méthodes déterministes est qu'elles ne nécessitent pas de point de départ. Ces méthodes permettent de bien gérer les contraintes, contrairement aux méthodes stochastiques et peuvent s'appliquer aux problèmes mixtes (variables réelles, entières et de catégories).

Elles garantissent l'obtention de la solution globale du problème. Cependant, il faut savoir que les méthodes déterministes globales restent utilisables tant que le nombre de variables ne devient pas trop important. Au delà d'une vingtaine de variables, elles atteignent leurs limites. Ces méthodes permettent d'obtenir, à la convergence, la solution exacte du problème d'optimisation considéré avec une garantie absolue. Les méthodes d'optimisation exactes les plus réussies dans la littérature sont : la méthode de Branch-and-Bound, la méthode avec retour arrière, la programmation dynamique et la programmation linéaire [34].

II.4.2 Les méthodes approchées

Les méthodes approchées ont pour but de trouver une solution admissible en temps raisonnable, mais sans garantie de l'optimalité de cette solution. L'avantage principal de ces méthodes est qu'elles peuvent s'appliquer à n'importe quelle classe de problèmes, faciles ou très difficiles. De plus, elles ont démontré leurs robustesses et efficacités face à plusieurs problèmes d'optimisation combinatoires. Elles regroupent deux classes : les méthodes heuristiques et les méthodes méta-heuristiques.

II.4.2.1 Les méthodes heuristiques spécifiques:

Le terme heuristique, dérive du grec ancien 'heuriskêin' et signifie «trouver », est un terme didactique qui signifie l'art d'inventer, de faire des découvertes..

Les méthodes heuristiques utilisent des règles empiriques simples basées sur l'expérience du problème à résoudre. Elles ne fournissent pas nécessairement une solution optimale. Il est à souligner ici qu'une méthode heuristique peut être déterministe ou stochastique.

Il y a une multitude d'heuristiques qui ont déjà été proposées dans la littérature. Nous pouvons citer des heuristiques très simples telles que les algorithmes gloutons [36, 37] ou les approches par amélioration itérative [38]. Le principe des méthodes gloutonnes est de faire une succession de choix optimaux localement, jusqu'à ce que l'on ne puisse plus améliorer la solution, et ce, sans retour en arrière possible. Le fonctionnement d'une heuristique gloutonne est similaire à celui d'un algorithme glouton exact. La différence réside dans le fait que nous n'imposons plus que la solution obtenue soit optimale, nous obtenons donc un algorithme d'approximation.

II.4.2.2 Méthodes méta-heuristique

Des heuristiques plus poussées, adaptables à un grand nombre de problèmes différents, sans changements majeurs dans l'algorithme, ont été mises au point et ont donné naissance à une nouvelle famille d'algorithmes d'optimisation stochastiques : les méthodes d'optimisation méta-heuristiques. Le terme méta-heuristique a été inventé par Fred Glover en 1986, lors de la conception de la méthode de recherche tabou. Les méthodes méta-heuristiques forment une famille d'algorithmes d'optimisation visant à résoudre des problèmes d'optimisation difficile, pour lesquels nous ne connaissons pas des méthodes classiques plus efficaces. Elles sont généralement utilisées comme des méthodes génériques pouvant optimiser une large gamme de problèmes différents, d'où le qualificatif 'méta'. Leur capacité à optimiser un problème à partir d'un nombre minimal d'informations est contrebalancée par le fait qu'elles n'offrent

aucune garantie quant à l'optimalité de la meilleure solution trouvée. Cependant, du point de vue de la recherche opérationnelle, ce constat n'est pas forcément un désavantage, puisque l'on préfère toujours une approximation de l'optimum global trouvée rapidement à une valeur exacte trouvée dans un temps prohibitif. Il existe un grand nombre de méthodes méta-heuristiques différentes, allant de la simple recherche locale à des algorithmes complexes de recherche globale. Elles sont groupées en deux sous-classes :

- des méthodes à base de solutions simple comme la méthode de recherche locale guidée, le recuit simulé et la recherche tabou, la méthode de descente, la méthode de bruitage, ...etc ;
- des méthodes à base de population de solutions comme l'algorithme génétique, l'algorithme PSO, l'algorithme de colonie des fourmis, l'algorithme évolutionnaire différentielle, ...etc,

La plupart des méthodes méta-heuristiques utilisent des processus aléatoires comme moyens de récolter de l'information et de faire face à des problèmes comme l'explosion combinatoire. Les méthodes méta-heuristiques peuvent être considérées comme des algorithmes stochastiques itératifs, où elles manipulent une ou plusieurs solutions à la recherche de l'optimum. Les itérations successives doivent permettre de passer d'une solution de mauvaise qualité à la solution optimale. L'algorithme s'arrête après avoir atteint un critère d'arrêt, consistant généralement en l'atteinte du temps d'exécution imparti ou en une précision demandée. Ces méthodes tirent leur intérêt de leur capacité à éviter les optima locaux, soit en acceptant des dégradations de la fonction objectif au cours du traitement, soit en utilisant une population de points comme méthode de recherche. Les méthodes méta-heuristiques sont souvent inspirées de processus naturels qui relèvent de la physique (l'algorithme du recuit simulé), de la biologie de l'évolution (les algorithmes génétiques) ou encore de l'éthologie (les algorithmes de colonies de fourmis ou l'optimisation par essaim particulier). Les méthodes méta-heuristiques se caractérisant par leur capacité à résoudre des problèmes très divers, elles se prêtent naturellement à des extensions pour des problèmes optimisation multi objectifs, multimodales, dynamique [39].

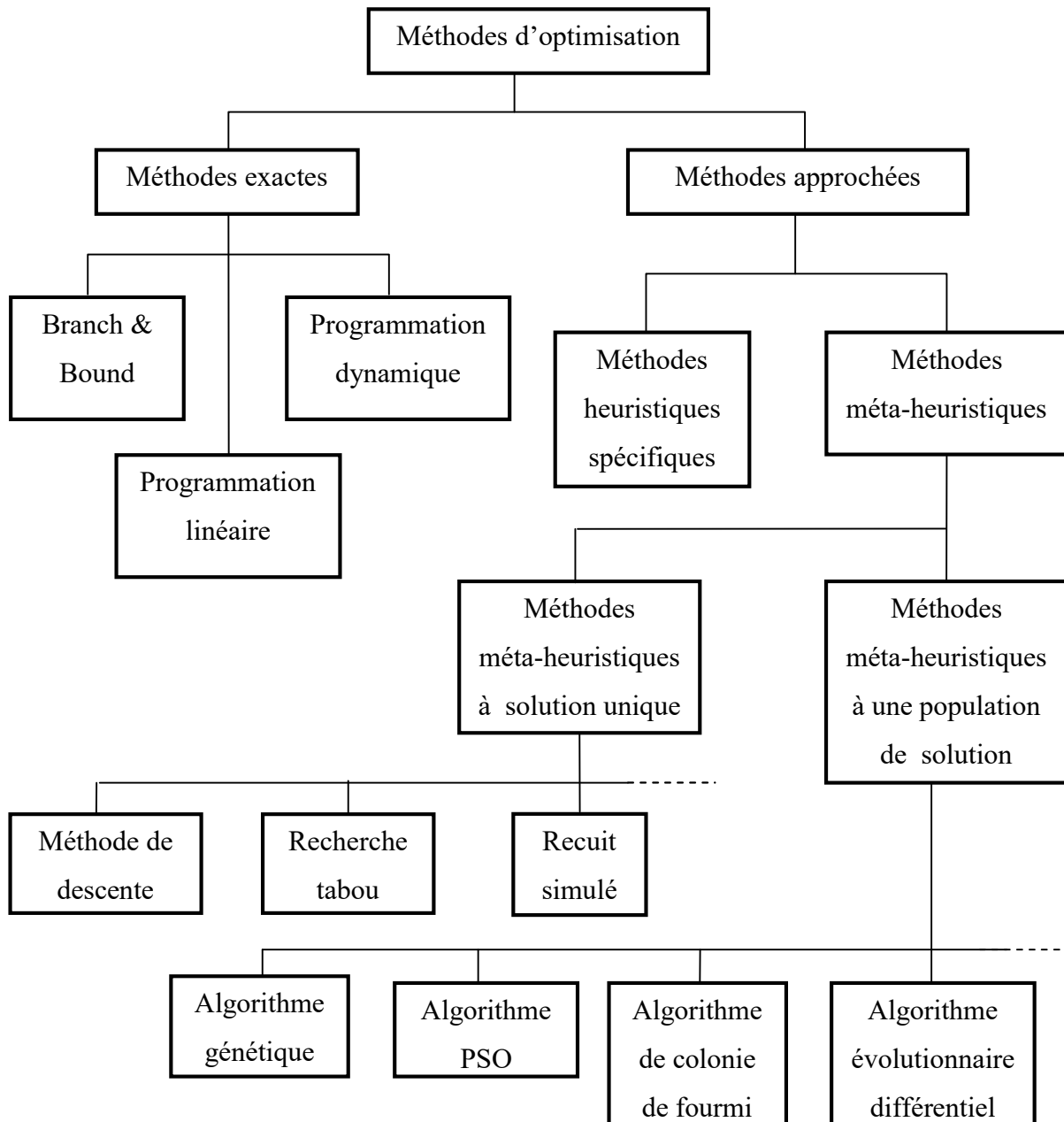


Figure II.3 Classification des méthodes d'optimisation.

II.5 Principe de méthodes méta-heuristiques

La popularité des méthodes méta-heuristiques est due principalement aux mécanismes d'exploration qui cherchent, dans l'espace des solutions, des zones prometteuses où pourrait se trouver la solution. Cependant, l'algorithme ne doit pas seulement explorer l'espace, mais aussi exploiter les candidats obtenus pour tenter de les améliorer. Exploitation et exploration ne sont possibles qu'avec un mécanisme de mémorisation qui permet de conserver certaines informations passées, Figure II.4, [40].

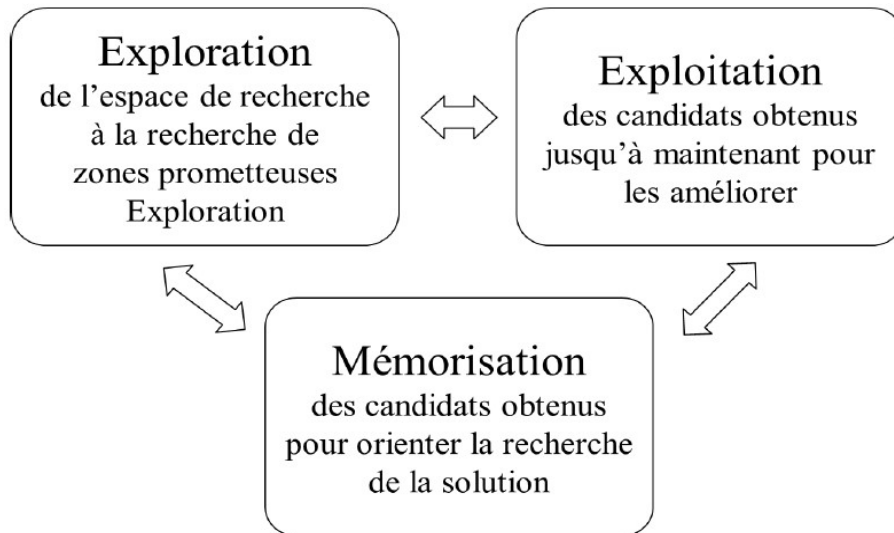


Figure II.4 Les fonctions principales des méthodes d'optimisation méta-heuristiques.

II.6 Optimisation par Essaim Particulaire (PSO):

Cet algorithme a été proposé par Russel Eberhart (ingénieur en électricité) et James Kennedy (socio-psychologue) en 1995 [33]. L'origine de cette méthode vient des observations faites lors des simulations informatiques de vols groupés d'oiseaux et de bancs de poissons REYNOLD [41], HEPPNER et GRELANDER [42]. Les individus observés sont à la recherche de sources de nourriture qui sont dispersés de façon aléatoire dans un espace de recherche, et dès lors qu'un individu localise une source de nourriture, les autres individus vont alors chercher à le reproduire. Ce comportement social basé sur l'analyse de l'environnement et du voisinage constitue alors une méthode de recherche d'optimum par l'observation des tendances des individus voisins. Chaque individu cherche à optimiser ses chances en suivant une tendance qu'il modère par ses propres vécus.

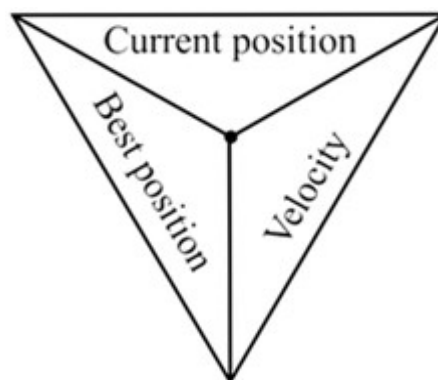


Figure II.5 Structure schématique d'une particule dans l'algorithme PSO

II.7 Principe de fonctionnement de PSO :

L'optimisation par essaim particulaire repose sur un ensemble d'individus originellement disposés de façon aléatoire et homogène, que nous appellerons dès lors des particules, qui se déplacent dans l'espace de recherche et constituent, chacune, une solution potentielle. Chaque particule est caractérisée par sa position dans l'espace de recherche, la vitesse de déplacement dans cet espace et sa meilleure solution visitée, Figure II.5, ainsi que la capacité de communiquer avec les particules constituant son entourage.

À partir de ces informations, la particule va suivre une tendance faite, d'une part, de sa volonté à retourner vers sa solution optimal ; Est d'autre part, de son mimétisme par rapport aux solutions trouvées dans son voisinage. La Figure II.6 montre l'agissement des particules d'un essaim. À partir d'optimums locaux et empiriques, l'ensemble des particules va, normalement, converger vers la solution optimale globale du problème traité. Le déplacement d'une particule est influencé par les trois composantes suivantes [43]:

- 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.

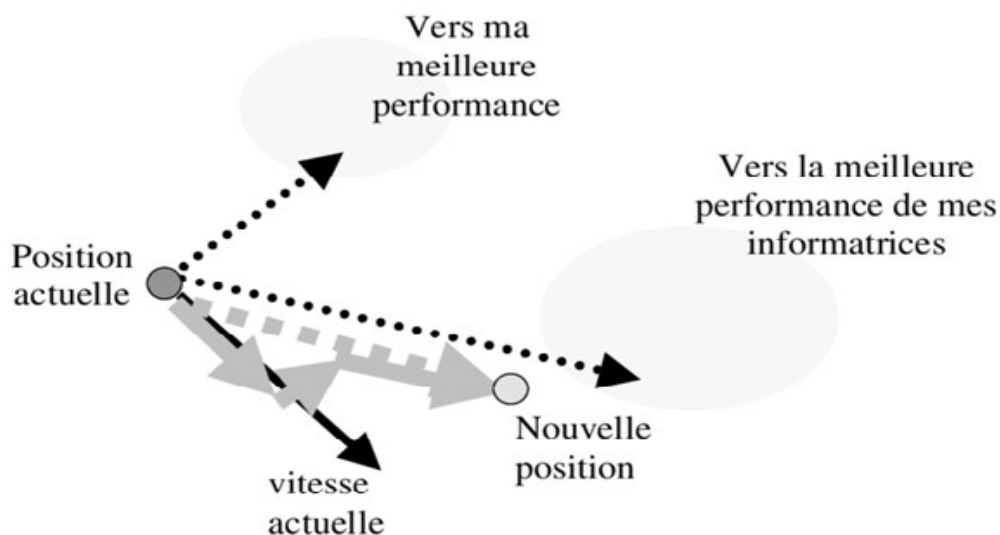


Figure II.6 schéma de principe du comportement des particules d'un essaim

Chaque particule maintient sa position, composée de la solution candidate et de son aptitude évaluée, et de son rapidité. En outre, il se souvient de la meilleure valeur de fitness qu'il a obtenue jusqu'à présent pendant le fonctionnement de l'algorithme, appelé la meilleure aptitude individuelle, et la solution candidate qui a atteint cette aptitude, désignée comme la meilleure position individuelle ou la meilleure solution individuelle de candidat. Enfin, l'algorithme PSO maintient la meilleure valeur de forme physique obtenue parmi toutes les particules de l'essaim, appelée meilleure forme physique mondiale, et la solution candidate qui a atteint cette aptitude, appelée meilleure position mondiale ou meilleur candidat mondial Solution [44]

Afin d'illustrer le contexte de la production et le développement de l'algorithme PSO, nous introduisons d'abord ici le modèle simple précoce, à savoir le modèle Boid (Bird-oid). Ce modèle est conçu pour simuler le comportement des oiseaux et est également une source directe de l'algorithme PSO.

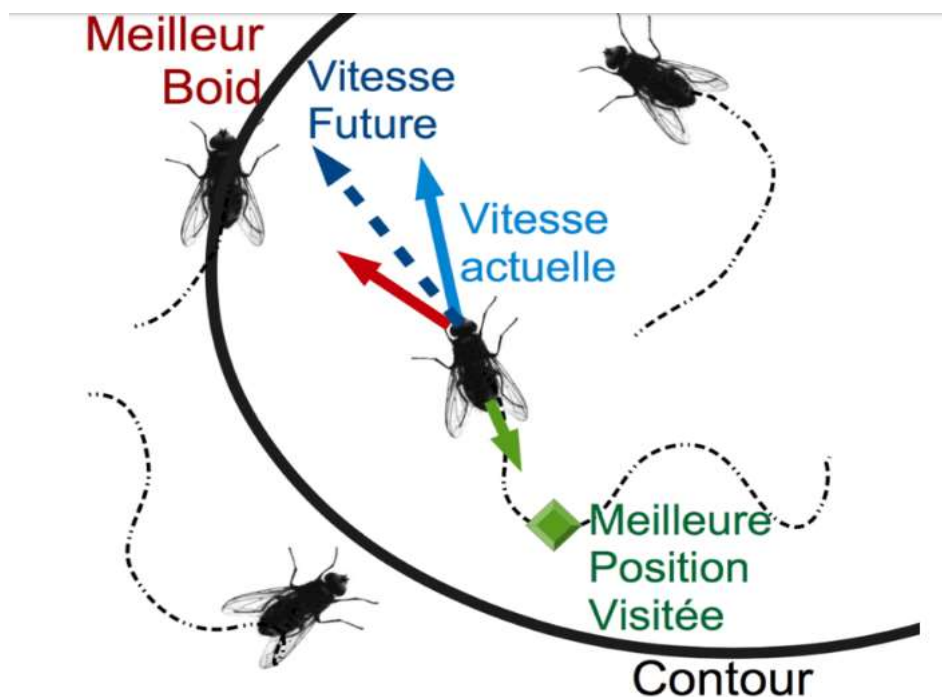


Figure II.7 Exemple de modèle de simulation du comportement de particule

Le modèle le plus simple peut être représenté comme suit. Chaque individu des oiseaux est représenté par un point dans le système de coordonnées cartésien, attribué au hasard avec la vitesse et la position initiales. Exécutez ensuite le programme conformément à «la règle de correspondance de vitesse de proximité la plus proche », de sorte qu'un individu ait la même vitesse que son voisin le plus proche. Avec l'itération se déroulant de la même manière, tous

les points auront rapidement la même vitesse. Comme ce modèle est trop simple et éloigné des cas réels, une variable aléatoire est ajoutée à l'élément de vitesse. C'est-à-dire qu'à chaque itération, en plus de rencontrer «la correspondance de vitesse de proximité la plus proche », chaque vitesse sera ajoutée avec une variable aléatoire, ce qui fait que la simulation totale se rapproche du cas réel.

Heppner a conçu un «modèle de champ de maïs » pour simuler le comportement d'alimentation d'un troupeau d'oiseaux [42]. Supposons qu'il y avait un «modèle de champ de maïs » dans l'avion, c'est-à-dire l'emplacement de la nourriture, et les oiseaux dispersés au hasard dans l'avion au début. Afin de trouver l'emplacement de la nourriture, ils se sont déplacés selon les règles suivantes.

Premièrement, nous supposons que la coordonnée de position du champ de maïs est (x_0, y_0) , et que la coordonnée de position et la coordonnée de vitesse de chaque oiseau sont (x, y) et (v_x, v_y) , respectivement. La distance entre la position actuelle et le champ de maïs est utilisée pour mesurer les performances de la position et de la vitesse actuelles. Plus la distance par rapport au «champ de maïs» est grande, meilleures sont les performances, au contraire, les performances sont pires. Supposons que chaque oiseau a la capacité de mémoire et peut mémoriser la meilleure position qu'il ait jamais atteinte, notée p_{best} . 'a' est la constante d'ajustement de la vitesse, $rand$ désigne un nombre aléatoire dans $[0,1]$, le changement de l'élément de vitesse peut être défini selon les règles suivantes:

$$\text{Si } x > p_{bestx}, v_x = v_x - rand \times a, \text{ sinon, } v_x = v_x + rand \times a.$$

$$\text{Si } y > p_{besty}, v_y = v_y - rand \times a, \text{ sinon, } v_y = v_y + rand \times a.$$

Supposons ensuite que l'essaim puisse communiquer d'une manière ou d'une autre et que chaque individu soit capable de connaître et de mémoriser le meilleur emplacement (marqué comme g_{best}) de l'essaim total jusqu'à présent. Et 'b' est la constante d'ajustement de la vitesse ; puis, une fois que l'élément de vitesse a été ajusté selon les règles ci-dessus, il doit également se mettre à jour selon les règles suivantes :

$$\text{Si } x > g_{bestx}, v_x = v_x - rand \times b, \text{ sinon, } v_x = v_x + rand \times b.$$

$$\text{Si } y > g_{besty}, v_y = v_y - rand \times b, \text{ sinon, } v_y = v_y + rand \times b.$$

En étudiant le comportement des animaux sociaux avec la théorie de la vie artificielle, pour savoir comment construire les systèmes de vie artificielle en essaim avec un comportement coopératif par ordinateur, Millonas a proposé cinq principes de base:

- Proximité : l'essaim doit être capable d'effectuer de simples calculs spatiaux et temporels.
- Qualité : l'essaim doit être capable de ressentir le changement de qualité de l'environnement et d'y réagir.
- Réponse diversifiée : l'essaim ne doit pas limiter son chemin pour obtenir les ressources dans un périmètre étroit.
- Stabilité : l'essaim ne doit pas changer de mode de comportement à chaque changement environnemental.
- Adaptabilité : l'essaim doit changer de mode de comportement lorsque ce changement est valable.

Sur cette base, l'algorithme PSO peut être résumé comme suit : l'algorithme PSO est une sorte de processus de recherche basé sur un essaim, dans lequel chaque individu est appelé une particule définie comme une solution potentielle du problème optimisé dans l'espace de recherche D-dimensionnel, et il peut mémoriser la position optimale de l'essaim et celle du sien, ainsi que la vitesse. À chaque génération, les informations sur les particules sont combinées pour ajuster la vitesse de chaque dimension, qui est utilisée pour calculer la nouvelle position de la particule. Les particules changent constamment d'état dans l'espace de recherche multidimensionnel, jusqu'à ce qu'elles atteignent l'équilibre ou l'état optimal, ou au-delà des limites de calcul. Une connexion unique entre différentes dimensions de l'espace du problème est introduite via les fonctions objectives.

De nombreuses preuves empiriques ont montré que cet algorithme est un outil d'optimisation efficace. L'organigramme de l'algorithme PSO est illustré à la Figure II.8 :

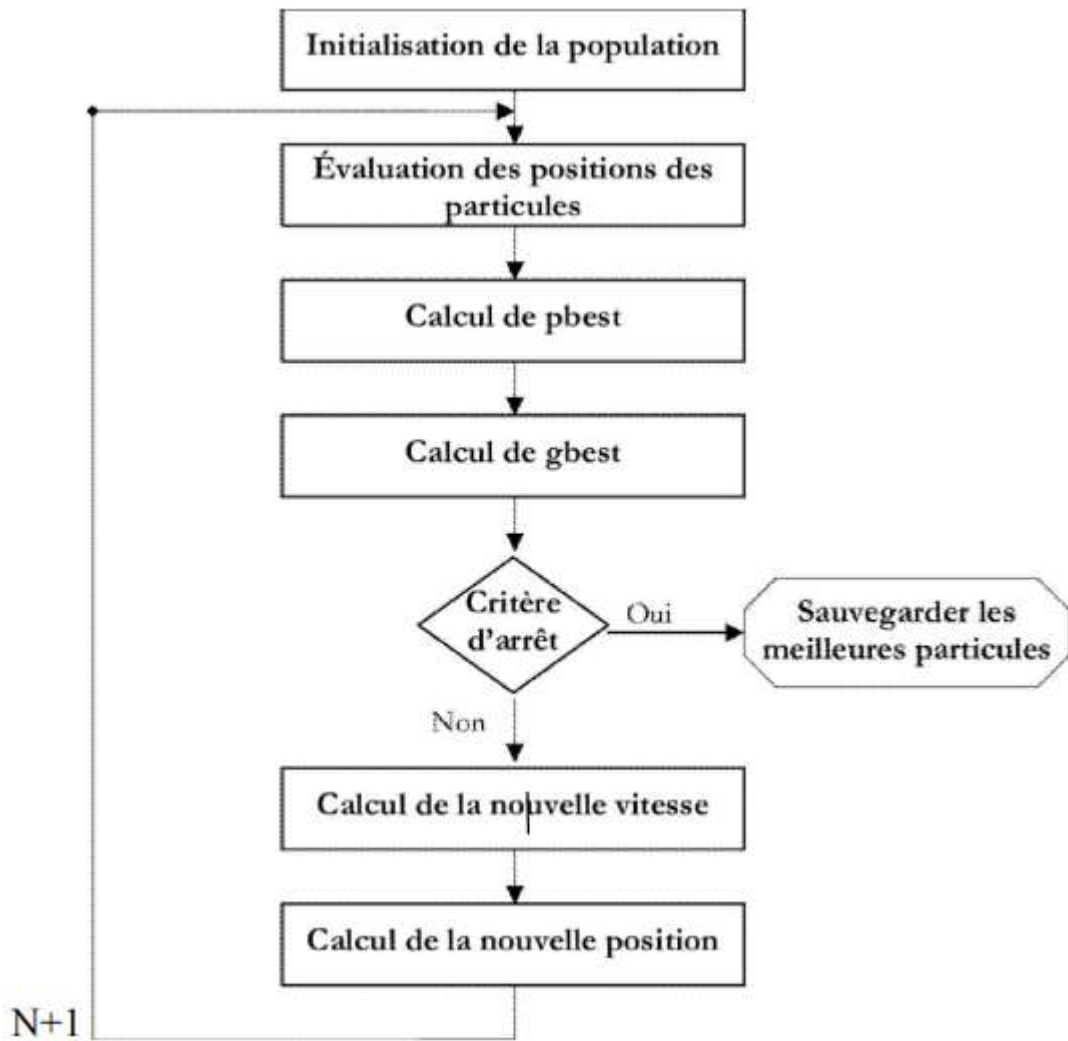


Figure II.8 Organigramme général de la méthode PSO

Dans un espace de recherche de dimension D la particule i de l'essaim est modélisée par son vecteur position $X_{ij} = (X_{i1}X_{i2} \dots X_{iD})^T$ et par son vecteur vitesse $V_{ij} = (V_{i1}V_{i2} \dots V_{iD})^T$. La qualité de sa position est déterminée par la valeur de la fonction objective (coût) en ce point. Cette particule garde en mémoire la meilleure position par laquelle elle est déjà passée, que l'on $P_{ij} = (P_{i1}P_{i2} \dots P_{iD})^T$. La meilleure position atteinte par ses particules voisines est notée $g_{ij} = (P_{i1}P_{i2} \dots P_{iD})^T$.

Le vecteur vitesse est calculé à partir de l'équation suivante [45]

$$V_{ij} = w.V_{ij}(k-1) + c_1r_1(P_{ij}(k-1) - X_{ij}(k-1)) + c_2r_2(g_{ij}(k-1) - X_{ij}(k-1)) \quad (II.3)$$

La position à l'itération de la particule est obtenue, Figure II.9, par :

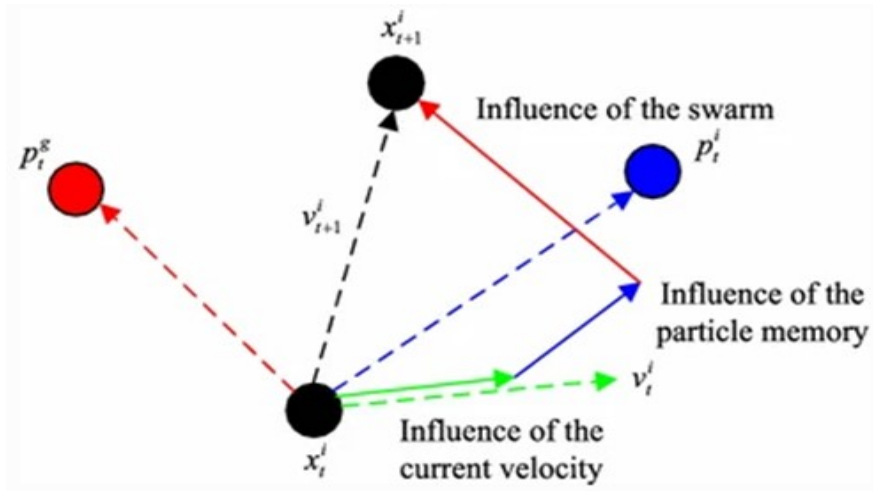


Figure II.9 Schéma d'itération des particules

$$X_{ij}(k) = X_{ij}(k - 1) + V_{ij}(k) \quad (\text{II.4})$$

Avec $i=1, 2, 3, \dots, N_p$, $j=1, 2, 3, \dots, N_d$, $k=1, 2, 3, \dots, k_{max}$ et sachant que :

N_p : nombre de particules de l'essaim ;

N_d : nombre de variables du problème (c-à-d la dimension d'une particule) ;

k_{max} : nombre maximal d'itération ;

$V_{ij}(k)$: la $j^{\text{ème}}$ composante de vitesse de la $i^{\text{ème}}$ particule de l'essaim, à la $k^{\text{ème}}$ itération

P_{ij} : la $j^{\text{ème}}$ composante de la meilleure position occupée par la $i^{\text{ème}}$ particule de l'essaim enregistrée dans les itérations précédentes (local best) ;

g_{ij} : la $j^{\text{ème}}$ composante de la meilleure position occupée par la $i^{\text{ème}}$ particule globale de l'essaim (global best) ;

$X_{ij}(k)$: $j^{\text{ème}}$ Coordonnée de la position actuelle de particule i à la $k^{\text{ème}}$ itération ;

W : constante appelée coefficient d'inertie, c_1 et c_2 : sont deux constantes appelées coefficients d'accélération, r_1 et r_2 : sont deux nombres aléatoires tirés uniformément dans $[0,1]$ à chaque itération et pour chaque dimension.

Les gammes appropriées de valeur pour c_1 et c_2 sont de 1 à 2, mais 2 est le plus approprié dans beaucoup de cas [46].

II.8 Algorithme de base de PSO :

L'algorithme de base le plus simple est donné sur la Figure II.10, [47], où g la meilleure position connue de l'essaim et $f(x)$ la fonction qui le critère de la position courante x .

Début

Pour chaque particule :

On initialise sa position

On initialise sa meilleure position p connue comme étant sa position initiale

Si $f(p) < f(g)$,

On met à jour la meilleure position de l'essaim

Fin si

On initialise la vitesse de la particule.

Fin pour

Tant que (l'on n'a pas atteint l'itération maximale ou une certaine valeur du critère) :

Pour chaque particule i :

On tire aléatoire r_1 et r_2

On met à jour la vitesse de la particule selon l'équation (3.3)

On met à jour la position x_i selon l'équation (3.4)

Si $f(x_i) < f(p_i)$,

On met à jour la meilleure position de la particule

Fin si

Si $f(p_i) < f(g)$,

On met à jour la meilleure position de l'essaim

Fin si

Fin pour

g est l'optimum.

Fin

Figure II.10 L'algorithme PSO de base.

II.9 Configuration des paramètres

II.9.1 Nombre de particules

Le nombre de particule utilisé pour la résolution du problème dépend essentiellement de deux facteurs, 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. En général le choix se fait aléatoirement.

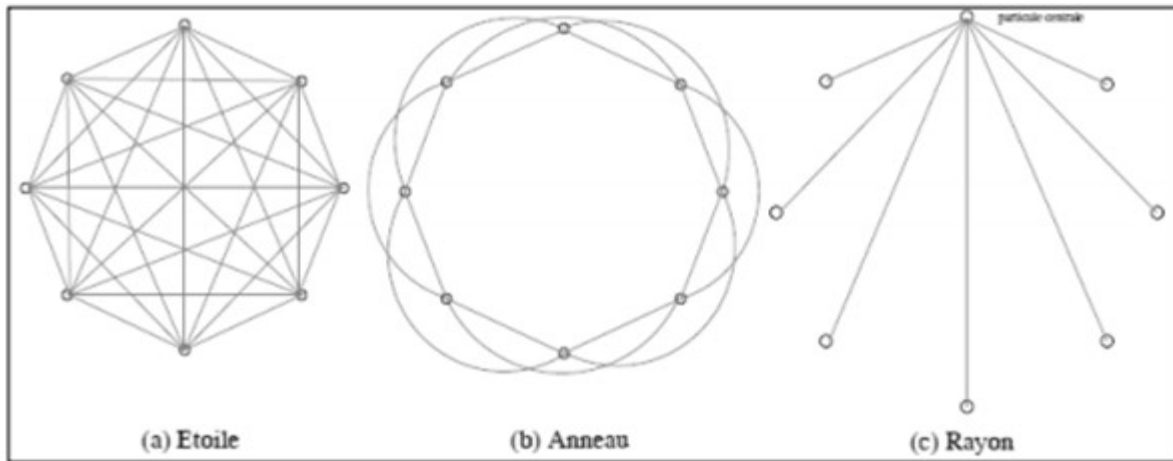


Figure II.11 Topologies du voisinage les plus utilisées.

II.9.2 Taille et topologie de voisinage

La topologie du voisinage constitue la structure du réseau social et définit avec qui chacune des particules va pouvoir communiquer. Il existe de nombreuses combinaisons dont les suivantes sont les plus utilisées, Figure II.11.

a) Topologie en étoile : chaque particule est reliée à toutes les autres, l'optimum du voisinage est l'optimum global.

b) Topologie en anneau: chaque particule est reliée à n particules ($n = 3$ en général), c'est la plus utilisée.

c) Topologie en rayon : les particules ne communiquent qu'avec une seule particule centrale.

II.9.3 Coefficients de confiance et coefficient d'inertie

Les coefficients c_1 et c_2 de l'équation (II.3) sont appelés coefficients de confiance, ils permettent de pondérer les tendances des particules à suivre leur instinct de conservation ou leur panurgisme. Les coefficients r_1 , r_2 sont des variables aléatoires évaluées à chaque

itération suivant une loi uniforme sur le domaine $[0 \ 1]$ et c_1, c_2 sont des constantes définies par la relation $c_1 + c_2 \leq 4$. Le coefficient d'inertie w permet de définir la capacité d'exploration de chaque particule en vue d'améliorer la convergence de la méthode. Fixer ce paramètre revient à trouver un compromis entre une exploration globale (> 1) et une exploration locale (< 1). Il représente l'instinct aventureux de la particule.

II.10 Amélioration des algorithmes PSO

Malgré la simplicité et la facilité de l'utilisation de l'algorithme PSO, il présente un risque de divergence ou de convergence rapide qui permet de stagner dans un optimum local, de ce fait plusieurs améliorations ont été apportées à l'algorithme de base [39].

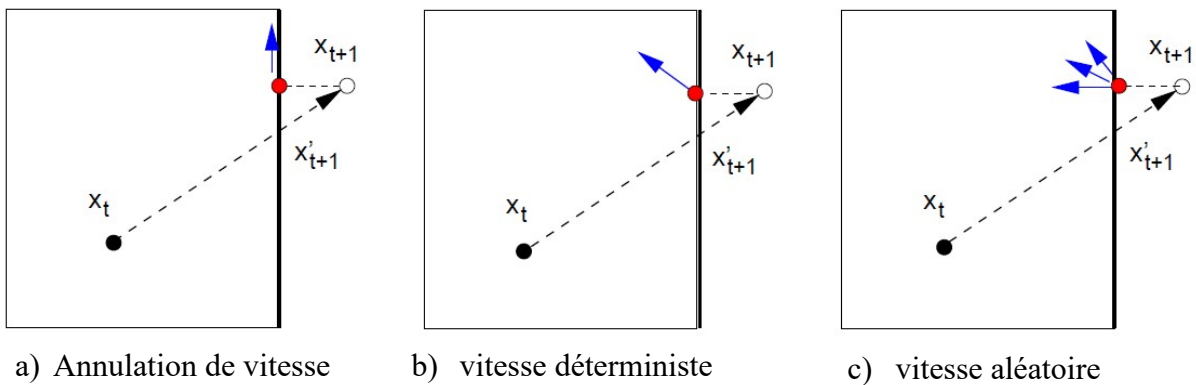


Figure II.12 Confinement de position par absorption.

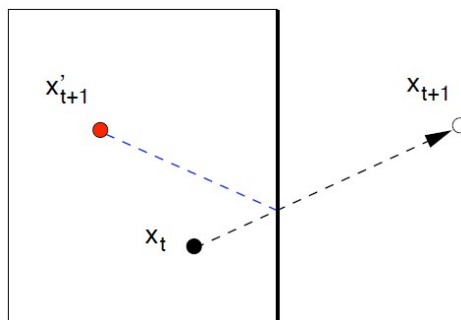


Figure II.13 Confinement de position par réflexion.

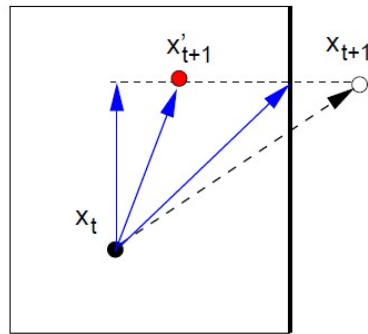


Figure II.14 Confinement de position par retour aléatoire.

II.10.1 Confinement des particules

Pour éviter que le déplacement des particules soit trop rapide, ce qui conduit à sortir de l'espace de recherche, nous pouvons introduire un nouveau paramètre V_{max} , qui permet de limiter la vitesse sur chaque dimension et ainsi de contrôler les particules. Notons que cela ne restreint pas les valeurs de x_i à l'intervalle $[V_{min}, V_{max}]$, mais limite seulement la distance maximale qu'une particule va parcourir au cours d'une itération. Cette méthode permet de contrôler la divergence de l'algorithme et de réaliser ainsi un compromis efficace entre intensification et diversification. De plus, une stratégie de confinement des particules peut être introduite. Une telle stratégie permet de ramener une particule sortie de l'espace de recherche à l'intérieur de celui-ci, [39]. Dans ce cadre, plusieurs méthodes peuvent être employées :

- La particule est laissée à l'extérieur de l'espace de recherche, mais on n'évalue pas sa fonction objective. Ainsi, elle ne pourra pas attirer les autres particules en dehors de l'espace de recherche.
- La particule est arrêtée à la frontière et les composantes associées à sa vitesse sont soit annulées ou multipliées par un coefficient tiré aléatoirement dans l'intervalle $[-1,0]$, multipliées par un coefficient bien spécifique, Figure II.12.
- La particule rebondit sur la frontière, vers l'intérieur de l'espace de recherche par symétrie par rapport à la frontière, Figure II.13.
- La particule est retournée à l'espace de recherche avec une position aléatoire, Figure II.14.

II.10.2 Coefficient de constriction

L'étude de la dynamique des particules au sein de l'essaim a conduit à la recherche de solutions pour éviter la divergence de l'algorithme, comme par exemple l'introduction du

paramètre V_{max} que nous avons vu dans le paragraphe précédent et qui permet de limiter la divergence des particules. De nombreuses études ont été menées sur la dynamique des particules concernant l'analyse des conditions de convergence de l'essaim.

La combinaison des paramètres w , c_1 et c_2 permet de régler l'équilibre entre les phases de diversification et d'intensification du processus de recherche. L'utilisation d'un coefficient de constriction χ (ou facteur de constriction) permet de mieux contrôler la divergence de l'essaim et de s'affranchir de la définition de V_{max} .

Cette variante de PSO, qui a été largement utilisée dans la littérature, est connue sous le nom de l'algorithme PSO canonique. En utilisant le coefficient de constriction, l'équation (II.3) devient :

$$V_{ij} = \chi \left(V_{ij}(k-1) + \phi_1 r_1 (P_{ij}(k-1) - X_{ij}(k-1)) + \phi_2 r_2 (g_{ij}(k-1) - X_{ij}(k-1)) \right) \quad (\text{II.5})$$

Avec

$$\chi = \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4\phi}}$$

Où $\phi = \phi_1 + \phi_2$, $\phi > 4$

Les valeurs optimales de ϕ_1 et ϕ_2 se déterminent, en effectuant de nombreux tests. En général, les valeurs utilisées sont $\phi = 4,1$ et $\phi_1 = \phi_2$, ce qui donne un coefficient $\chi = 0,7298844$.

Dans plusieurs études de recherche, Il s'est avéré que l'utilisation d'un coefficient de constriction donne généralement un meilleur taux de convergence, sans avoir à fixer de vitesse maximale.

II.10.3 Coefficient d'inertie

Le coefficient d'inertie w , introduit par Y. Shi et Eberhart [48], contrôle l'influence de la direction de la particule sur le déplacement futur. Le but de l'introduction de ce paramètre est de réaliser un équilibre entre la recherche locale (exploitation) et la recherche globale (exploration). L'intensité de l'exploration de l'espace de recherche dépend de la valeur du poids d'inertie, une grande valeur de w facilite l'exploration globale, alors qu'une petite valeur facilite l'exploration locale. Du fait de son influence sur les performances de l'algorithme PSO, le poids d'inertie a suscité un grand intérêt de la part de la communauté des chercheurs. Dans [48], les auteurs ont proposé un coefficient d'inertie dynamique qui varie au

cours du temps. Il commence par une valeur proche de 0,9 et descend linéairement pour arriver à 0,4. Cette stratégie a beaucoup amélioré les performances du PSO dans plusieurs problèmes d'optimisation. Le coefficient d'inertie w décroît linéairement avec le temps selon la formule suivante :

$$w = w_{max} - (w_{max} - w_{min}) \cdot \frac{t}{T_{max}} \quad (\text{II.6})$$

où t est l'itération courante et T_{max} est le nombre maximal d'itérations. Les coefficients w_{max} et w_{min} désignent respectivement les valeurs maximum et minimum du coefficient d'inertie. Dans une autre variante Y.Shi et Eberhart ont proposé une valeur du coefficient d'inertie choisie au hasard selon une distribution uniforme dans l'intervalle [0.5 1], [39].

II.10.4 Stratèges FIBS

Kennedy et Mendes [49] ont proposé une nouvelle manière d'utiliser la topologie Gbest, appelée FIPS (Fully Informed Particle Swarm). FIPS utilise une partie des informations de chaque voisine, au lieu de se baser seulement sur les informations de la meilleure voisine et la meilleure expérience propre à la particule. Par conséquent, l'utilisation d'une topologie entièrement connectée ne signifie pas que l'information utilisée soit seulement la meilleure solution trouvée par l'essaim. En effet, dans FIPS, elle est toujours utilisée, mais elle n'est pas la seule. Pour les algorithmes basés sur le principe de la topologie FIPS, l'information utilisée pour déplacer les particules est issue de toutes les autres particules. Ainsi, toutes les voisines contribuent à l'ajustement de la vitesse d'une particule :

$$V_i^{t+1} = x \left(v_i^t + \sum_{n=1}^{N_i} \frac{U(\theta, \varphi)(p_{nbr(n)}^t - x_i^t)}{N_i} \right) \quad (\text{II.7})$$

Où N_i est le nombre de voisins de la particule i , $nbr(n)$ est la n ième particule voisine de la particule i et φ est la constante d'accélération, qui permet de contrôler la convergence des particules [39].

II.10.5 Algorithme TRIBES

TRIBES est un algorithme d'optimisation par essaim particulaire sans paramètres de contrôle, qui a été proposé par Clerc [50]. Cet algorithme présente la particularité d'être totalement adaptatif, c'est-à-dire que tous les paramètres de contrôle sont calculés de manière autonome par l'algorithme. En effet, TRIBES est défini comme une boîte noire, pour laquelle l'utilisateur n'a plus aucun paramètre à régler. Il doit seulement définir le problème à résoudre (i.e. la fonction objectif, l'espace de recherche, les contraintes), ainsi que son critère d'arrêt.

Cependant, il est à signaler que TRIBES ne peut pas résoudre tous les problèmes. De plus, ses résultats sont probabilistes à cause de son caractère stochastique. Le but de TRIBES, d'après son auteur, est d'être efficace dans la plupart des cas et de permettre à ses utilisateurs de gagner du temps, en évitant l'étape de réglage de la méta-heuristique, [39].

Dans l'algorithme TRIBES, l'essaim particulière est divisé en plusieurs sous-essaims appelés «tribus ». Les tribus sont de tailles différentes, qui évoluent au cours du temps. Le but est d'explorer simultanément plusieurs régions de l'espace de recherche, généralement des optima locaux, avant de prendre une décision globale. Dans le but de prendre une telle décision, les tribus échangent leurs résultats tout au long du traitement. Deux types de communications sont donc à définir : la communication intra-tribu et la communication inter-tribus. Chaque tribu est composée d'un nombre variable de particules. En effet, une tribu qui peine à améliorer ses résultats génère des particules plus «exploratrices ». Les particules générées par les différentes tribus forment une nouvelle tribu, qui reste en communication avec ses génitrices. Inversement, une tribu efficace tendra à supprimer celles de ses particules qui n'ont pas contribué à sa bonne performance.

TRIBES est un algorithme compétitif, qui permet de trouver rapidement des optima locaux (très utile pour l'optimisation dynamique). Cependant, les particules ont tendance à rester dans ces optima locaux et ont du mal à en sortir.

La thèse de Yann Cooren [51] s'attache particulièrement à l'algorithme TRIBES, qui est en détail. C'est ainsi que Cooren a proposé deux idées qui permettent d'améliorer les performances de cet algorithme :

- La première idée consiste à utiliser un nouveau mode d'initialisation (initialisation régulière) pour assurer une couverture plus uniforme de l'espace de recherche par les particules. En pratique, les particules sont initialisées de manière à être les plus éloignées possible les unes des autres et les plus éloignées possible des frontières de l'espace de recherche.
- La deuxième idée consiste à utiliser une nouvelle stratégie de déplacement, basée sur une hybridation avec un algorithme à estimation de distribution, pour maintenir la diversité au sein de l'essaim, tout au long du traitement [39].

II.10.6 PSO et hybridation

Ces dernières années, l'hybridation des algorithmes a attiré l'attention de nombreux chercheurs afin d'améliorer leurs performances. L'objectif de l'hybridation est de combiner les caractéristiques de plusieurs algorithmes pour tirer profit de leurs avantages [52]. Mais l'algorithme résultant risque d'hériter également de leurs faiblesses. De plus, un algorithme résultant de l'hybridation de plusieurs algorithmes peut avoir une complexité importante. Comme pour toutes les méta-heuristiques, l'hybridation a aussi touché le domaine de PSO dans le but d'améliorer ses performances.

Dans ce qui suit, nous présentons quelques exemples d'hybridations entre PSO et d'autres algorithmes. Dans [Ange 98], Angeline a proposé la première hybridation d'un algorithme PSO. Il introduit un processus de sélection et un processus de mutation inspirés des algorithmes évolutionnaires. Le processus de sélection est utilisé pour choisir des «bonnes » particules qui vont subir une mutation, et des «mauvaises » particules qui sont éliminées.

Dans [53], une approche originale d'hybridation, qui est une combinaison de PSO et DE (DE-PSO), est proposée. Cette approche consiste à définir une stratégie de déplacement aléatoire pour accroître la capacité d'exploration et en même temps accélérer la convergence de l'algorithme, en utilisant des opérateurs de l'algorithme DE. Dans cette approche, trois stratégies de mise à jour de la particule ont été utilisées : DE Updating Strategy (DEUS), Random Updating Strategy (RUS) et PSO Updating Strategy (PSOUS). Dans [Mira 02], une hybridation entre PSO et les stratégies évolutionnaires est proposée. Les paramètres χ , ϕ_1 et ϕ_2 , ainsi que $\sim g$, sont perturbés selon une distribution gaussienne. La variance de cette distribution est déterminée à l'aide d'un processus de sélection [39].

II.10.7 Autres variantes de PSO

Hsieh et al. proposent un algorithme (EPU-PSO) basé sur une population de taille variable qui utilise trois idées principales pour améliorer les performances de PSO.

- La première idée introduit un gestionnaire de la population où, selon l'état de la recherche, l'algorithme ajoute ou supprime des particules. Cette dynamique affecte considérablement les performances et augmente la capacité à trouver l'optimum global.
- La deuxième idée est basée sur la stratégie dite stratégie de partage de solution, qui permet aux meilleures particules de partager leurs informations et de mettre à jour leurs vitesses.

- La troisième idée porte sur la technique dite searching range sharing (SRS), qui empêche les particules de tomber dans un optimum local. En outre, SRS étend l'exploration de l'espace de recherche à des zones inexplorées.

CLPSO est une variante de PSO qui utilise une nouvelle stratégie d'apprentissage (learning strategy) pour mettre à jour les vitesses des particules utilisant des informations historiques. UPSO est un algorithme de PSO qui regroupe les variantes globale et locale de PSO, en combinant leurs capacités d'exploration et d'exploitation [39].

II.10.8 Optimisation par essaim de particules à convergence rapides

L'optimisation par essaim de particules rapide (FCPSO : fast convergence particle swarm optimization en anglais) est basée sur l'équilibre entre la diversité du lieu de la particule individuelle en introduisant un nouveau paramètre qui est la dimension particulaire moyenne (PMD) de toutes les particules qui peut améliorer les performances du PSO.



Figure II.15 Calcul de la dimension particulaire moyenne pour N particules [54]

Après les mises à jour de l'essaim de particules, à partir de la génération de t à $t + 1$ en dehors de la suite de la x_{pi} et g , la particule pourrait suivre qui choisir parmi l'essaim de particules. Le troisième paramètre PMD_i de particules i -ème et l'équation de vitesse sont générés par les équations suivantes:

$$PMD_i = (x_{i1} + x_{i2} + \dots + x_{iD})/D \quad (II.8)$$

$$v_i(t + 1) = w v_i(t) + c_1 r_1 [x_{pi}(t) + x_i(t)] + c_2 r_2 [g(t) + x_i(t)] + c_3 r_3 [PMDi(t) + x_i(t)] \quad (II.9)$$

Où c_3 est un coefficient constant fixé par l'utilisateur tel que : $c_1 + c_2 + c_3 \geq 4$. r_3 est un nombre aléatoire dans la plage $[0, 1]$ tiré à chaque itération. Après l'implantation du PMDi à la formule de vitesse, x_{pi} , g et PMDi fournissent des informations à la prochaine génération et augmentent ainsi la quantité d'informations. Par conséquent, il est possible de trouver la solution optimum rapidement. La position de g est utilisée pour améliorer le taux de convergence, mais réduit la diversité de la population qui mène à un optimum local. Dans le même temps le nouveau paramètre ajouté PMDi peut déplacer les particules à un meilleur emplacement et affaiblir l'attraction de g vers les optimaux locaux [39].

II.11 Avantages et inconvénient de l'algorithme PSO :

II.11.1 Avantages

- Peut converger rapidement vers des bonnes solutions.
- Implémentations simples, avec peu de paramètres.
- Versatilité : peut résoudre beaucoup de différents problèmes.

II.11.2 Les inconvénients

- Tendance à une convergence rapide et prématurée en milieu Optimum
- Convergence lente en phase de recherche détaillée (faible capacité de recherche locale)
- Choix des paramètres d'optimisation, ces derniers sont le plus souvent spécifiques pour chaque type de problème.

II.12 Conclusion

Dans ce chapitre, nous avons présenté des généralités sur de l'optimisation et les méthodes méta-heuristiques. On a détaillé ensuite le principe de l'algorithme PSO et ses variantes ainsi que ces avantages et ces inconvénients. L'algorithme PSO est un algorithme d'optimisation stochastique basé sur la population motivée par le comportement collectif intelligent de certains animaux tels que des troupes d'oiseaux ou des bancs de poissons. Depuis sa présentation en 1995, elle a connu une multitude d'améliorations. Au fur et à mesure que les chercheurs ont découvert la technique, ils ont dérivé de nouvelles versions visant différentes demandes, développé de nouvelles applications dans une foule de domaines, publié des études

théoriques sur les effets des différents paramètres et proposé de nombreuses variantes de l'algorithme.

III Chapitre III : Algorithme PSO pour l'identification d'un système fractionnaire

III.1 Introduction

L'identification d'un système d'ordre fractionnaire est plus difficile que l'identification d'un système d'ordre entier car elle nécessite la détermination simultanée du nombre d'opérateurs fractionnaires, leurs ordres fractionnaires et leurs coefficients. La plupart des méthodes d'identification classiques ne peuvent pas s'appliquer sur ce problème complexe. Dans les deux dernières décennies, les méthodes d'optimisation méta heuristiques ont suscité beaucoup d'intérêt pour optimiser les paramètres des systèmes d'ordre fractionnaire. Parmi ces méthodes, l'algorithme PSO est considéré le plus simple avec le moins de paramètres de contrôle à spécifier. Dans ce chapitre on va investiguer l'application d'un algorithme PSO pour identifier les ordres fractionnaires et les coefficients d'un système d'ordre fractionnaire à structure connue. Ce problème d'identification est formulé alors en un problème d'optimisation des ordres fractionnaires et leurs coefficients simultanément.

III.2 Le système d'ordre fractionnaire à identifier

On suppose que la structure du système d'ordre fractionnaire à identifier est connue. La fonction de transfert de ce système est donnée par :

$$G(s) = \frac{Y(s)}{X(s)} = \frac{1}{a_4 s^{\alpha_4} + a_3 s^{\alpha_3} + a_2 s^{\alpha_2} + a_1 s^{\alpha_1} + a_0} \quad (\text{III.1})$$

Où $a_i \in [0, 100]$, $\alpha_1, \alpha_2 \in [1, 2]$, $\alpha_3 \in [2, 3]$; $\alpha_4 \in [3, 4]$;

Les simulations dans ce travail utilisent le système ayant la fonction de transfert suivante :

$$G(s) = \frac{Y(s)}{X(s)} = \frac{1}{2 s^{3.501} + 2.8 s^{2.42} + 2.6 s^{1.798} + 2.5 s^{1.31} + 1.5} \quad (\text{III.2})$$

III.3 Principe d'identification d'un système fractionnaire à base d'un algorithme PSO

La Figure III.1 montre le schéma de principe d'identification paramétrique proposée qui inclut l'algorithme PSO, le modèle à optimiser ses paramètres et la base de données d'identification qui contient l'entrée/sortie du système réel. L'algorithme PSO crée et évolue des solutions potentielles (qui sont les paramètres du système). Pour chacune de ces paramètres, la sortie de ce modèle est celui du système réel sont comparé suite a une même entrée. L'erreur de comparaison est utilisée pour évaluer la solution en question. Ensuite la valeur de la fonction

objective est transmise à l'algorithme PSO qui continue le processus d'optimisation jusqu'à la satisfaction du critère d'arrêt.

La base de données d'identification est générée à partir de la fonction de transfert de l'équation (III.2) pour une entrée échelon.

Base de données d'identification
(E/S du système réel)

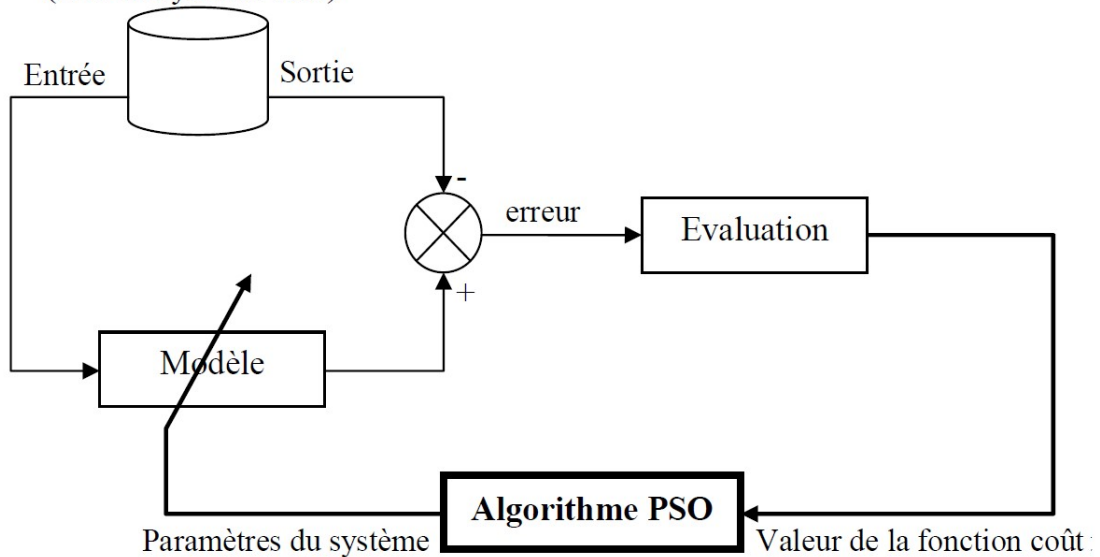


Figure III.1 Schéma de principe d'identification paramétrique par l'algorithme PSO.

III.4 L'algorithme de particule d'essaim (PSO)

L'algorithme d'optimisation utilisé est un algorithme PSO à poids d'inertie décroissant linéairement. Il manipule des éléments de traitement appelés particules. Chaque particule, a une position spécifique dans l'espace de recherche et se déplace à une certaine vitesse. La position des particules représente les paramètres du système d'ordre fractionnaire à identifier (les ordres fractionnaires et les coefficients). Dans un premier temps, l'algorithme PSO génère au hasard la position initiale et la vitesse des particules. Ces positions sont utilisées pour calculer les fonctions objectives associées, c'est-à-dire évaluer les particules. En mettant à jour la position et la vitesse de chaque particule en fonction de la meilleure position trouvée par la particule elle-même et de la meilleure position globale trouvée dans l'essaim, un nouvel ensemble de particules est obtenu puis évalué. Ce processus est répété jusqu'à la satisfaction du critère de terminaison. Dans notre travail, ce critère est l'expiration d'un nombre spécifié d'itérations. L'organigramme sur la Figure III.3 donne le principe de cet algorithme PSO.

La position y_i , la vitesse v_i de la i ème particule et le coefficient d'inertie w sont mises à jour comme suit :

$$v_i = \alpha \cdot v_i + c_1 \cdot rand \cdot (p_i - y_i) + c_2 \cdot rand \cdot (p_g - y_i) \quad (\text{III.3})$$

$$y_i = y_i + v_i \cdot \Delta t \quad (\text{III.4})$$

$$w = w_{max} - (w_{max} - w_{min}) \cdot \frac{t}{T_{max}} \quad (\text{III.5})$$

Où c_1 et c_2 sont des constantes qui mesurent respectivement les comportements cognitifs et sociaux, c'est-à-dire les tendances à se rapprocher des meilleures positions atteintes personnellement par la $i^{\text{ème}}$ particule (p_i) et l'ensemble de l'essaim (p_g), respectivement; $rand$ est un nombre aléatoire uniforme compris entre 0,0 et 1,0; Δt est la taille du pas de temps généralement égale à l'unité par souci de simplicité; w_{max} et w_{min} sont respectivement les valeurs de maximale et minimale du coefficient d'inertie w .

L'algorithme PSO utilisé peut-être décrit dans les étapes suivantes :

- ♣ Etape 1 : Initialisation aléatoire de l'essaim de particules : Les valeurs de la vitesse et la position de toutes les particules de l'essaim sont générées de façon aléatoire à l'intérieur de l'intervalle de recherche prédéterminé.
- ♣ Etape 2 : Mise à jour de la vitesse : A chaque itération, les vitesses de toutes les particules sont mises à jour selon l'équation (III.3).
- ♣ Etape 3 : confinement/cantonnement de la vitesse : Si la nouvelle valeur de vitesse de particule excède les limites prédéterminées, appliquer la stratégie de confinement de la vitesse.
- ♣ Etape 4 : Mise à jour de la position : A chaque itération, les positions de toutes les particules sont mises à jour selon l'équation (III.4).
- ♣ Etape 5 : Mise à jour du coefficient d'inertie : A chaque itération, le coefficient d'inertie est mis à jour selon une loi décroissant de l'équation (III.5).
- ♣ Etape 6 : Mise à jour des meilleures particules : A chaque itération, mettre à jour la meilleure position atteinte personnellement (p_{best}) et la meilleure position atteinte par l'essaim (g_{best}) lorsque la condition est remplie.
- Etape 7 : Vérifier le critère d'arrêt : Répéter les étapes 2 à 6 jusqu'à ce que le critère d'arrêt (nombre maximum d'itérations atteint) soit satisfait.

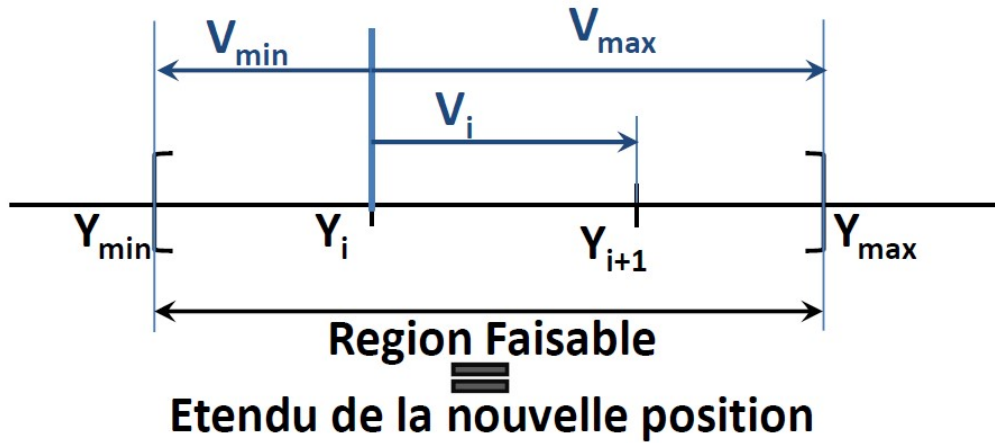


Figure III.2. Stratégie de confinement de vitesse.

III.4.1 La stratégie de confinement de la vitesse

La particularité de l'algorithme de PSO utilisé est que la vitesse de chaque particule (i) est contrôlée dans un intervalle $[V_i \text{ min}, V_i \text{ max}]$ par l'algorithme suivant :

If $(v_i < V_i \text{ min})$ ou $(v_i > V_i \text{ max})$ **alors** $v_i = \text{rand} * (V_i \text{ max} - V_i \text{ min})$

Où

$$V_i \text{ min} = X_i \text{ min} - x_i$$

$$V_i \text{ max} = X_i \text{ max} - x_i$$

Donc l'intervalle de contrôle de vitesse $[V_i \text{ min}, V_i \text{ max}]$ dépend de l'intervalle de contrôle de position $[X_i \text{ min}, X_i \text{ max}]$. De la sorte, la position de particule sera toujours confinée dans son espace de recherche, Figure III.2.

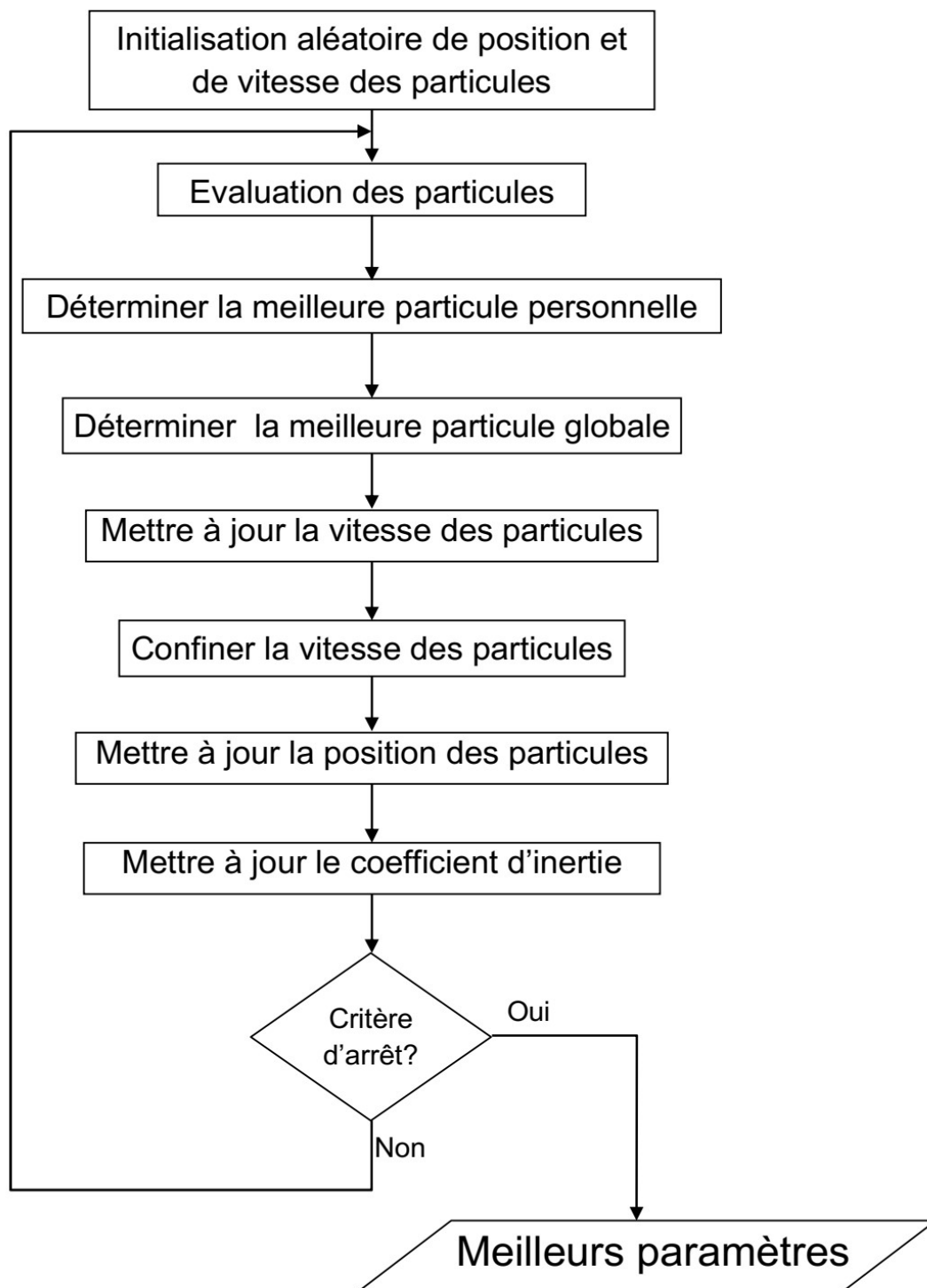


Figure III.3. Organigramme du PSO utilisé.

III.4.2 Les paramètres à identifier

L'algorithme PSO proposé a pour rôle l'identification des paramètres du système fractionnaire décrit par l'équation (III.2). Plus précisément, il optimise simultanément les ordres fractionnaires et les coefficients du dénominateur : « $a_0, a_1, a_2, a_3, a_4 ; \alpha_1, \alpha_2, \alpha_3, \alpha_4$ ».

III.4.3 L'espace de recherche

L'espace de recherche représente l'espace de variation des paramètres à optimiser. Il est déterminé par les valeurs maximales et minimales de ces paramètres. Le nombre de ces paramètres constitue la dimension de l'espace de recherche. Dans ce travail, l'espace de recherche est de dimension «9» et l'espace de recherche est défini par les intervalles de recherche suivants :

$$a_i \in [0, 100] \text{ tel que } i = 0 \dots 4$$

$$\alpha_1, \alpha_2 \in [1 \ 2] \alpha_3 \in [2 \ 3]; \alpha_4 \in [3 \ 4];$$

III.4.4 Fonction d'objectif

La fonction objectif (J) à minimiser par l'algorithme PSO proposé est la racine de la moyenne des erreurs carrées (RMSE : Root Mean square error) entre la sortie du système réel et celle du système identifié.

Elle est définie par :

$$J = \sqrt{\frac{\sum_{i=1}^N (y_i - y_r)^2}{N}}$$

Où y_i est la sortie du système identifié et y_r est la sortie du système réel. N est le nombre des échantillons.

III.5 Simulation et discussions :

Dans ce qui suit les simulations sont menées pour investiguer l'efficacité de l'algorithme PSO proposé dans l'identification des systèmes fractionnaires.

Les programmes de simulation sont réalisés par le l'outil de développement Matlab version R2018b.

Les valeurs des paramètres de contrôle de l'algorithme PSO utilisés dans la simulation sont :

- Le nombre maximum des itérations est 400.

- La taille de population est 30.
- Coefficients d'accélération $r1$ et $r2 = 2$.
- La valeur maximale du poids d'inertie (w_{max}) est 1 et sa valeur minimale (w_{min}) est 0.

A cause de la nature stochastique de l'algorithme utilisé, l'exécution de l'algorithme utilisé est répétée 10 fois avec des populations initiales différentes ce qui assure sa fiabilité. Ces exécutions fournissent donc 10 ensembles de paramètres du système fractionnaire. On choisit parmi eux l'ensemble de paramètres avec la plus faible valeur de la fonction objectif pour représenter la solution finale qui correspond aux paramètres du système fractionnaire considéré.

Le temps de simulation est de 30 sec et le pas d'échantillonnage est 0.01 sec donc le nombre d'échantillons est 3000.

III.5.1 Analyse de l'évolution de recherche :

L'évolution de la meilleure valeur de la fonction objectif au cours des itérations dans les 10 exécutions est illustrée sur la Figure III.4. La meilleure entre eux est représentée en gras.

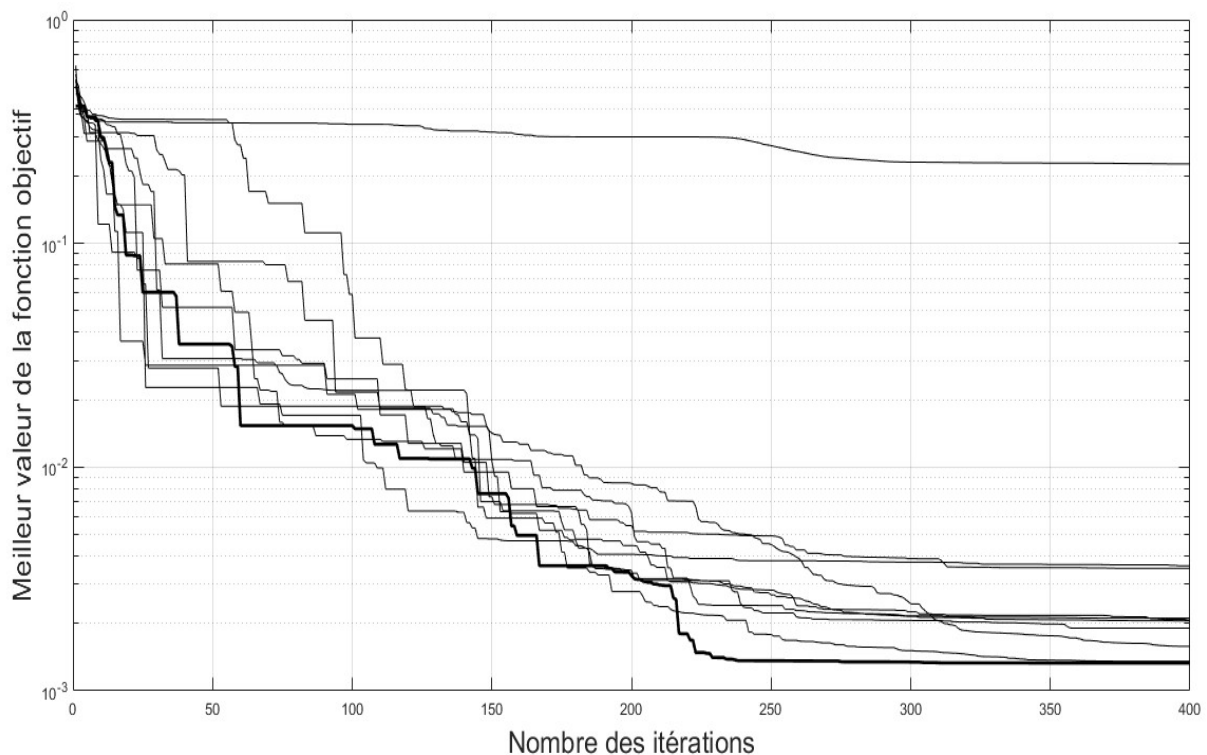


Figure III.4 courbes de convergence de l'algorithme PSO avec des essais initiaux différentes.

Sur cette figure on remarque que l'algorithme PSO a réussi à converger dans presque toutes les exécutions. Dans une seule exécution l'algorithme est tombé dans un minimum local. Ceci est dû à l'aspect multimodal complexe du problème d'identification des systèmes fractionnaires. Dans la meilleure exécution, la convergence a commencé à partir de 240^{ième} itération.

Le Tableau III-1 présente les paramètres de système optimisés par le PSO proposé pour des essais de particules initiales différents avec leurs valeurs réelles ainsi que la meilleure valeur de la fonction objectif (J) correspondante.

Tableau III-1. Les paramètres du modèle optimisés par l'algorithme PSO proposé dans les 10 exécutions.

	Valeurs réels	Valeurs optimisées									
		Run1	Run2	Run3	Run4	Run5	Run6	Run7	Run8	Run9	Run10
a_4	2	3.30026	1.01751	3.71922	1.84916	4.38616	1.50327	2.71790	7.57353	6.10526	2.89696
a_3	3.8	2.60602	2.07615	2.38286	4.52571	2.13812	5.52252	2.53458	6.86777	1.24529	0.43597
a_2	2.6	3.20097	3.01735	3.13854	1.64905	0.44860	0.29780	2.87791	9.64161	2.99049	5.97956
a_1	2.5	1.51490	4.79142	0.95470	2.45842	3.56695	3.42932	2.34222	18.02243	0.58785	2.25012
a_0	1.5	1.49955	1.49464	1.50327	1.49986	1.50165	1.49467	1.50225	1.61120	1.49101	1.49740
α_1	1.3	1.31063	1.21611	1.43092	1.25694	1.29703	1.15088	1.35528	1.00239	1.07720	1.33070
α_2	1.79	1.99978	1.99738	1.96526	1.82332	1.85019	1.87983	1.79950	1.99667	1.86625	1.33398
α_3	2.42	2.01055	2.00058	2.77493	2.63702	2.90284	2.00307	2.39627	2.32473	2.81650	2.27957
α_4	3.50	3.13615	3.28674	3.08511	3.13875	3.59290	3.15643	3.20290	3.67173	3.25277	3.73908
J	---	1.35e-03	1.57e-03	3.62e-03	1.90e-03	2.11e-03	1.33e-03	2.07e-03	2.26e-01	2.05e-03	3.52e-03

Les meilleures valeurs des paramètres du modèle qui on donné la meilleure valeur de la fonction objective sont celles de l'exécution 'Run6'. L'algorithme PSO a réussi de minimiser la valeur de la fonction objective à une valeur acceptable mais il n'a pas trouvé les valeurs exactes du modèle à cause des effets de couplage entre les ordres fractionnaires et leurs coefficients

III.5.2 Performance d'identification :

En utilisant les meilleurs paramètres optimisés du modèle du système fractionnaire, la réponse indicielle du modèle et celle synthétisée du système réel sont représenté sur la Figure III.5. Cette figure montre que les réponses du modèle optimisé par l'algorithme PSO et le système réel sont presque identique.

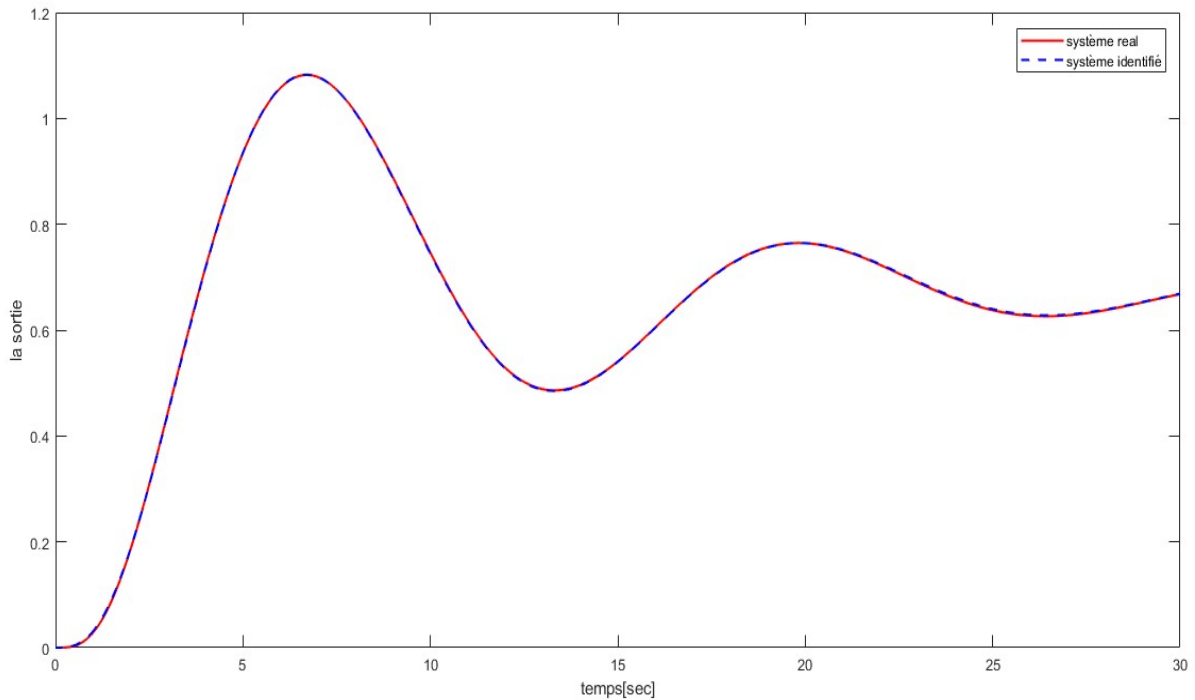


Figure III.5 La réponse indicielle du modèle identifié et du système réel.

III.5.3 Validation du modèle optimisé

Pour valider le modèle optimisé par l'algorithme PSO proposé, on a utilisé trois signaux d'entrée : un signal sinusoïdal, un signal rampe et un signal pseudo aléatoire binaire (PRBS : Pseudo Random Binary Sequence) Figure III.8.

Les sorties du modèle optimisé et celles du système réel pour ces entrée sont illustrées sur les figures : Figure III.6, Figure III.7 et Figure III.9.

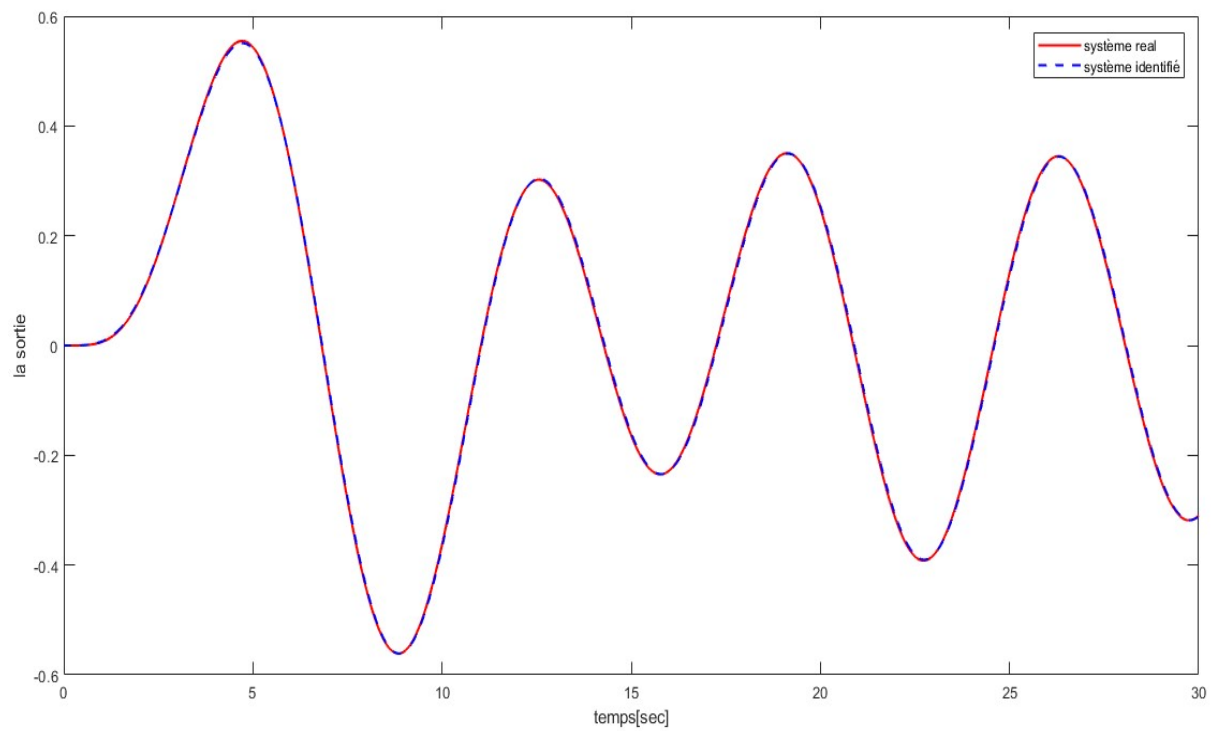


Figure III.6 La réponse du modèle identifié et du système réel à un signal d'entrée sinusoïdal.

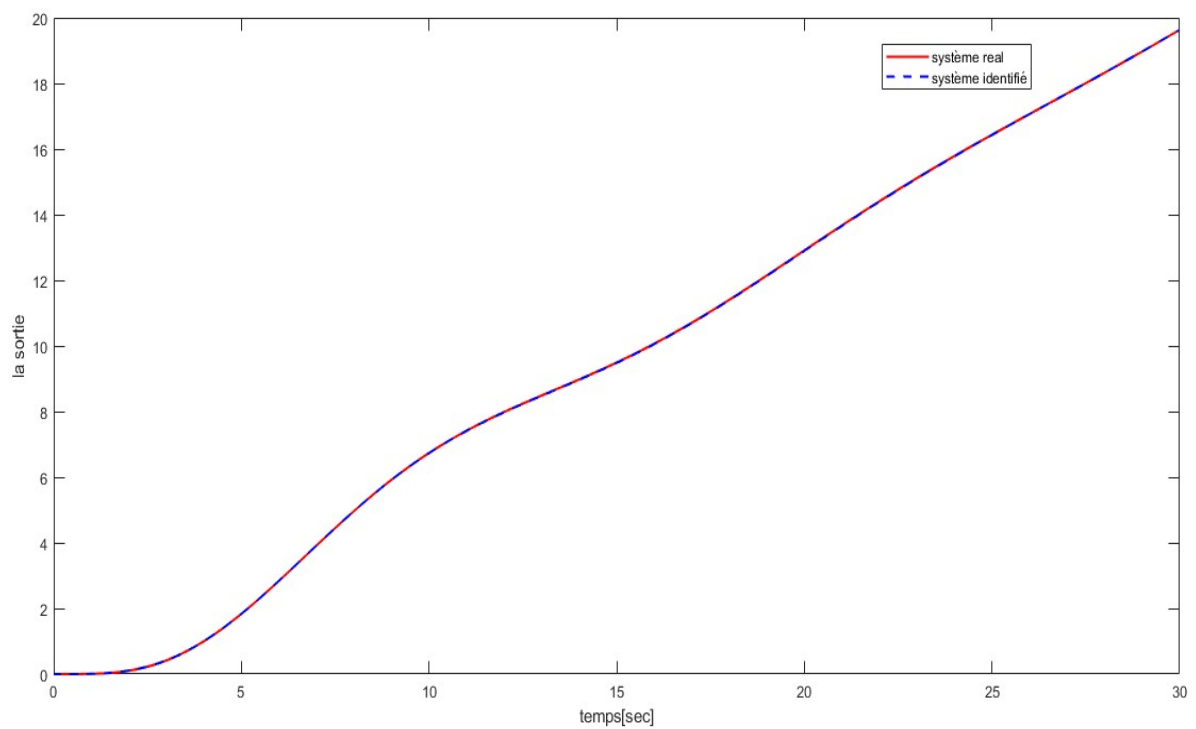


Figure III.7 La réponse du modèle identifié et du système réel à un signal d'entrée rampe.

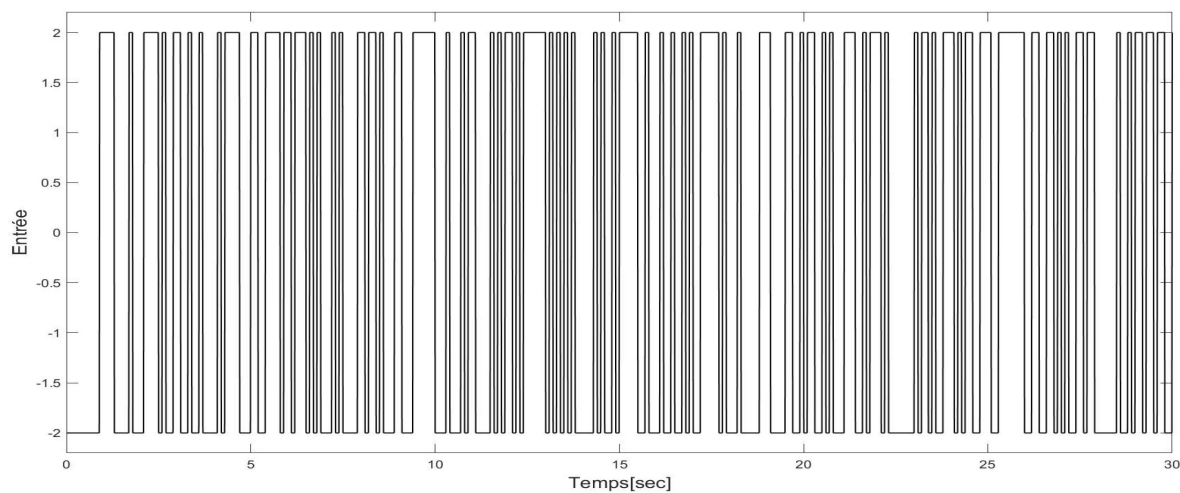


Figure III.8. Signal d'entrée PRBS.

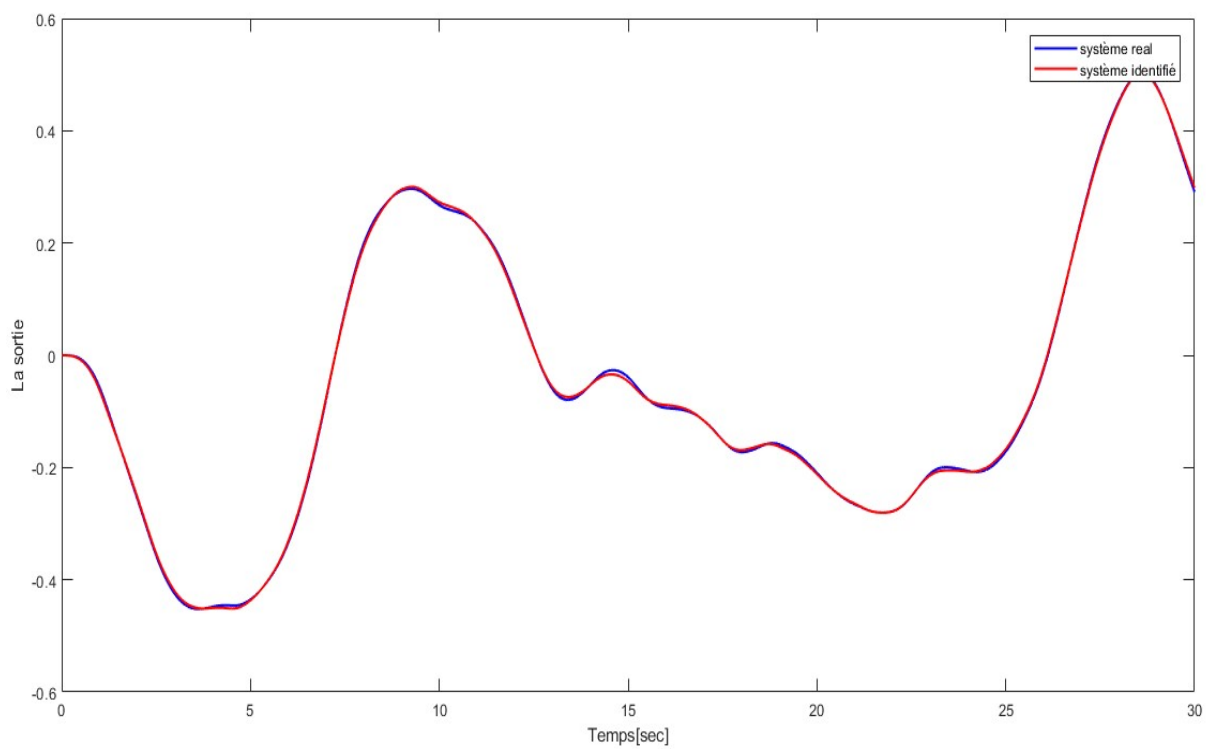


Figure III.9 La réponse du modèle identifié et du système réel à un signal d'entrée PRBS.

Interprétations :

La sortie du modèle optimisé et celle du système réel pour un signal d'entrée sinusoïdal illustrés sur la Figure III.6, montre des performances d'identification très satisfaisante. En effet, la racine de la valeur moyenne de l'erreur quadratique était $RMSE= 4.10e^{-3}$ qui est une valeur très acceptable.

La Figure III.7 compare la sortie du modèle optimisé et celle du système réel pour un signal d'entrée rampe. On peut observer que les deux réponses sont très proches et la racine de la valeur moyenne de l'erreur quadratique a atteint la valeur $RMSE=6.13e^{-3}$ qui est toujours une valeur très acceptable.

La Figure III.9 présente la sortie du modèle optimisé et celle du système réel pour un signal non canonique qui le signal pseudo aléatoire binaire –PRBS–. Malgré les différentes dynamiques excitées par ce signal d'entrée, il est clair que le modèle identifié reproduit ces dynamiques avec des erreurs très acceptables. La valeur de la racine de la moyenne de l'erreur quadratique dans ce cas est $RMSE= 3.39e^{-3}$.

On remarque que la valeur moyenne de l'erreur quadratique atteinte avec l'entrée rampe ($RMSE=6.13e^{-3}$) est relativement grande par rapport aux autres signaux. Cela est due au fait que l'amplitude du signal de sortie est élevée mais l'erreur reste très satisfaisante.

III.6 Conclusion :

Dans ce chapitre, une méthode d'identification méta heuristique basé sur l'algorithme de PSO est présentée. Les résultats des simulations menées sur un système d'ordre fractionnaires à structure connue ont montré que les performances de convergence de l'algorithme proposé est assez bonne vue la difficulté du problème traité. Nous avons montré aussi que malgré que les paramètres du modèle obtenu ne soient pas les même que le système réel, cette technique a réussie à identifier les dynamiques du système fractionnaire avec une précision très satisfaisante pour différentes entrées : entrée sinusoïdal, entrés rampe et une entrée PRBS.

Conclusion générale

Le concept de la différentiation et l'intégration non-entière appelé 'calcul fractionnaire' est une généralisation de la différentiation et l'intégration entière à un ordre réel ou complexe. Il a permis de donner un autre horizon dans la communauté scientifique et d'ingénierie pour mieux décrire, identifier et contrôler les systèmes réels.

Le travail présenté dans ce mémoire porte sur l'identification des systèmes d'ordres fractionnaires à partir des mesures d'entrée/sortie du système. Nous avons présenté une approche de l'identification paramétrique d'un système d'ordre fractionnaire à structure connue. Afin de déterminer les paramètres de ce système, nous avons fait appel à une méthode d'optimisation méta heuristique : l'algorithme d'optimisation par essaim particuliers (PSO).

Nous avons utilisé un algorithme PSO à poids d'inertie décroissant linéairement avec un mécanisme de confinement sur la vitesse seulement. Cet algorithme optimise simultanément les ordres fractionnaires et les coefficients d'un système fractionnaire à structure connue.

Les résultats obtenus en termes de convergence et de performances d'identification sont très encourageants en considérant la complexité de ce problème optimisation. Ils ont bien démontré que la technique d'identification à base de l'algorithme PSO utilisé permet de converger vers le modèle optimal et assurer les meilleures performances du système identifié.

Perspectives

On ce qui concerne la poursuite de ce travail et les perspectives envisagées, on propose les points suivants :

Pour améliorer les performances d'identification :

- Investiguer d'autres choix d'entrée pour générer la base de données.
- Utiliser d'autres méthodes d'optimisation caractérisée par une capacité de recherche globale, peut être la version TRIBE du PSO et l'algorithme évolutionnaire différentiel constitue des candidats intéressants pour ce problème.

Pour améliorer les performances de convergence de l'algorithme PSO :

- Etudier l'influence des stratégies de confinement pour faire face au problème d'effets de couplage entre les ordres et les coefficients des systèmes fractionnaires.
- Etudier l'influence d'autres lois d'adaptation de coefficient d'inertie.

Références bibliographique

- [1] B. BOUROUBA, "Contribution à la Commande Adaptative Optimale d'Ordre Fractionnaire", Département d'Electronique, Université des Frères Mentouri, Constantine-1, Algérie, Thèse de doctorat 2018.
- [2] K. Diethlem, "Efficient solution of multi-term fractional differential equations using p(ec)me methods ". Computing, vol.71, pp.305–319., 2003.
- [3] R. Koeller, "Application of fractional calculus to the theory of viscoelasticity ", Appl. Mech. vol. 51, pp. 299–307, 1984.
- [4] S. Ladaci, and A. Charef, "On fractional adaptive control ", Nonlinear Dynamics, vol. 43, no. 4, pp. 365-378., 2006.
- [5] M. Reyes-Melo, Martinez-Vega, J., Guerrero-Salazar, C., and Ortiz-Mendez, U. "Application of fractional calculus to modelling of relaxation phenomena of organic dielectric materials ". In: IEEE Int. Conf. Solid Dielectrics, pp.530–533. Toulouse, France, 2004.
- [6] A.Necaibia, Ladaci, S., Charef, A., and Loiseau, J. J, "Fractional order extremum seeking approach or maximum power point tracking of photovoltaic panels ". Front. Energy, vol. 9, no. 1, pp. 43–53, 2015.
- [7] K. Rabah, , and Ladaci, S., "Fractional adaptive sliding mode control laws for fractional-order chaotic systems synchronization ", In Int. Conf. On Sciences and Techniques of Automatic control and computer engineering (STA'2016),vol. 17, pp. 293–302., 2016.
- [8] DJOUAMBI, "Contribution à la Commande Crone ", Université Mentouri ,Constantine, Algérie, thèse de doctorat 2008.
- [9] K.B Oldham, and J. Spanier. *Fractional Calculus* , Academic Press, New York, 1974.
- [10] A.A.Kilbas, Srisvastava, H.M., and Trujillo, J.J. *Theory and Applications of Fractional Differential Equations*, Elsevier, North-Holland., 2006.
- [11] Oustaloup., *La Dérivation non Entière*, Paris : Hermès, 1995.
- [12] S Manabe, "The non-integer integral and its application to control systems".JIEE,1960.
- [13] A. Oustaloup, *Systèmes Asservis Linéaires d'Ordre Fractionnaire :Théorie et Pratique*, Editions Masson, Paris, 198.

- [14] O.P.Agrawal, "A quadratic numerical scheme for fractional optimal control ", ASMEJ Dynamic Syst., Measurement, Control, vol. 130, no. 1, pp. 011010.1–011010.6, 2008.
- [15] K. Miller. and Ross. B. *An Introduction to the Fractional Calculus and Fractional Differential Equations* . Wiley: New York, 1993.
- [16] B. BOUROUBA, "Contribution à la Commande Adaptative Optimale d'Ordre Fractionnaire", Département d'Electronique, Université des Frères Mentouri, Constantine-1, Algérie, Thèse de doctorat 2018.
- [17] D. Idiou, A. Charef et A. Djouambi, , "identification récursive des systèmes à dérivée fractionnaire ", 3rd international conference on systems and processing information, Guelma, Algeria, May 12-14, 2013.
- [18] K. Belkadi, S. Hernane, M. Benyetto "Application d'un modèle parallèle distribué de la PSO à un problème d'ordonnancement de type FSH ", conférence internationale sur le productique, Sétif , Algérie, 2007.
- [19] D. IDIOU "Modélisation et Identification des Systèmes d'Ordre Fractionnaire ", Université des Frères Mentouri, Constantine-Algerie, Thèse de doctorat 2017.
- [20] J. Sabatier, O. Agrawal & J. Machado, *Advances in fractional calculus: Theoretical Development and Applications in Physics and Engineering*. Dordrecht, the Netherlands: Springer Verlag, 2007.
- [21] Podlubny, *Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*. San Diego: Academic Press, 1998.
- [22] Petráš, I. Podlubný, P. O'Leary, Ľ. Dorčák, and B. M. Vinagre, "Analogue realizations of fractional order controllers", Nonlinear Dynamics, vol. 29, no. 1, pp. 281-296, 2002.
- [23] L. Le Lay, "Identification fréquentielle et temporelle par modèle non entier", Université de Bordeaux I, Talence, France, Thèse de doctorat 1998
- [24] O. Cois, A. Oustaloup, E. Battaglia, & J. L. Battaglia, "Non integer model from modal decomposition for time domain system identification", Proceedings of the 12th IFAC SYSID, Santa Barbara. USA, pp. 989-994, 2000.
- [25] M. Aoun, R. Malti, F. Levron and A. Oustaloup, "Synthesis of fractional Laguerre basis for system approximation", Automatica, vol. 43, no. 9, pp. 1640-1648, 2007

- [26] Maachou, "Identification par modèle non entier non linéaire : application à la modélisation de la diffusion thermique", Université de Bordeaux I, France, Thèse de doctorat 2012.
- [27] Monje, Y. Chen, B. M. Vinagre, D. Xue and V. Feliu-Batlle, *Fractional-order systems and controls: Fundamentals and applications* . London : Springer, 2010.
- [28] Valério, "Fractional Robust System Control ", Instituto Superior Tecnico, Universidad Técnica de Lisboa, Thèse de doctorat 2005.
- [29] PJ Lawrence, Rogers G. "Sequential transfer-function synthesis from measured data, Proc ". IEE, vol. 126, n. 1, pp 104–106, 1979.
- [30] R. Malti, S. Victor and A. Oustaloup, "Advances in System Identification Using Fractional Models", Journal of Computational and Nonlinear Dynamics, vol. 3, no. 2, p. 021401, 2008.
- [31] S. Victor, "Identification par modèle non entier pour la poursuite robuste de trajectoire par platitude ", Université de Bordeaux I, France, Thèse de doctorat 2010.
- [32] N. Durand. "Algorithmes Génétiques et autres méthodes d'optimisation appliqués à la gestion de trafic aérien". Optimisation et contrôle. INPT, 2004. fftel-01293722f
- [33] J. Kennedy and Eberhart, R., " Particle Swarm Optimization". In the IEEE International Joint Conference on Neural Networks, pages 1942–1948, 1995.
- [34] H. HACHIMI, "Hybridations d'algorithmes Méta-heuristiques en Optimisation Globale et Leurs Applications ", Ecole Mohammadia D'ingénieurs, Université Mohammed V AGDAL, thèse de doctorat 2013.
- [35] J. Dréo, "Adaptation de la méthode des colonies de fourmis pour l'optimisation en variables continues - Application en génie biomédical", université Paris12, thèse de doctorat 2004.
- [36] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to algorithms*, chapitre 16: Greedy Algorithms. MIT Press and McGraw-Hill, 1st Ed., 1990.
- [37] R. A. DeVore and V. N. Temlyakov. "Some remarks on greedy algorithms ". *Advances in Computational Mathematics*, vol. 5, no. 1, pp. 173–187, 1996.
- [38] V. R. Basili and A. J. Turner. "Iterative enhancement: A practical technique for software development ". *IEEE Transactions on Software Engineering*, vol. 1, no. 4, pp. 390–396, 1975.

- [39] El Dor, "Perfectionnement des algorithmes d'Optimisation par Essaim Particulaire. Applications en segmentation d'images et en électronique ", École Doctorale Mathématiques Et STIC, Université Paris-Est, thèse de doctorat 2012.
- [40] M. Ben Mazouz, Khouani A., "Optimisation paramétrique d'un classifieur neuronale par méta-heuristiques : Application données médicales ", Université Abou Bekr Belkaid, Département de Génie Bio Médical, mémoire de master 2015.
- [41] Berrim and Megdoud D. A., "Prédiction de la tension de contournement par l'approche d'optimisation PSO ", Université El Oued, Algérie, Mémoire de master 2019.
- [42] M. Clerc, M. and J. Kennedy, "The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space ", In Proceedings of the IEEE Transactions on Evolutionary Computation, 2002
- [43] Y. Cooren, "Perfectionnement d'un algorithme adaptatif d'optimisation par essaim particulaire - applications en génie médical et en électronique", l'Université Paris 12, Val de Marne, Thèse de Doctorat 2008
- [44] J. Blondin. Particle swarm optimization: A tutorial, (2009) [*Online]. http://cs.Armstrong.edu/saad/csci8100/psa_tutorial.pdf.
- [45] M.T Huynh, "Application de l'Optimisation par Essaim Particulaire à la représentation de matrice de préférence valuées ", Université Libre De Bruxelles, mémoire de magistère 2009.
- [46] R. Eberhart, Y. Shi, "comparing inertial weights and constriction factor in particle swarm optimization ", Proceeding of the International Congress on Evaluation ing Computation, San Diego, California, IEEE Service Center, Piscataway, NJ, pp. 84- 88, 2000.
- [47] M. Cherif., "Capacité d'une mémoire associative a fonction de sortie chaotique ", Université du Québec à Montréal, kanada, Mémoire de maîtrise 2010.
- [48] Y. Shi and R. C. Eberhart "Empirical study of particle swarm optimization " Proceedings of the 1999 Congress on Evolutionary Computation pp. 1945-1950, Washington, USA ,1999.
- [49] R. Mendes, J. Kennedy, and J. Neves. "The Fully Informed Particle Swarm: Simpler, Maybe Better ". IEEE Trans. Evolutionary Computation, vol. 8, no. 3, pp. 204–210, 2004.
- [50] M. Clerc. "TRIBES - Un exemple d'optimisation par essaims particuliers sans paramètres de contrôle ". In: Conférence OEP'03, Paris, France, Octobre 2003.

- [51] Y. Cooren. "Perfectionnement d'un algorithme adaptatif d'Optimisation par Essaim Particulaire. Applications en génie médical et en électronique ", Université Paris-Est Créteil, Thèse de doctorat 2008.
- [52] E. G. Talbi. "A Taxonomy of Hybrid Metaheuristics ". *Journal of Heuristics*, vol. 8, no. 5, pp. 541–564, 2002.
- [53] C. Zhang, J. Ning, S. Lu, D. Ouyang, and T. Ding. "A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization ". *Oper. Res. Lett.*, vol. 37, no. 2, pp. 117–122, March 2009.
- [54] Amaresh Saha, Sushanta Kumar Panigrahi, Sabyasachi Pattnaik. "Fast Convergence Particle Swarm Optimization for Functions Optimization ", *Procedia Technology*, vol. 4, 2012.