

« étude du bruit »
UNIVERSITÉ DES SCIENCES ET DE LA TECHNOLOGIE D' ORAN

U - S - T - O

INSTITUT D' ELECTRONIQUE

**MEMOIRE
DE
PROJET DE FIN D' ETUDES**

Pour l'obtention du

***Diplôme d' Ingénieur d' Etat
en ELECTRONIQUE***

Option : *CONTROLE*

THEME

**RECONSTRUCTIONS DE SIGNAL A PARTIR
DE LA PHASE SPECTRALE
« ETUDE DU BRUIT »**

Soutenu le : *29 Juin 1996*

Par :

***Mr BELBACHIR Ahmed Nabil
Mr KICHOU Hassane Baghdad***

Devant le Jury :

Président (e) : Mme HAMADA A.

Encadreur (s) : Mr BELBACHIR M. F.

Examineur (s) : Mr OULD MAMMAR

ANNEE UNIVERSITAIRE 1995-1996

DEDICACES

REMERCIEMENT

Au terme de ce travail de recherche que nous avons entrepris et qui a abouti au mémoire de fin d'études que nous présentons aujourd'hui, nous ne saurons oublier d'adresser nos remerciements les plus chaleureux à Monsieur **M. F. BELBACHIR** notre directeur de projet: sans ses conseils inlassables, sans le temps précieux qu'il nous a consacré, ce travail n'aurait pas vu le jour.

Que **M. BELBACHIR** soit remercié pour tout ce qu'il nous a appris: le goût du travail et de la culture, la discipline, la rigueur et la intransigeance de l'homme de science.

Faire partie de son laboratoire et de son équipe de recherche aura été pour nous un grand honneur.

Le témoignage de notre reconnaissance c'est de l'assurer que nous saurons être fidèle à son message. Puisse ce modeste travail apporter une contribution à l'expansion du jeune laboratoire qu'il dirige.

Nos remerciements iront également à Monsieur **N. BERRACHED**. Le directeur de l'Institut d'Electronique. Sous son patronage, l'Institut a acquis un rayonnement dont nous avons été les premiers bénéficiaires.

Nous remercierons également Madame et Messieurs les membres du jury dont l'immobilisation a permis la réalisation de ce travail.

Nous remercierons aussi tout nos amis qui nous ont aidé et soutenu durant la réalisation du projet: { Mohammed, Mustapha, Mustapha, Kamel, Amina, Nadia, Linda }

Nos regrets pour tout ceux que nous avons oublié.

SOMMAIRE

INTRODUCTION

CHAPITRE I 'ALGORITHMES STANDARDS'

I.1	Introduction.....	1
I.2	Théorème 1.....	1
I.2.1	Enoncé.....	1
I.2.2	Démonstration.....	2
I.3	Théorème 2.....	3
I.3.1	Enoncé.....	3
I.3.2	Démonstration.....	4
I.4	Présentation des algorithmes.....	4
I.4.1	Présentation de l'algorithme itératif.....	5
I.4.2	Présentation de l'algorithme direct.....	8
I.5	Résultats.....	11
I.5.1	Résultats de l'algorithme itératif.....	11
I.5.2	Résultats de l'algorithme direct.....	12
I.6	Exemple.....	13
I.7	Conclusion.....	15

CHAPITRE II 'ALGORITHMES UTILISANT LA SEGMENTATION'

II.1	Introduction.....	16
II.2	Présentation des algorithmes avec segmentation.....	16
II.2.1	Définition de la méthode de segmentation....	16
II.2.2	Algorithme itératif avec segmentation.....	17
II.2.3	Algorithme direct avec segmentation.....	18
II.3	Résultats.....	19
II.3.1	Résultats de l'algorithme itératif avec segmentation.....	19
II.3.2	Résultats de l'algorithme direct avec segmentation.....	20
II.4	Comparaison entre les deux algorithmes avec segmentation.....	21
II.5	Exemples.....	21
II.6	Intérêt de la segmentation dans la reconstruction du signal à partir de la phase.....	24
II.7	Conclusion.....	25
CHAPITRE III 'ETUDE DU BRUIT LORS DE LA RECONSTRUCTION'		
III.1	Introduction.....	26
III.2	Etude expérimentale.....	26
III.3	Schémas utilisés.....	27
III.4	Etude de l'effet du bruit lors de la reconstruction.....	29
III.4.1	Etude de l'algorithme direct en contexte bruité.....	29
III.4.2	Etude de l'algorithme direct avec segmentation en contexte bruité.....	31
III.4.3	Etude de l'algorithme itératif en contexte bruité.....	32

III.4.4 Etude de l'algorithme itératif avec
segmentation en contexte bruité.....34

III.5 Comparaison des résultats obtenus pour les
différents algorithmes traités.....35

III.6
Exemple.....37

III.7
Conclusion.....40

CONCLUSION

BIBLIOGRAPHIE

ABSTRACT

In this paper, we study many algorithm such as:

- *Standard algorithms: itérative and closed form ones (ALGO-ITER & ALGO-DIR) {chap I} and the algorithms using segmentation (ALGO-ITER-S & ALGO-DIR-S) {chap II}.*
- *We also use these algorithms to reconstruct sequences from degraded phase {chap III}.*

A set of experiments is used to study the effects of the additive noise on signal reconstruction. It shows that standard algorithms are less efficient than those using the segmentation.

An important result appear : the reconstruction error for an 'M' sequence samples using the segmentation by 'NS' is equivalent to those obtained with a standard algorithm for another sequence samples which length is " N=NS". This displays the advantage of the segmentation on signal reconstruction.

ääüüüüüüüü

ïãßääÇ ÍÕÑ ÚäääÇ ÇääÊääÉá Ýí ÁÚÇÏÉ Êßæïä ÇáÂÔÇÑÇÊ ÈæÇÓØÉ ÇáØæÑ ÇáØíÝí
Ýí ÝßÑÊïä :

ÇáÃæáì ÊËäËá Ýí ÌÑÇÓÉ ÇáíæÇÑÒäíÇÊ ÇáÚÇííÉ (ÇáÈÇÈ ÇáÃæá) ÇáãæìæíÉ ;
ßÇáíæÇÑÒäíÉ ÇáãÈÇÒÑÉ æ ÇáíæÇÑÒäíÉ ÇáËí ÊËäËá Úáì ÇáËßÑÇÑ ; æ
ÇáíæÇÑÒäíÇÊ ÇáËí ÊÓËÚää ÊËÓíä ÇáÃÒÇÑÇÊ (ÇáÈÇÈ ÇáËÇáí) ; ÆäÇ ÇáËÇáíÉ Ýáí
Ëáß ÇáËí ÊËäËá ÊËßæíä ÇáÃÒÇÑÇÊ Úä ØÑíß ÇáØæÑ ÇáØíÝí ÇáããÒæì ÈÇáÖæÖÇÁ
(ÇáÈÇÈ ÇáËÇáË) .

ÞääÇ ÊäÞÇÑäÉ ÇáíæÇÑÒäíÇÊ ÇáÚÇííÉ æ Êáß ÇáËÓ ÊÓËÚää ÇáËËÓíä ; æíäÇ áÐä
ÇáÁííÑÉ ÆßËÑ ÝÚÇáíÉ ää ÓÇËÞÇËáÇ ; áÇ ää äÇííÉ Óää ÇáíÓÇÈ æ áÇ ää äÇííÉ
ÇáääÇÚÉ Öí ÇáÖæÖÇÁ .

äÓËäËí Æä Êßæíä ÃÒÇÑÉ äÇ ØæääÇ " M " äÞÓäÉ Ááì ÀìÒÇÁ Øæá ÇáìÒÁ äæ " N.S "
íßæä äØÇËÞÇ áÐáß ÇááíÒä Úáíä Úä ØÑíß ÇáíæÇÑÒäíÉ ÇáÚÇííÉ ; áÃÒÇÑÉ ØæääÇ " N
= NS " .

æ ÈÇáËÇáí íËËíä ääÇ ÞíäÉ ÇáíæÇÑÒäíÇÊ ÇáËí ÊÓËÚää ÇáËËÓíä .

RESUME

On peut subdiviser les travaux contenus dans ce mémoire en deux parties: l'une concerne l'étude des algorithmes standards (Algo-Dir et Algo-Iter (CHAP.I)) et les algorithmes utilisant une segmentation (Algo-Dir-S et Algo-Iter-S (CHAP.II), l'autre partie concerne l'étude de ces algorithmes en présence du bruit blanc (CHAP.III).

Une étude comparative des performances de ces algorithmes a montré que les algorithmes utilisant une segmentation sont plus intéressants tant au point de vue calcul que du point de vue robustesse au bruit.

Un résultat important à été mis en relief: l'erreur de reconstruction pour une suite de longueur M quelconque segmentée avec NS est équivalente à celle obtenue avec un algorithme sans segmentation pour une suite de longueur N = NS.

Ceci explique l'intérêt de l'approche de la reconstruction en utilisant une segmentation.

CHAPITRE I

'ALGORITHMES STANDARDS'

I.1 INTRODUCTION

Hays et al [2] ont montré, moyennant des conditions satisfaites par la plupart des signaux rencontrés dans la pratique, que la reconstruction des signaux à partir de leur phase spectrale est possible.

Les conditions nécessaires et suffisantes de reconstruction sont données par les théorèmes (I.2, I.3). Ces résultats ont permis de mettre en oeuvre deux types d'algorithmes qui permettent la reconstruction d'un signal à partir de la phase. L'algorithme itératif (I.4.1) qui nécessite un ensemble d'itérations pour atteindre la solution alors que l'algorithme direct (I.4.2) ne nécessite qu'une seule itération qui consiste à inverser une matrice. Une discussion sur les avantages et inconvénients de ces deux algorithmes standards sera donnée.

La reconstruction de signal à partir de la phase a été intensivement utilisée pour des signaux dont la phase a été sauvegardée alors que l'amplitude spectrale a été entièrement détériorée [1].

Nous donnons un exemple illustratif, la figure I.1.a. représente un signal image dont l'amplitude a été dégradée par le bruit, alors que sa phase a été retenue intacte. La figure I.1.b. représente l'image reconstruite.

Nous remarquons que l'image ainsi reconstruite correspond à l'image d'origine.

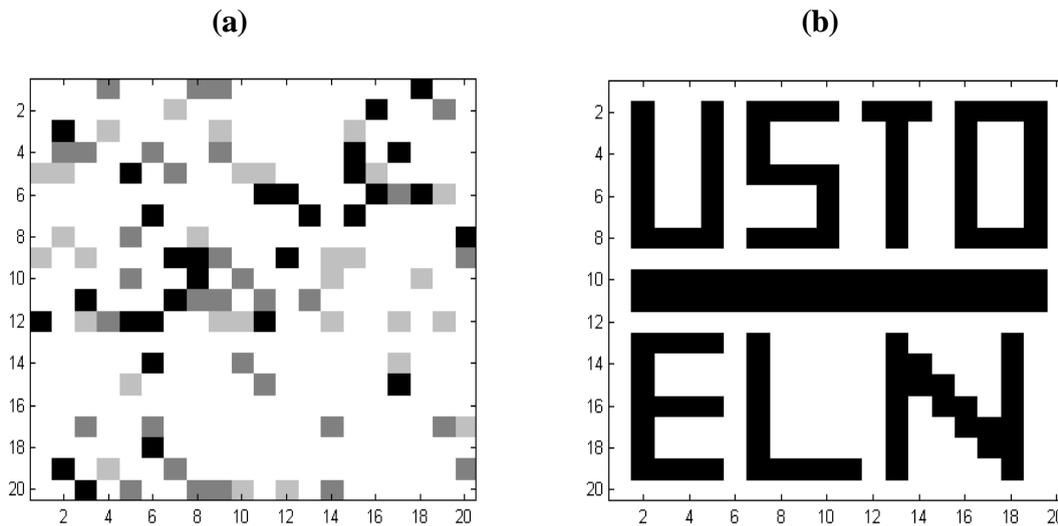


figure I.1 Reconstruction d'une image:

a) image 20x20 pixels dont l'amplitude est bruitée.

b) image reconstruite à partir de la phase spectrale de l'image a).

Nous remarquons que l'image reconstruite en b) est indiscernable de l'image d'origine.

I.2 THEOREME 1

I.2.1 Enoncé

« Soient $x(n)$ et $y(n)$ deux suites de longueur finie dont les transformées en z n'ont pas de zéros sur le cercle unité ou par paires réciproques conjuguées:

Si $\text{Ph}_x(w) = \text{Ph}_y(w), \forall w$
 alors $x(n) = \beta \cdot y(n)$, avec β constante positive; et

Si $\text{tg}(\text{Ph}_x(w)) = \text{tg}(\text{Ph}_y(w)), \forall w$
 alors $x(n) = \beta \cdot y(n)$, avec β constante réelle » [2].

avec $\text{Ph}_x(w)$ et $\text{Ph}_y(w)$ représentent respectivement les phases de $X(z)$ et $Y(z)$.

I.2.2 Démonstration

Si les transformées en z des deux suites $x(n)$ et $y(n)$ n'ont pas de zéros par paire réciproque ou sur le cercle unité, implique que:

$$\text{Si } X(z_0) = 0 \Rightarrow X(1/z_0) \neq 0,$$

et

$$\text{Si } Y(z_0) = 0 \Rightarrow Y(1/z_0) \neq 0.$$

Soit $g(n)$ une suite de longueur finie qui est définie comme suit:

$$g(n) = x(n) \otimes y(-n) \quad (\text{I-1})$$

Sa transformée en z , $G(z)$ est:

$$G(z) = X(z) \cdot Y(1/z) \quad (\text{I-2})$$

que l'on peut écrire comme suit:

$$G(z) = |G(z)| \cdot \exp(j\text{Ph}_g(w)) \quad (\text{I-3})$$

en utilisant (I-2) on a:

$$G(z) = |X(z)| \cdot |Y(1/z)| \cdot \exp(j(\text{Ph}_x(w) - \text{Ph}_y(w))) \quad (\text{I-4})$$

d'où

$$\text{Ph}_g(w) = \text{Ph}_x(w) - \text{Ph}_y(w) \quad (\text{I-5})$$

avec $\text{Ph}_x(w)$ et $\text{Ph}_y(w)$ représentent respectivement les phases de $X(z)$ et $Y(z)$.

Si $\text{Ph}_x(w) = \text{Ph}_y(w) \Rightarrow \text{Ph}_g(w) = 0$;

Si $\text{tg}(\text{Ph}_x(w)) = \text{tg}(\text{Ph}_y(w)) \Rightarrow \text{Ph}_g(w) = \{ 0 \text{ ou } \pi \}$.

En remplaçant $\text{Ph}_g(w)$ dans (I-3), on aboutit à:

$$\Rightarrow \begin{aligned} G(z) &= |G(z)| \exp(j\text{Ph}_g(w)) \\ G(z) &= |G(z)| \cos(\text{Ph}_g(w)) \end{aligned} \quad (\text{I-6})$$

A partir de (I-6), on déduit que:

$$G(z) \text{ est réelle et paire } \Rightarrow G(z) = G(1/z). \quad (\text{I-7})$$

A partir de (I-7), on trouve que $G(z)$ a des zéros par paire réciproque.

Si on pose que z_0 est un zéro pour $G(z)$:

$$\Rightarrow G(z_0) = 0 \text{ et } G(1/z_0) = 0.$$

A partir de (I-2) et (I-7):

$$G(z_0) = X(z_0) \cdot Y(1/z_0) = 0 \quad (\text{I-8})$$

et

$$G(z_0) = X(1/z_0) \cdot Y(z_0) = 0 \quad (\text{I-9})$$

Si z_0 est un zéro pour $X(z)$ et $G(z)$, ceci implique que $X(z_0)=0$, or par hypothèse $X(z)$ n'admet pas des zéros par paires réciproques ou sur le cercle unité, donc $X(1/z_0) \neq 0$, en comparant ces résultats avec les deux formules (I-8) et (I-9), on trouve que $Y(z_0) = 0$ et $Y(1/z_0) \neq 0$ (par hypothèse).

En conclusion, les zéros de $X(z)$ et $Y(z)$ sont identiques.

A partir de (I-2) et (I-7), on a:

$$\begin{aligned} G(z) &= X(z).Y(1/z) \\ G(z) &= X(1/z).Y(z) \\ \Rightarrow X(z) &= \frac{G(z)}{Y(1/z)} = \frac{X(1/z)}{Y(1/z)}.Y(z) \end{aligned}$$

$$\Rightarrow X(z) = \beta.Y(z) \quad (I-10)$$

$$x(n) = \beta.y(n) \quad (I-11)$$

En remplaçant (I-11) dans (I-1), on trouve que:

$$\begin{aligned} g(n) &= \beta.y(n) \otimes y(-n) \\ G(z) &= \beta.Y(z).Y(1/z) \end{aligned} \quad (I-12)$$

A partir de (I-3) et (I-12) on a:

$$G(z) = |G(z)| \exp(jPh_g(w)) = |\beta| \cdot |Y(z)| \cdot |Y(1/z)| \exp[j(Ph_x(w)-Ph_y(w))].$$

On remarque que:

$$\begin{aligned} \text{Si } Ph_x(w) &= Ph_y(w) \Rightarrow Ph_g(w) = Ph_x(w) - Ph_y(w) = 0 \\ &\Rightarrow \beta \text{ est une constante positive;} \\ \text{Si } tg(Ph_x(w)) &= tg(Ph_y(w)) \Rightarrow Ph_g(w) = \{0 \text{ ou } \pi\} \\ &\Rightarrow \beta \text{ est une constante réelle.} \end{aligned}$$

I.3 THEOREME 2

I.3.1 Enoncé

« Soient $x(n)$ et $y(n)$ deux suites de longueur finie qui sont nulles en dehors de l'intervalle $[0, N-1]$ dont les transformées z n'ont pas de zéros sur le cercle unité ou par paires réciproques conjuguées:

Si $Ph_x(w) = Ph_y(w)$ à $(N-1)$ fréquences distinctes dans l'intervalle $0 < w < \pi$ alors:
 $x(n) = \beta.y(n)$ avec β constante positive.

Si $\tan(Ph_x(w)) = \tan(Ph_y(w))$ à $(N-1)$ fréquences distinctes dans l'intervalle $0 < w < \pi$ alors:
 $x(n) = \beta.y(n)$ avec β constante réelle » [2].
avec $Ph_x(w)$ et $Ph_y(w)$ représentent respectivement les phases de $X(z)$ et $Y(z)$.

I.3.2 Démonstration

La validité de ce théorème suit d'une manière presque identique celle du théorème 1(I.2).
Soit une séquence de longueur finie $g(n)$ pour laquelle $g(n) = 0$ en dehors de l'intervalle de $(-N+1) \leq n \leq (N-1)$

et soit $G(w) = |G(w)| \cdot \exp(j \cdot Ph_g(w))$, la transformée de Fourier de $g(n)$ avec la phase $Ph_g(w)$ égale à 0 ou π à $(N-1)$ fréquences distinctes w_1, w_2, \dots, w_{N-1} entre 0 et π c'est-à-dire,

$$Ph_g(w_k) = 0 \text{ ou } \pi \text{ pour } k = 1, 2, \dots, N-1.$$

$$\text{avec } w_k \neq w_l \text{ pour } k \neq l \quad \text{et} \quad 0 < w_k < \pi \text{ pour } k = 1, 2, \dots, N-1.$$

donc, $G(w_k)$ est réel et:

$$IM[G(w_k)] = \sum_{n=-N+1}^{N-1} g(n) \cdot \sin(n \cdot w_k) = 0$$

$$\text{donc } \sum_{n=-N+1}^{-1} g(-n) \cdot \sin(-n \cdot w_k) + \sum_{n=1}^{N-1} g(n) \cdot \sin(n \cdot w_k) = 0$$

$$\text{puis } - \sum_{n=1}^{N-1} g(-n) \cdot \sin(n \cdot w_k) + \sum_{n=1}^{N-1} g(n) \cdot \sin(n \cdot w_k) = 0$$

$$\text{ceci est équivalent à } \left[\sum_{n=1}^{N-1} (g(n) - g(-n)) \cdot \sin(n \cdot w_k) = 0 \right] \quad \text{pour } n = 1, \dots, N-1.$$

Ceci signifie que $g(n)$ est une séquence paire.

Considérons à présent $x(n)$ et $y(n)$ satisfont les conditions de ce théorème et qui ont la même phase à $(N-1)$ fréquences distinctes entre 0 et π , définissons $g(n)$ par:

$$g(n) = x(n) \otimes y(-n).$$

puisque $g(n) = 0$ hors l'intervalle $-(N+1) \leq n \leq (N-1)$ et $Ph_g(w_k) = 0$ ou π ,
avec $w_l \neq w_k$ pour $k \neq l$ et $0 < w_k < \pi$ pour $k = 1, \dots, N-1$;
alors, $g(n)$ est une séquence paire.

De même que le théorème 1, on conclut que:

$$x(n) = \beta \cdot y(n)$$

avec β constante si les échantillons de la phase de $x(n)$ et de $y(n)$ sont égaux, et réelle si les tangentes des échantillons de la phase sont égaux. Quoique, ce théorème exige que $x(n)$ soit une séquence tout zéros.

I.4 PRESENTATION DES ALGORITHMES

Le théorème 1 (I.2) nécessite la connaissance de la phase sur tout l'intervalle $[0, \pi]$, alors que le théorème 2 (I.3) nécessite la connaissance de la phase sur un nombre fini de points sur l'intervalle $[0, \pi]$. Ce théorème a permis de mettre en oeuvre deux algorithmes, le premier qui est un « **Algorithme itératif (Algo-Iter)** » nécessite plusieurs itérations pour atteindre la solution désirée, tandis que le deuxième algorithme que nous nommerons par « **Algorithme direct (Algo-Dir)** », nécessite une seule itération pour la reconstruction du signal.

I.4.1 PRESENTATION DE L'ALGORITHME ITERATIF

I.4.1.1 Définition

Cet algorithme exploite le fait que, dans le domaine temporel, la longueur du signal est connue, et dans le domaine fréquentiel la phase du signal est connue. Et donc, l'algorithme consiste à faire des aller-retour du domaine temporel au domaine fréquentiel tout en exploitant ces deux connaissances [2].

1.4.1.2 Algorithme Itératif (Algo-Iter)

Nous avons distingué deux étapes dans l'algorithme. La partie 'Analyse' consiste à construire le vecteur phase qui sera utilisé pour la reconstruction du signal. La partie synthèse consiste à reconstruire le signal à partir de la phase obtenue dans l'étape 'Analyse'.

a) Partie analyse

On considère $x(n)$ une suite de support $[0, N-1]$.

- 1)-Lecture de la suite d'origine $x(n)$ de support $[0, N-1]$;
- 2)-Détermination du vecteur phase Ph_x par la FFT (La transformée de Fourier rapide) de $x(n)$:

$$Ph_x = \{ Ph_1, Ph_2, \dots, Ph_{N-1} \}.$$

b) Partie synthèse

La FFT à M points de $x(n)$ est:

$$X(k) = |X(k)| \cdot \exp(j \cdot Ph_x(k)) \quad \text{où} \quad M \geq 2 \cdot N.$$

Donc, une technique itérative pour reconstruire $x(n)$ à partir de N échantillons de sa phase « $Ph_x(k)$ », avec $k=0, 1, 2, \dots, M-1$, est décrite comme suit:

Etape 1

$|X_0(k)|$ est une désignation initiale de l'amplitude de la FFT de la séquence inconnue à une valeur unité comme exemple '1', donc,

$$X_1(k) = |X_0(k)| \exp(j \cdot Ph_x(k))$$

puis, on calcule la FFT inverse de « $X_1(k)$ » d'où provient $x_1(n)$. Puisque la FFT est faite sur M points, alors, $x_1(n)$ est une séquence de longueur M .

Etape 2

Fenêtrage de $x_1(n)$ pour aboutir à $y_1(n)$.

$$y_1(n) = \begin{cases} x_1(n) & 0 \leq n \leq N-1 \\ 0 & N \leq n \leq M-1 \end{cases}$$

Etape 3

Calcul de la FFT de $y_1(n)$:

$Y_1(k) = |Y_1(k)| \cdot \exp(j \cdot Ph_y(k))$ sur M points et prendre l'amplitude $|Y_1(k)|$ qu'on considère comme une nouvelle estimation de $|X(k)|$ et on associe aussi à cette amplitude la phase exacte de $x(n)$, donc on aura:

$$X_2(k) = |Y_1(k)| \cdot \exp(j \cdot Ph_x(k))$$

De là, une nouvelle estimation $x_2(n)$ est obtenue à partir de la FFT inverse de « $X_2(k)$ ».
Une application répétitive de l'étape 2 et 3 définit l'itération.

Une illustration de cet algorithme itératif utilisant la TF et la TFI est décrite dans la figure I.2.

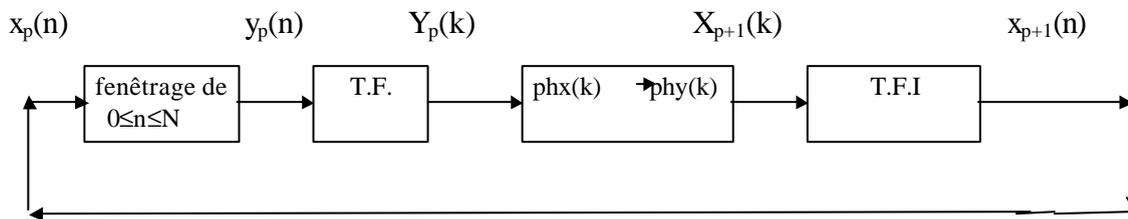


Figure I.2. schéma bloc de l'algorithme itératif.

c) Convergence de l'Algorithme

Dans cette procédure itérative, l'erreur quadratique totale entre $x(n)$ et son estimation $x_p(n)$ diminue après chaque itération. Soit « $x_p(n)$ » l'estimation après p ème itération, alors, l'erreur est définie comme suit:

$$E_p = \sum_{n=0}^{M-1} \{x(n) - x_p(n)\}^2$$

D'après le théorème de PARSEVAL on a:

$$E_p = 1/M \cdot \sum_{k=0}^{M-1} |X(k) - X_p(k)|^2.$$

puisque $X(k)$ et $X_p(k)$ ont la même phase donc:

$$E_p = 1/M \cdot \sum_{k=0}^{M-1} [|X(k)| - |X_p(k)|]^2$$

$$E_p = 1/M \cdot \sum_{k=0}^{M-1} [|X(k)| - |Y_{p-1}(k)|]^2.$$

En utilisant l'inégalité de triangle pour la différence des vecteurs si $Ph_x(k) = Ph_{p-1}(k)$ où « $Ph_{p-1}(k)$ » est la phase de $Y_{p-1}(k)$ donc on aura :

$$E_p \leq 1/M \cdot \sum_{k=0}^{M-1} |X(k) - Y_{p-1}(k)|^2.$$

En utilisant encore le théorème de PARSEVAL on aura :

$$E_p \leq 1/M \cdot \sum_{n=0}^{M-1} \{x(n) - y_{p-1}(n)\}^2$$

puisque $y_{p-1}(n) = x_{p-1}(n)$ pour $0 \leq n \leq N-1$
 $y_{p-1}(n) = x(n) = 0$ pour $N \leq n \leq M-1$.

alors:

$$\sum_{n=0}^{M-1} \{x(n) - y_{p-1}(n)\}^2 \leq \sum_{n=0}^{M-1} \{x(n) - x_{p-1}(n)\}^2 = E_{p-1}$$

Avec cette égalité si et seulement si $y_{p-1}(n) = x_{p-1}(n)$ cependant:

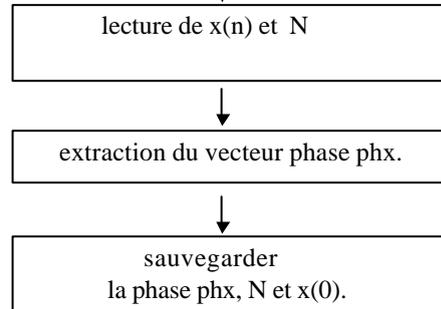
$$E_p \leq E_{p-1}$$

(I-13)

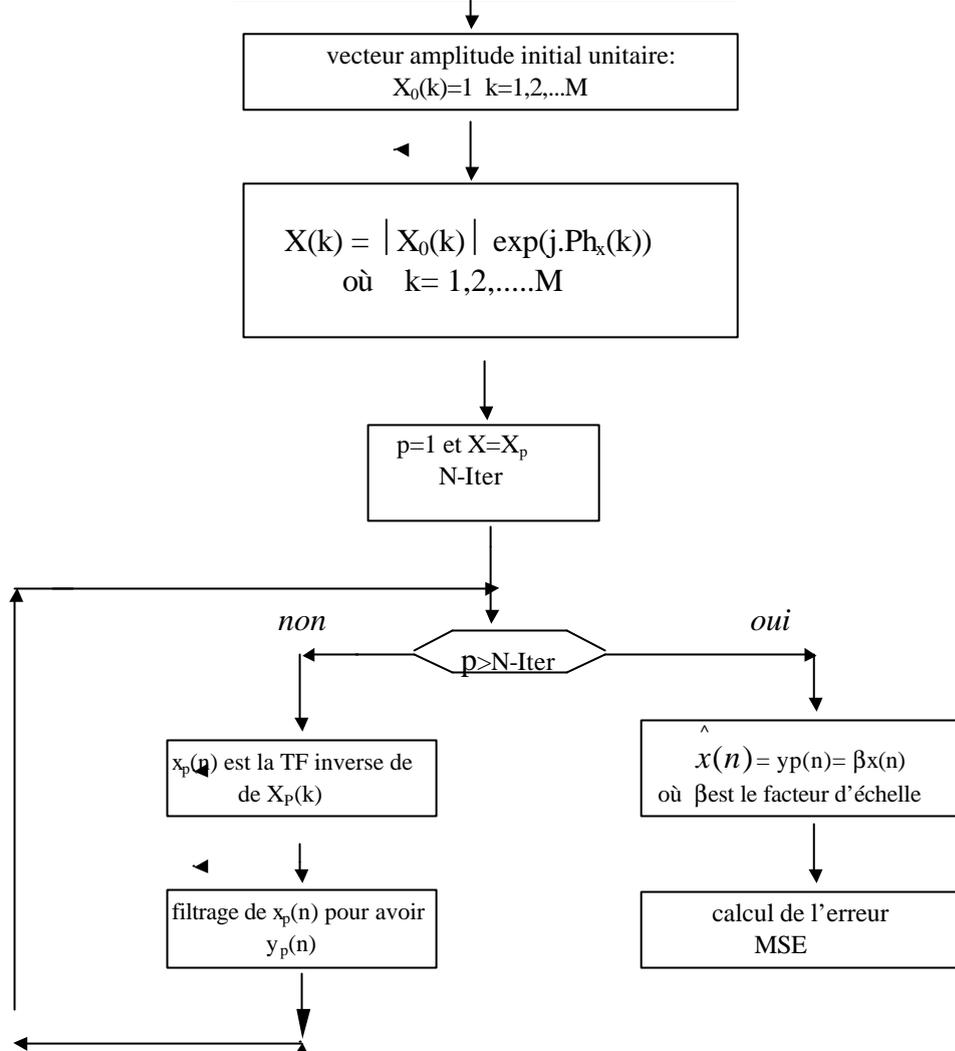
L'expression (I.13) montre que l'algorithme converge mieux lorsque le nombre d'itérations augmente.

d) L'Organigramme

'l'Algorithme Itératif 'partie analyse'



'l'Algorithme Itératif 'partie synthèse'



avec:

N-Iter: le nombre d'itérations

MSE: l'erreur quadratique moyenne défini comme suit,



$$\text{MSE} = \frac{1}{N} * \sum_0^{N-1} (x(n) - \hat{x}(n))^2 \quad (1-14)$$

$\hat{x}(n)$: la suite reconstruite par l'algorithme itératif.

I.4.2 PRESENTATION DE L'ALGORITHME DIRECT (Algo-Dir)

I.4.2.1 Définition

Cet algorithme permet la reconstruction du signal en une seule itération. Considérons une suite réelle $x(n)$ de support $[0, N-1]$.

$$\begin{aligned} X(\omega) &= TF(x(n)) = \sum_{n=0}^{N-1} x(n) \exp(-jn\omega) \\ \Rightarrow X(\omega) &= \sum_{n=0}^{N-1} x(n) \cos(-n\omega) + \sum_{n=0}^{N-1} x(n) \sin(-n\omega) \\ X(\omega) &= \sum_{n=0}^{N-1} x(n) \cos(n\omega) - \sum_{n=0}^{N-1} x(n) \sin(n\omega) \\ \Rightarrow \text{tg}[Ph_x(\omega)] &= -\frac{\sum_{n=0}^{N-1} x(n) \sin(n\omega)}{\sum_{n=0}^{N-1} x(n) \cos(n\omega)} \quad \text{avec } Ph_x(\omega): \text{ la phase de } TF(x(n))[3]. \end{aligned}$$

I.4.2.2 Les Etapes Principales de l'Algorithme

Etape 1

$$\text{tg}[Ph_x(\omega)] = -\frac{\sum_{n=0}^{N-1} x(n) \sin(n\omega)}{\sum_{n=0}^{N-1} x(n) \cos(n\omega)} \quad [2] \quad (I-15)$$

Etape 2

D'après le théorème, on échantillonne $\text{tg}[Ph_x(\omega)]$ en $N-1$ échantillons à $N-1$ fréquences distinctes

$\omega_1, \omega_2, \dots, \omega_{N-1}$.

avec:

$$0 < \omega_k < \pi \quad (I-16)$$

et

$$k=1, 2, \dots, N-1 \quad (I-17)$$

on sait que:

$$\omega_k = 2\pi k \Delta f \quad (I-18)$$

donc pour arriver à (I-16), Δf est choisie comme suit:

$$\Rightarrow 0 < k < \frac{1}{2\Delta f} \quad \Rightarrow \quad 1 \leq k \leq \frac{1}{2\Delta f} - 1 \quad (\text{I-19})$$

on comparant l'inégalité (I-17) et (I-19) on trouve que:

$$\begin{aligned} \Delta f &= \frac{1}{2N} \\ \Rightarrow \mathbf{w}_k &= 2\mathbf{p} \cdot k\Delta f = 2\mathbf{p} \cdot k \frac{1}{2N} \\ \mathbf{w}_k &= \frac{\mathbf{p} \cdot k}{N} \end{aligned} \quad (\text{I-20})$$

avec: $k=1, \dots, N-1$

Etape 3

A partir de l'expression (I-15), on déduit un système d'équation linéaire qu'on peut résoudre à un facteur multiplicatif près avec:

$$-\pi \leq \text{Ph}_x(\omega_k) \leq \pi$$

on a:

$$\begin{aligned} \text{tg}(\text{Ph}_x(\mathbf{w}_k)) &= \frac{\sum_{n=0}^{N-1} x(n) \sin(n\mathbf{w}_k)}{\sum_{n=0}^{N-1} x(n) \cos(n\mathbf{w}_k)} \\ \Rightarrow \sin(\text{Ph}_x(\mathbf{w}_k)) \sum_{n=0}^{N-1} x(n) \cos(n\mathbf{w}_k) &= -\cos(\text{Ph}_x(\mathbf{w}_k)) \sum_{n=0}^{N-1} x(n) \sin(n\mathbf{w}_k) \\ \Rightarrow \sum_{n=0}^{N-1} [\sin(\text{Ph}_x(\mathbf{w}_k)) x(n) \cos(n\mathbf{w}_k) + \cos(\text{Ph}_x(\mathbf{w}_k)) x(n) \sin(n\mathbf{w}_k)] &= 0 \\ \Rightarrow \sum_{n=0}^{N-1} x(n) [\cos(n\mathbf{w}_k) \sin(\text{Ph}_x(\mathbf{w}_k)) + \sin(n\mathbf{w}_k) \cos(\text{Ph}_x(\mathbf{w}_k))] &= 0 \\ \sum_{n=0}^{N-1} x(n) \sin[(n\mathbf{w}_k) + \text{Ph}_x(\mathbf{w}_k)] &= 0 \end{aligned} \quad (\text{I-21})$$

avec $k=1, \dots, N-1$.

Le système (I-21) peut s'écrire sous cette forme:

$$\sum_{n=1}^{N-1} x(n) \sin[(n\mathbf{w}_k) + \text{Ph}_x(\mathbf{w}_k)] = -x(0) \sin(\text{Ph}_x(\mathbf{w}_k)) \quad (\text{I-22})$$

L'équation (I-22) forme un système d'équation linéaire de (N-1) équations à (N-1) inconnues (si x(0) est considéré connu).

La forme matricielle de ce système est comme suit:

$$s.x = x(0).b \quad (\text{I-23})$$

s: c'est une matrice de dimension [N-1,N-1];
b: c'est un vecteur de longueur N-1;
x: le vecteur inconnu correspondant au signal reconstruit;
x(0): le premier élément non nul de la suite d'origine.

Dans la résolution du système (I-23), on considère deux cas:

a) Dans le cas où x(0) est connu, on peut résoudre ce système d'équation sans problème

$$\text{avec:} \quad x = x(0).s^{-1}.b \quad (\text{I-24})$$

b) Dans le cas contraire, on aura un système de (N-1) équations a N inconnues, alors, on pose :

$$\hat{x} = \frac{x}{x(0)} = s^{-1}.b. \quad (\text{I-25})$$

Dans ce cas la résolution de ce système donne une suite reconstruite à un facteur multiplicatif près.

- Les élément de la matrice 's' et du vecteur 'b' son calculés comme suit:

$$s(k, n) = \sin(n\mathbf{w}_k + \text{Ph}_x(\mathbf{w}_k)) \quad (\text{I-26})$$

$$b(k) = -\sin(\text{Ph}_x(\mathbf{w}_k)) \quad (\text{I-27})$$

$$\text{avec} \quad \mathbf{w}_k = \frac{\mathbf{p}.k}{N}, \quad \text{pour } n=1,\dots,N-1 \text{ et } k=1,\dots,N-1. \quad (\text{I-28})$$

I.4.2.3 Mise en Oeuvre de L'Algorithme Direct (Algo-Dir)

Pour cet Algorithme, nous avons distingué deux étapes.

a) Partie analyse

Il s'agit d'extraire la phase du signal x(n). La démarche est identique à celle utilisée pour l'Algo-Iter (I.4.2.a)

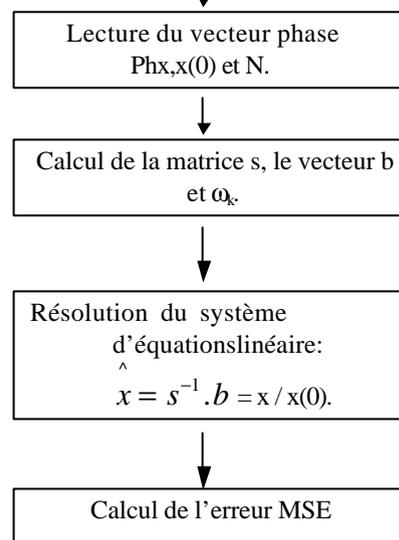
b) Partie synthèse

A partir du vecteur phase Ph_{x_k} , on construit la matrice 's' et le vecteur 'b', en appliquant les expressions (I-26), (I-27) et (I-28), puis on résoud le système (I-23).

c) Organigramme

L'organigramme de la partie analyse est identique à celui de l'Algo-Iter (I.4.1 (d)).

l'algorithme direct 'partie synthèse'



I.5 RESULTATS

Les deux algorithmes standards Algo-Iter et Algo-Dir ont été réalisés et testés sur MATLAB. Une étude expérimentale sur les performances de ces deux algorithmes a été effectuée sur plusieurs centaines de signaux.

I.5.1 RESULTATS DE ALGO-ITER

Les résultats obtenus pour Algo-Iter sont schématisés dans la figure I.3 a et b:

On remarque que :

*)La variation du temps d'exécution est liée à la longueur de la séquence et du nombre des itérations c'est-à-dire:

- plus la longueur de la séquence est grande, plus le temps CPU est grand,
- plus le nombre des itérations est grand, plus le temps CPU est grand.

*)L'erreur de reconstruction diminue avec la longueur du signal à reconstruire et le nombre des itérations

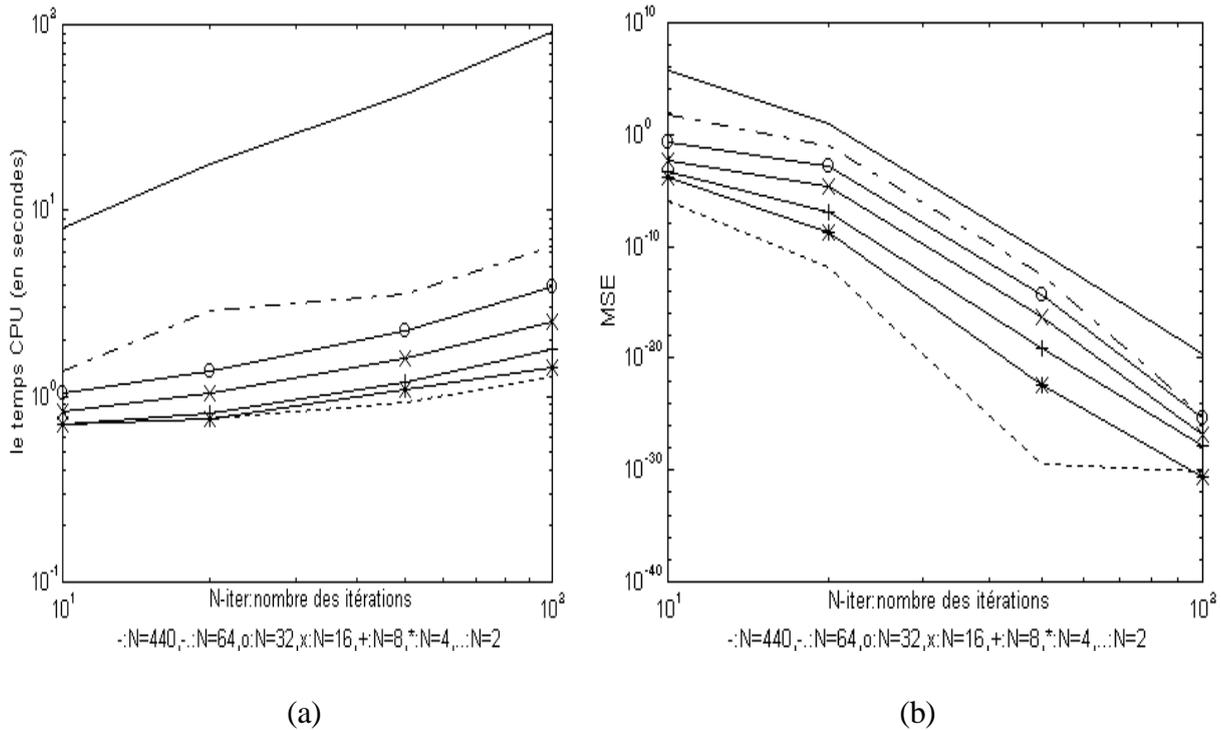


figure I.3: a) Représentation de la variation du temps CPU en fonction du nombre d'itérations pour plusieurs longueurs de suites, b) représentation de la variation de l'erreur quadratique moyenne (MSE) en fonction du nombre des itérations pour plusieurs longueurs de suites (Algo-iter).

Ces remarques nous mènent vers des problèmes tels que:

- *)Pour des signaux de longueur très grande, le calculateur est pratiquement incapable de retrouver le signal à cause de la lourdeur et la complexité des calculs.
- *)Le facteur d'échelle: Le signal reconstruit est connu à un facteur multiplicatif près du signal d'origine, alors, il faudrait toujours connaître sa première valeur.

I.5.2 RESULTATS DE ALGO-DIR

Les résultats obtenus pour Algo-Dir sont schématisés dans la figure I.3:

Les mêmes remarques citées pour les résultats de Algo-Iter sont faites pour Algo-Dir, sauf que ce dernier ne se fait qu'en une seule itération.

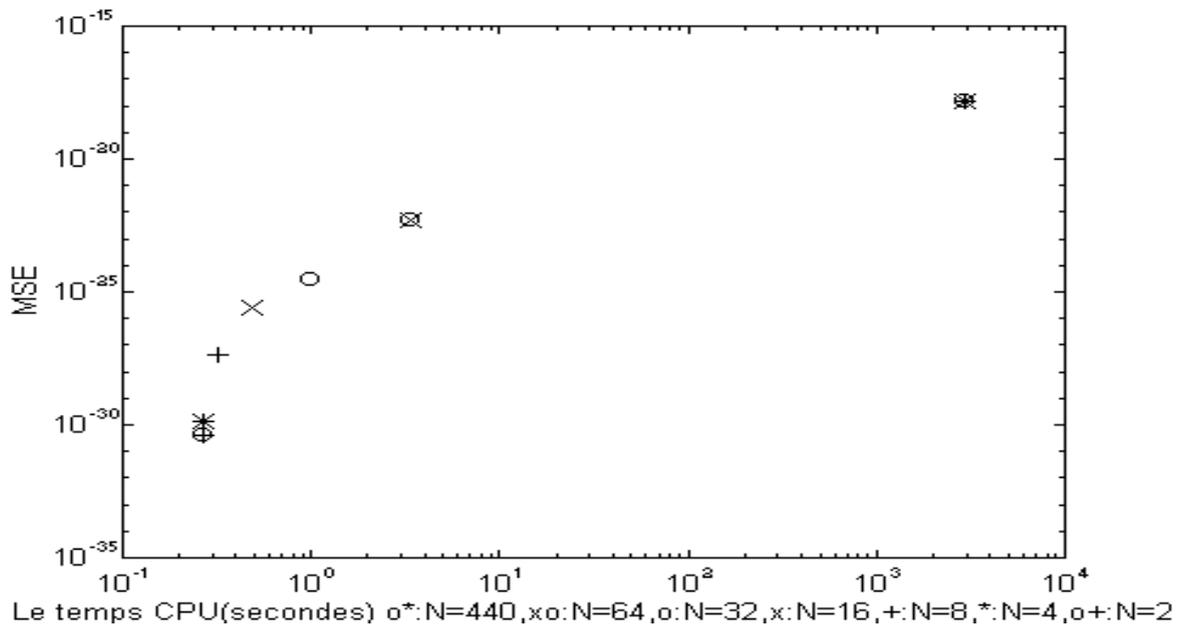


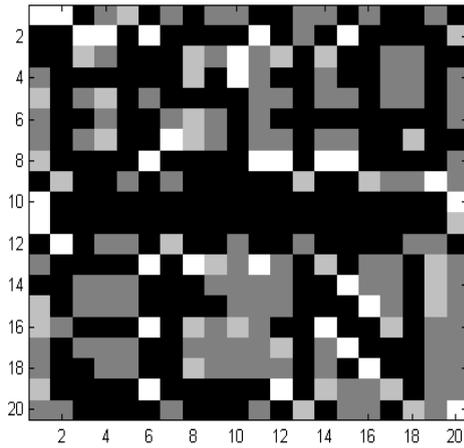
figure 1.4: représentation de MSE et du temps CPU pour différentes longueurs de suites (Algo-dir).

Les mêmes problèmes discutés dans l'Algo-Iter se répètent, sauf que la longueur du signal a plus d'effet sur la reconstruction par l'Algo-Dir car, si N(longueur de signal) est grande, alors, le calculateur aura à remplir puis inverser une matrice de $(N-1) \times (N-1)$, ceci risque une attente infinie(blocage) ; et même s'il arrive à reconstruire la séquence, elle sera entachée d'erreur, et divergera de celle d'origine.

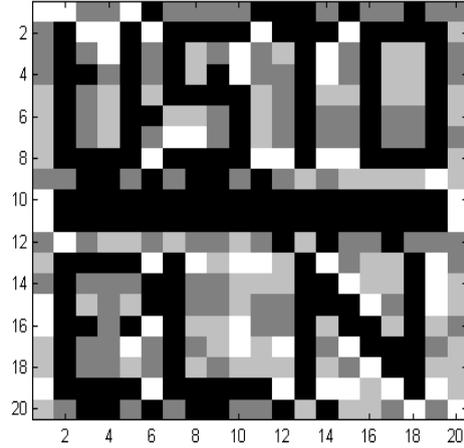
A ce propos, il a été envisagé des solutions afin de perfectionner la reconstruction à partir de la phase spectrale, et d'améliorer les algorithmes (les rendre plus crédible); réponse à cela dans le chapitre suivant.

I.6 EXEMPLE

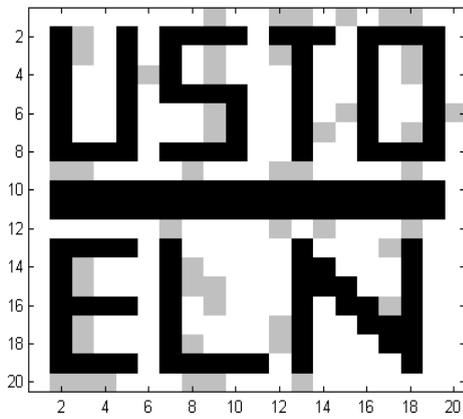
Un exemple illustratif est donné, basé sur la reconstruction d'image (20x20) pixels. Les figures 1.5 a, b et c représentent les images obtenues par Algo-Iter 100, 1000 et 20000 itérations et pour des temps CPU de 16s, 144s et 1200s. La figure I.5.d a été reconstruite en utilisant Algo-Dir (temps CPU = 1620s). Cet exemple montre que pour des images de dimensions réduite (de l'ordre de 400 pixels), Algo-Dir est plus performant. Cependant, pour des images de taille importante Algo-Dir est confronté à des problèmes d'instabilité de quantité de calcul trop importante.



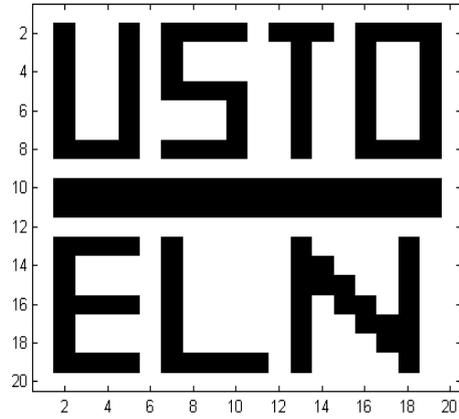
a) Algo-Iter (N-Iter=100,CPU=16s)



b) Algo-Iter (N-Iter=1000,CPU=144s)



c) Algo-Iter (N-Iter=20000,CPU=1200s)



d) Algo-Dir (CPU=1620s).

figure 1.5: Exemple de reconstruction d'une image (20x20) pixels.

a), b) et c) images obtenues à l'aide de Algo-Iter, d) image obtenue à l'aide de Algo-Dir.

I.7 CONCLUSION

Dans ce chapitre, on a développé un certain nombre de conditions où une séquence est uniquement spécifiée par les échantillons de la phase spectrale, et traité deux algorithmes qui peuvent être utilisés pour la reconstruction du signal « Algo-Iter » et « Algo-Dir ».

Comparé à l'Algo-Iter, l'Algo-Dir présente l'avantage que la séquence trouvée soit la solution désirée sans aucune itération. D'un autre côté, ce dernier doit inverser une matrice de $(N-1) \times (N-1)$ qui nous mènera à des problèmes numériques et de grandes erreurs, particulièrement quand N devient grande, alors que l'Algo-Iter peut être utilisé.

La solution à ces problèmes est vue dans le chapitre suivant .

CHAPITRE II 'ALGORITHMES UTILISANT LA SEGMENTATION'

II.1 INTRODUCTION

Le chapitre I a été consacré à l'étude et à la présentation des algorithmes standards permettant la reconstruction à partir de la phase spectrale (algorithme de forme directe et de forme itérative).

Dans ce chapitre nous présentons une version améliorée de ces algorithmes grâce à l'utilisation d'une segmentation-insertion du signal à traiter. Le programme utilisant la forme segmentée de l'algorithme direct est nommé par Algo-Dir-S, tandis que celui correspondant à la forme itérative avec segmentation insertion portera le nom de Algo-Iter-S.

II.2 PRESENTATION DES ALGORITHMES AVEC SEGMENTATION

II.2.1 DEFINITION DE LA METHODE DE SEGMENTATION

Cette méthode consiste à reconstruire une suite d'origine « x(n) » à partir des sous-suites « x_i(n) », obtenues après segmentation. Il s'avère que chacune d'elles va être reconstruite avec un facteur multiplicatif différent. Afin de palier à cet inconvénient, l'idée est d'insérer un élément commun « x0 » au début de chaque sous-suite.

Partant du signal x(n) de longueur 'N', nous avons effectué les étapes suivantes:

Etape 1

*) Segmentation de « x(n) » de support [0, N-1] en plusieurs sous-suites « x_i(n) » de longueur 'NS-1' avec NS<N.

*) Lorsque le signal « x(n) » est de longueur non multiple de 'NS-1', il suffit d'ajouter des éléments nuls, nous obtenons donc:

$$x(n)=[x_1(n), x_2(n), \dots, x_i(n)]$$

Etape 2

*) Construction de sous-suites « x_{si}(n) » de longueur 'NS'

où x_{si}(n) = [1, x_i(n)], il y a eu donc insertion de '1' au niveau de chaque sous-suite x_i(n).

$$x_s(n) = [x_{s1}(n), x_{s2}(n), \dots, x_{si}(n)].$$

Etape 3

Calcul de la FFT de chaque sous-suite $x_{si}(n)$ sur $(2.NS)$ points auquel correspond « ϕ_{si} ».

Etape 4

Concaténer les phases ϕ_{si} dans un même vecteur 'Phx' où

$$Phx = [\phi_{s1}, \phi_{s2}, \dots, \phi_{si}].$$

Etape 5

La reconstruction de « $\hat{x}(n)$ » à partir de Phx au moyen des deux algorithmes existants décrits par la suite:

II.2.2 L'ALGO-ITER AVEC SEGMENTATION (Algo-Iter-S)

On a appliqué la méthode de segmentation pour l'Algo-Iter décrit en (Chap. I ,I-4-1). Ceci est en deux parties:

a) Partie analyse

Voir les étapes 1,2,3 et 4 ci-dessus.

b) Partie synthèse

Refaire les étapes de la partie synthèse de l'Algo-Iter (ChapI. I.4.1.2.b) avec les changements suivants:

*) Effectuer L'Algo-Iter pour chaque segment de phase ' ϕ_{si} ', « $\hat{x}_{si}(n)$ » est la suite reconstruite à partir

de ' ϕ_{si} ' avec:

$$\hat{x}_{si}(n) = [\hat{x}_{si}(0), \hat{x}_i(n)].$$

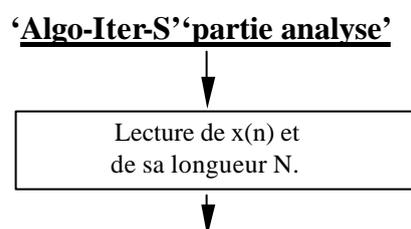
*) Concaténer les segments « \hat{x}_{si} » dans un vecteur en ôtant la première valeur $\hat{x}_{si}(0)$ correspondante à l'élément inséré 'x0':

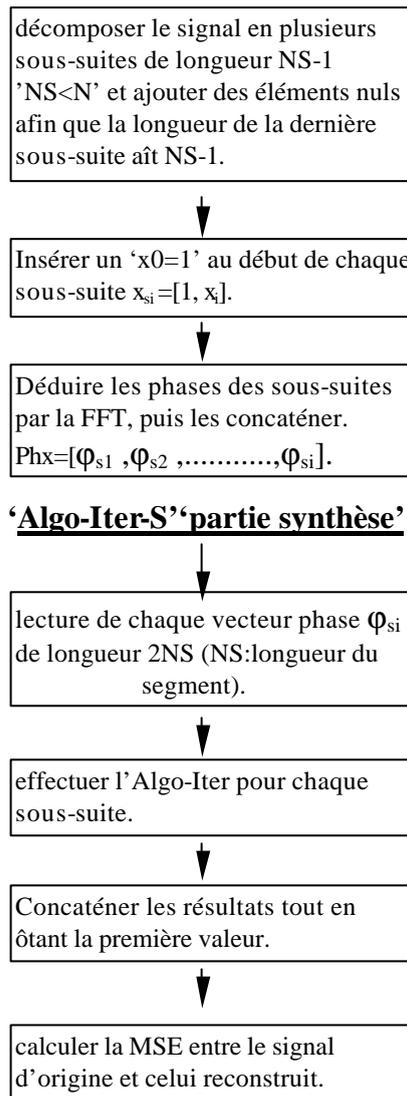
$$\hat{x}(n) = [\hat{x}_1(n), \hat{x}_2(n), \dots, \hat{x}_i(n)].$$

Avec $\hat{x}(n)$, la suite reconstruite de $x(n)$ après segmentation.

Puis, on calcule l'erreur MSE (voir formule (I-14) de I.4.1.2.(d) entre la suite d'origine $x(n)$ et celle reconstruite $\hat{x}(n)$.

c) L'organigramme





II.2.3 ALGORITHME DIRECT AVEC SEGMENTATION

L'algorithme se décompose de deux parties:

a) Partie analyse

Voir les étapes 1,2 ,3 et 4 de II.2.1.

b) Partie synthèse

On reconstruit les sous-suites $x_{si}(n)$ à partir des phases $\phi_{si}(\omega_k)$ sauvegardées dans le vecteur Phx comme suit:

*) On restitue les données nécessaires dans un fichier {Phx,N, NS, P, x0}.

Phx: Le vecteur qui contient les phases spectrales des sous-suites.

N: La longueur de la suite d'origine.

NS: La longueur de la sous-suite avec l'élément inséré x0.

P: Le nombre des sous-suites à reconstruire.

x0: L'élément inséré à chaque sous-suite.

*) On décompose le vecteur Phx en P vecteurs.

*) Effectuer Algo -Dir pour chaque phase $\phi_{si}(\omega_k)$ du vecteur Phx, ($i = 1, \dots, P$), avec P représente le nombre de sous-suites à reconstruire.

*) On juxtapose les sous-suites reconstruites « $\hat{x}_{si}(n)$ » à partir de $\phi_{si}(\omega_k)$ avec

$\hat{x}_{si}(n) = [\hat{x}_{si}(0), \hat{x}_{si}(n)]$, après avoir enlevé le premier élément correspondant à l'élément inséré ($x_0=1$). On

aura un vecteur $\hat{x}(n)$ qui correspond à la suite d'origine $x(n)$ de support $[0, N-1]$:

$$\hat{x}(n) = [\hat{x}_1(n), \hat{x}_2(n), \dots, \hat{x}_i(n)],$$

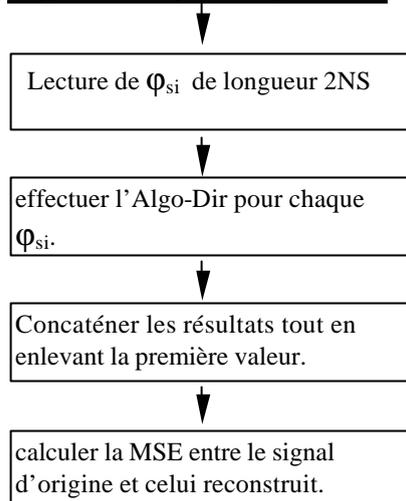
Avec $\hat{x}(n)$, la suite reconstruite de $x(n)$ après segmentation.

Puis, on calcule l'erreur MSE (voir formule (I-14) de I.4.1.2(d)) entre la suite d'origine $x(n)$ et celle reconstruite $\hat{x}(n)$.

c) Organigramme

L'organigramme de la partie analyse est identique à celui de l'Algo-Iter-S (II.2.2. (c)).

'Algo-Dir-S' 'partie synthèse'



II.3 RESULTATS

Une étude expérimentale a été effectuée pour étudier les performances des algorithmes Algo-Iter-S et Algo-Dir-S. L'expérimentation a consisté à reconstruire 100 suites de longueur $N=1000$ à l'aide des deux algorithmes. les résultats ainsi obtenus sont représentés dans les figures II.1. (Algo-Iter-S) et II.2. (Algo-Dir-S).

II.3.1 RESULTATS DE ALGO-ITER-S

Les deux figures II.1.(a) et (b) représentent respectivement la variation du temps CPU (en secondes) et de l'erreur MSE en fonction du nombre d'itérations pour les différentes longueurs de sous-suites.

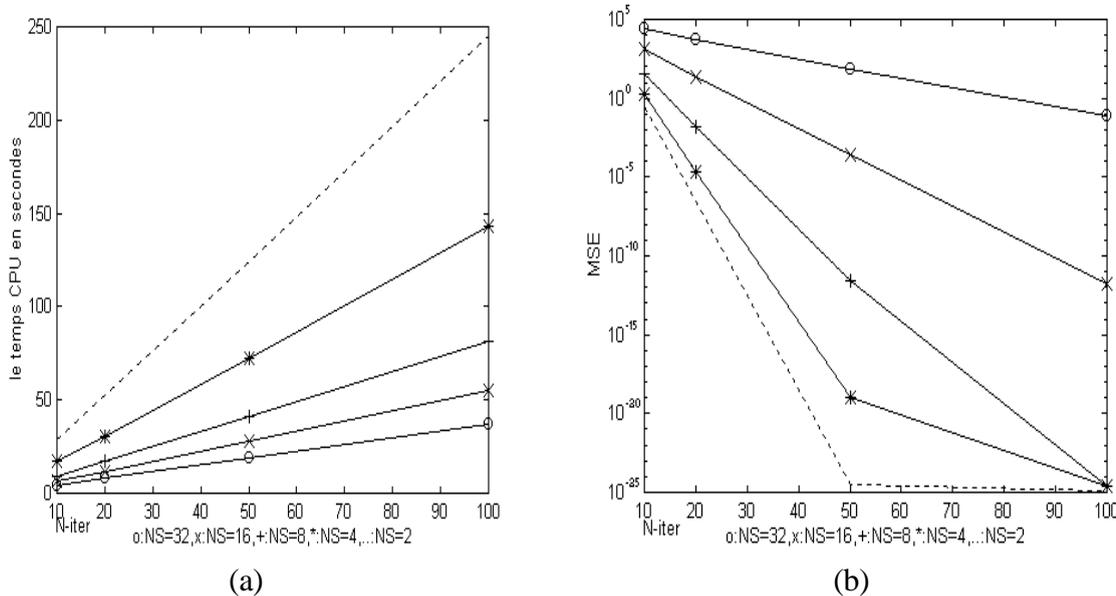


figure II.1 : Résultats expérimentaux sur 100 suites de longueur $N=1000$ à l'aide de l'Algo-Iter-S:

a) représente la variation du temps CPU en fonction du nombre d'itérations $N\text{-Iter} = (10, 20, 50, 100)$ pour une segmentation $NS=(2, 4, 8, 16, 32)$. b) représente la variation de l'erreur MSE en fonction du nombre d'itérations $N\text{-Iter} = (10, 20, 50, 100)$ pour une segmentation $NS = (2, 4, 8, 16, 32)$

Les figures de II.1. (a et b) mettent en évidence les remarques suivantes:

- l'erreur quadratique moyenne diminue avec la longueur du segment.
- l'erreur quadratique moyenne diminue quand le nombre des itérations augmente.
- le temps CPU en secondes augmente avec le nombre d'itérations.

II.3.2 RESULTATS DE ALGO-DIR-S

La figure II.2. représente la variation de l'erreur MSE en fonction du temps CPU pour une segmentation $NS = (2, 4, 8, 16, 32)$.

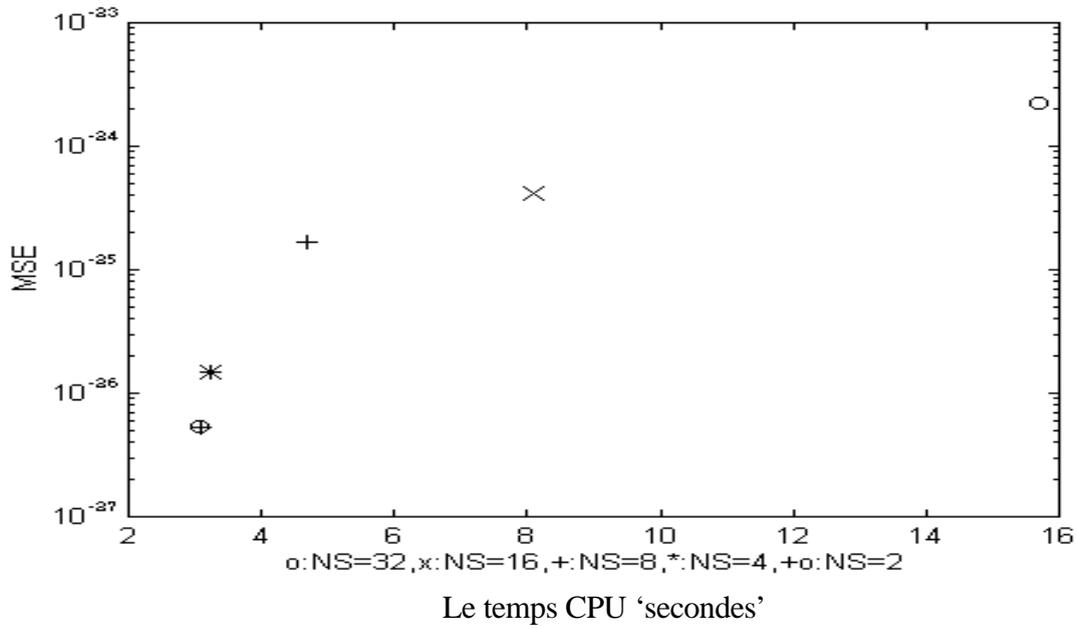


figure II.2 : Résultats expérimentaux sur 100 suites de longueur $N=1000$ à l'aide de l'Algo-Dir-S, représente l'erreur MSE en fonction du temps CPU pour une segmentation $NS = (2,4,8,16,32)$.

La figure II.2. montre que l'erreur MSE et le temps CPU augmentent avec la longueur du segment NS.

II.4 COMPARAISON ENTRE ALGO-DIR-S ET ALGO-ITER-S

NS	MSE			Le temps CPU en secondes		
	N-iter	Algo-Iter-S	Algo-Dir-S	N-iter	Algo-Iter-S	Algo-Dir-S
2	10	0.32	$5.28 \cdot 10^{-27}$	10	28.8	3.08
	20	$3.06 \cdot 10^{-7}$		20	52.3	
	50	$3.07 \cdot 10^{-25}$		50	124.5	
	100	$1.38 \cdot 10^{-25}$		100	245.2	
4	10	3.96	$1.49 \cdot 10^{-26}$	10	17	3.25
	20	$1.54 \cdot 10^{-4}$		20	30.7	
	50	$1.7 \cdot 10^{-17}$		50	72.3	
	100	$6.84 \cdot 10^{-25}$		100	143.1	
8	10	68.9	$1.69 \cdot 10^{-25}$	10	9.43	4.67
	20	0.0594		20	17.42	
	50	$7.46 \cdot 10^{-11}$		50	41.3	
	100	$2.4 \cdot 10^{-24}$		100	81.3	
16	10	19.60	$4.14 \cdot 10^{-25}$	10	6.31	8.08
	20	0.45		20	11.7	
	50	$1.029 \cdot 10^{-3}$		50	27.8	
	100	$3.15 \cdot 10^{-11}$		100	54.8	
32	10	26190	$2.17 \cdot 10^{-24}$	10	4.28	15.71
	20	6096		20	7.85	
	50	106.22		50	18.78	
	100	0.167		100	37.02	

tableau II.3

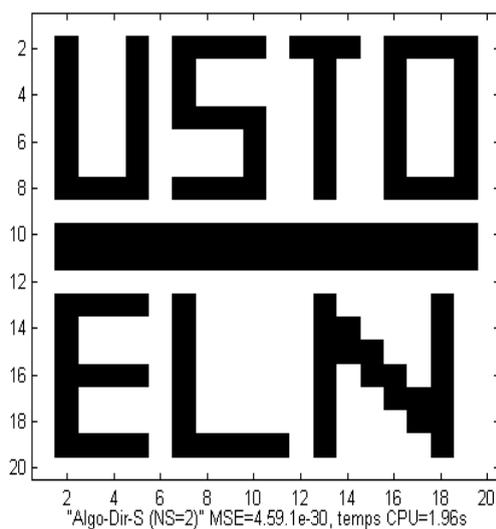
En considérant les résultats du tableau II.3, on remarque que l'erreur de reconstruction MSE et le temps CPU pour l'Algo-Dir-S sont nettement inférieurs à ceux pour Algo-Iter-S. Ce qui permet de dire que Algo-Dir-S est plus performant que Algo-Iter-S.

II.5 EXEMPLES

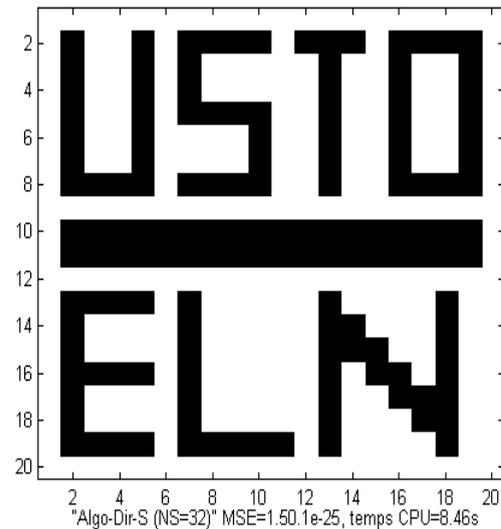
Les algorithmes utilisant une segmentation (Algo-Dir-S et Algo-Iter-S) ont été testés avec succès sur un ensemble d'images de taille importante. Rappelons que l'algorithme standard direct (Algo-Dir), ne peut être utilisé que pour des images de taille inférieure à (20x20) pixels.

Alors que l'algorithme standard itératif (Algo-Iter), nécessite un temps de calcul prohibitif et une reconstruction avec une grande dégradation.

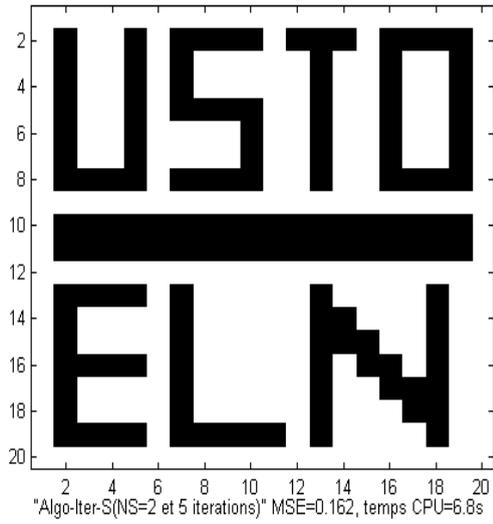
Deux exemples d'image «USTO - ELN » de taille 20 x 20 pixels et «LENNA » de taille (300 x 304) pixels sont donnés dans les figures II.3 et II.4, illustrant ainsi l'intérêt de l'approche utilisant une segmentation.



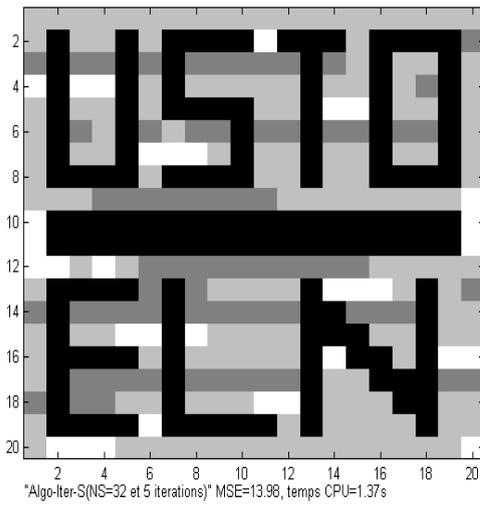
(a)



(b)



(c)



(d)

Figure II.3: reconstruction d'une image 20 x 20 pixels en utilisant la forme segmentée des algorithmes de reconstruction.

a) et b) images reconstruites (Algo-Dir-S) respectivement pour $NS = 2$ et $NS = 32$.

c) et d) images reconstruites (Algo-Iter-S) après 5 itérations respectivement pour $NS = 2$ et $NS = 32$.



(a)



Figure II.4: a) image originale « LENA » de 300 X 304 pixels.
b) image reconstruite par Algo-Dir-S avec un temps de 484 secondes.
c) image reconstruite par Algo-Iter-S pour un temps de 954s pour 50 itérations.

II.6 INTERET DE LA SEGMENTATION DANS LA RECONSTRUCTION A PARTIR DE LA PHASE

Cette méthode a été testée sur des signaux de grande taille pour les deux algorithmes. L'étude a montré que la méthode segmentation est plus rapide, la convergence est meilleure et d'autant plus grande que les sous-suite sont plus petites. Une comparaison entre les algorithmes du chapitre I (Algo-Iter et Algo-Dir) et du chapitre II (Algo-Iter-S et Algo-Dir-S) est élaborée dans le tableau II.6 pour un signal de longueur 440, une segmentation avec NS = 2 et pour différentes valeurs de l'élément inséré.

N-iter	Algo Iter					Algo Dir				
	standard		segment NS=2			standard		segment NS=2		
	MSE	temps(s)	x0	MSE	temps(s)	MSE	temps(s)	x0	MSE	temps(s)
10	5.49.10 ⁵	8	10 ⁻²	6.44.10 ⁴	12.6	1.6.10 ⁻¹⁸	2913	0.01	4.2.10 ⁻¹⁸	1.32
			1	4.09.10 ⁴	12.52					
			100	80.3	12.47					
			10 ⁴	7.47.10 ⁻²	12.41					
20	7.8	17.96	0.01	6.11.10 ⁴	23.06			1	3.9.10 ⁻²²	1.32
			1	4.97.10 ³	22.9					
			100	23.4	22.85					
			10 ⁴	1.18.10 ⁻⁷	22.8					
50	2.9.10 ⁻¹¹	42.45	0.01	59	54.16			100	2.7.10 ⁻²⁴	1.32
			1	6.11.10 ⁻³	53.9					
			100	2.23.10 ⁻⁵	54.05					
			10 ⁴	2.1.10 ⁻²³	53.9					
100	2.8.10 ⁻²⁰	91.06	0.01	2.78.10 ⁻⁹	107.34	10 ⁴	2.8.10 ⁻²⁷	1.27		
			1	2.7.10 ⁻¹³	106.9					
			100	2.2.10 ⁻¹⁵	106.8					
			10 ⁴	7.3.10 ⁻²⁷	106.7					

tableau II.6

Les résultats du tableau ci-dessus montrent que la segmentation a permis de réduire d'une manière appréciable l'erreur de reconstruction MSE. En effet pour une suite de longueur N = 440 (Algo-Dir), l'erreur MSE = 1.6 .10⁻¹⁸ alors que pour N = 440 et NS = 2 (Algo-Dir-S), l'erreur MSE = 2.8 .10⁻²⁷ (x0 = 10⁴).

La même remarque est valable si on compare l'Algo-Iter avec l'Algo-Iter-S, pour une suite de longueur N = 440 (Algo-Iter), l'erreur MSE = 2.8 .10⁻²⁰ (N-Iter = 100)alors que pour N = 440 et NS = 2 (Algo-Iter-S), l'erreur MSE = 7.3 .10⁻²⁷ (N-Iter = 100 et x0 = 10⁴).

Le tableau II.6 montre que la segmentation a permis de réduire le temps CPU, par exemple si on compare l'Algo-Dir avec l'Algo-Dir-S, le temps CPU en secondes est pratiquement divisé par 2000. Ce qui n'est pas le cas pour l'Algo-Iter-S, car le temps CPU augmente lorsque la longueur du segment NS diminue, mais la convergence est meilleure par rapport a l'Algo-Iter.

Le tableau montre aussi que l'erreur MSE calculée pour l'Algo-Dir-S et l'Algo-Iter-S dépend du choix de l'élément inséré x0, elle diminue lorsque ce dernier augmente, de ce fait, on conclut que l'erreur est contrôlée par x0.

II.7 CONCLUSION

Nous avons mis en évidence l'intérêt de l'utilisation de la segmentation dans les algorithmes de reconstructions. En effet, non seulement elle permet de traiter des suites de longueur très importantes (ce qui n'est pas le cas des algorithmes standards (chapitre I)) mais présente aussi des améliorations du point de vue erreur de reconstruction et de temps machine nécessaire.

Les résultats donnés par Hayes, Lim et Oppenheim ont une importance théorique significative, mais limités dans la pratique, puisqu'ils supposent que la phase utilisée est exacte. Ce qui n'est pas le cas réel, puisque l'environnement est entaché de bruit.

Le chapitre suivant traite le cas où la phase utilisée est dégradée par un bruit.

CHAPITRE III : 'ETUDE DE BRUIT LORS DE LA RECONSTRUCTION'

III.1 INTRODUCTION

Dans les chapitres précédents, une étude des performances des algorithmes de reconstruction a été menée. La discussion prend pour hypothèse que la phase spectrale est exactement connue et donc, l'erreur de reconstruction était due à la nature de l'algorithme intrinsèquement.

Nous avons pu mettre en relief dans ce cas de figure, que les algorithmes standards présentaient des limites: des signaux de grande taille nécessitaient des temps de calcul prohibitifs (Algo-Iter) et que des cas d'instabilité fréquents se présentaient lors de la reconstruction (Algo-Dir). Par contre, les algorithmes Algo-Iter-S et Algo-Dir-S, utilisant une segmentation, permettent de réduire drastiquement la quantité de calcul ainsi que l'erreur de reconstruction. Nous avons aussi montré que l'erreur est d'autant plus petite que la longueur 'N' de la suite à reconstruire est petite (Algo-Dir et Algo-Iter). En ce qui concerne les algorithmes utilisant une segmentation (Algo-Dir-S et Algo-Iter-S), l'erreur de reconstruction est d'autant plus petite que la sous-suite 'NS' est petite. Et donc, l'erreur de reconstruction dépendait principalement de la longueur de la sous-suite, indépendamment de la longueur de la suite d'origine à reconstruire.

Quand est-il de la reconstruction dans un contexte bruité ?

C'est la question que nous voulons élucider dans ce chapitre.

Notre propos sera de confirmer et d'étendre les résultats de Espy et Lim [3]. En effet, ceux-ci ont fait l'étude de Algo-Dir dans un contexte bruité.

Dans ce chapitre, une étude de tous les algorithmes présentés précédemment (Algo-Dir, Algo-Dir-S, Algo-Iter et Algo-Iter-S), seront étudiés dans un milieu bruité.

L'expérimentation se fera avec un bruit blanc additif à la phase. L'étude statistique se fera dans un premier temps sur 1000 suites (Algo-Dir, Algo-Dir-S et Algo-Iter) et pour des raisons pratiques (temps machine nécessaire pour plusieurs jours) nous avons été amené à procéder par la suite sur des échantillons de 200 suites (Algo-Iter-S). Nous avons à chaque fois vérifier que les résultats ainsi obtenus étaient consistants statistiquement.

III.2 ETUDE EXPERIMENTALE

Nous nous intéressons à l'étude de l'effet d'un bruit additif sur la phase lors de la reconstruction du signal. Dans toutes les expériences que nous allons présenter dans ce qui suit, nous avons utilisé pour notre étude un bruit blanc.

Le bruit blanc additif généré, est de densité uniforme appartenant à l'intervalle $[-br, br]$ avec $br = [\pi \cdot 10^{-5}, \pi \cdot 10^{-4}, \pi \cdot 10^{-3}, \pi \cdot 10^{-2}, \pi \cdot 10^{-1}]$, puisque pour la plupart des séquences, le bruit de niveau inférieur à $\pi \cdot 10^{-5}$ a de négligeables effets sur la séquence reconstruite, tandis que celui de niveau supérieur à $\pi \cdot 10^{-1}$ a des effets néfastes.

Pour que l'étude statistique soit représentative, nous avons reconstruit 1000 suites « $x(n)$ » auxquelles nous ajoutons le bruit additif « br ». Les suites $x(n)$ ont été générées aléatoirement suivant une loi gaussienne.

Pour étudier l'effet du bruit sur l'erreur de reconstruction dans l'expérience, on a implémenté nos algorithmes pour 1000 séquences. Pour une suite « $x(n)$ » l'erreur de reconstruction choisie est l'erreur quadratique moyenne normalisée « NMSE1 » calculée à partir de l'équation III-1:

$$NMSE1 = \frac{\sum_{n=0}^{N-1} (x(n) - \beta \hat{x}(n))^2}{\sum_{n=0}^{N-1} x^2(n)} \quad (III-1)$$

où $x(n)$ est la séquence d'origine de support $[0, N-1]$,

$\hat{x}(n)$: est la séquence reconstruite à partir de la phase dégradée.

β : le facteur d'échelle choisi pour minimiser l'erreur dans l'équation (III-1).

Nous utilisons aussi le logarithme de l'erreur «NMSE1 » que nous nommerons par LOG(NMSE1) où

$$LOG(NMSE1) = \ln(NMSE1) = \ln\left(\frac{\sum_{n=0}^{N-1} (x(n) - \beta \hat{x}(n))^2}{\sum_{n=0}^{N-1} x^2(n)}\right) \quad (III-2)$$

L'expérimentation que nous ferons portera sur un ensemble de suites. L'erreur «NMSE » ou LOG(NMSE) sera la moyenne des erreurs sur les 1000 suites reconstruite telle qu'elle a été utilisée par les auteurs ESPY et LIM [3]. Ceci nous permettra de faire une comparaison des résultats.

$$LOG(NMSE) = 1/1000 \sum_{j=1}^{1000} \left(\ln\left(\frac{\sum_{n=0}^{N-1} (x_j(n) - \beta \hat{x}_j(n))^2}{\sum_{n=0}^{N-1} x_j^2(n)}\right) \right)$$

où

$x_j(n)$: est la j-ème suite originale.

$\hat{x}_j(n)$: la j-ème suite reconstruite.

III.3 SCHEMAS UTILISES

On a deux types d'algorithmes de reconstruction, algorithmes standards et algorithmes utilisant une segmentation.

Pour les deux algorithmes standards (Algo-Dir et Algo-Iter), le processus de reconstruction utilisé est schématisé dans la figure III.1.a; dans cette figure $x(n)$ dénote la suite d'origine de support $[0, N-1]$ qui satisfait les conditions de HAYES [2], $Ph_x(\omega_k)$ représente la phase spectrale de $x(n)$ échantillonnée à $(N-1)$ fréquences distinctes entre 0 et π , 'br' bruit blanc ajouté à la phase exacte pour obtenir la phase dégradée, « $\hat{x}(n)$ » la séquence reconstruite à partir de cette phase dégradée.

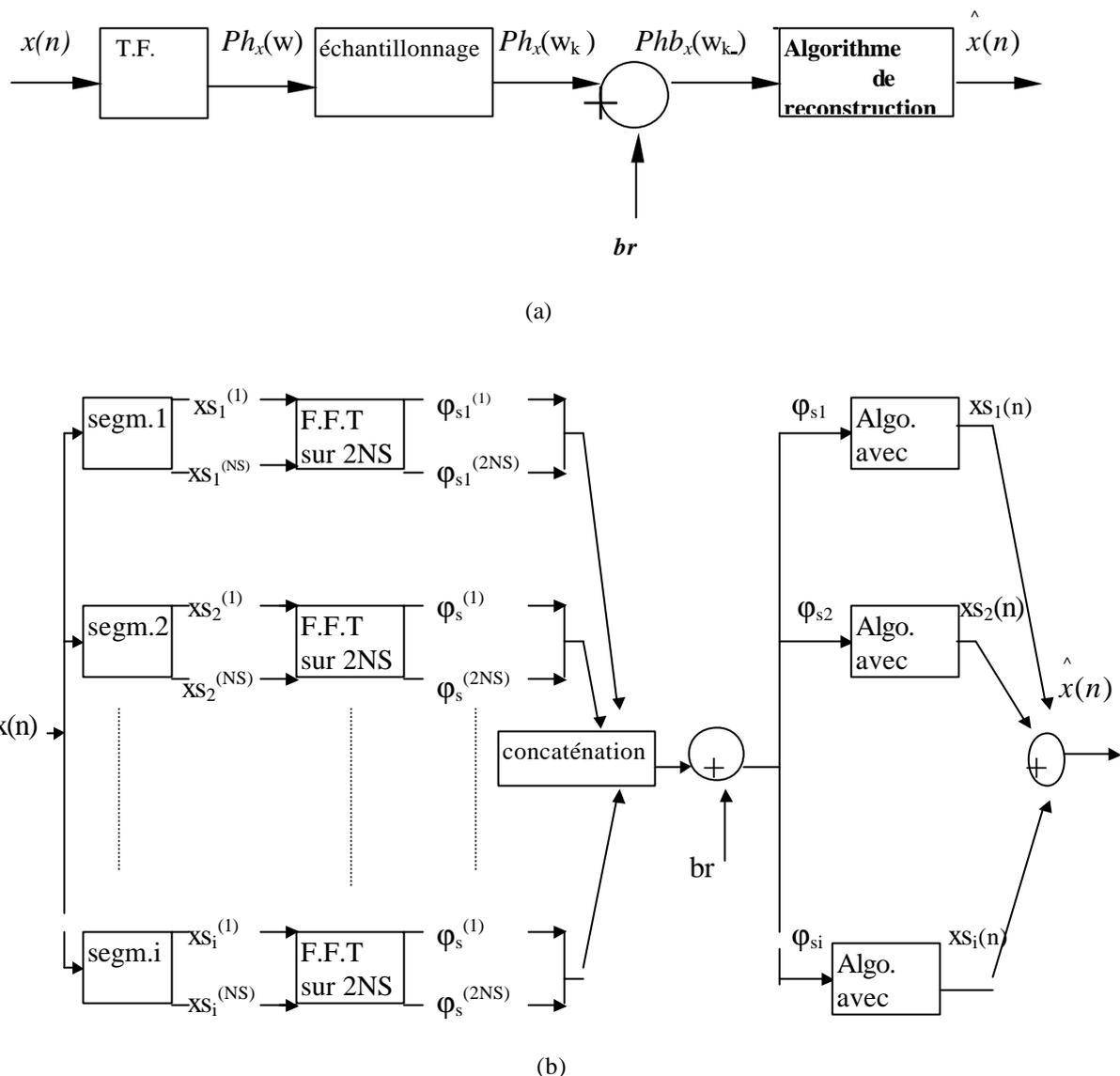


Figure III.1: Schéma utilisé pour l'expérimentation dans un milieu bruité en utilisant: a) les algorithmes standards (Algo-Dir et Algo-Iter), b) les algorithmes utilisant une segmentation (Algo-Dir-S et Algo-Iter-S).

Pour les algorithmes utilisant une segmentation (Algo-Dir-S et Algo-Iter-S), le processus de reconstruction utilisé est schématisé dans la figure III.1.b. Dans cette figure $x_{si}(n)$ dénote le i -ème segment de $x(n)$ de support $[0, NS]$, $\varphi_{si}(\omega_k)$ représente la phase spectrale de $x_{si}(n)$ échantillonnée à $(NS-1)$ fréquences distinctes entre 0 et π .

Les valeurs de NS (longueur du segment) et br sont ($NS=2,4,8,16,32$) et ($br=\pi \cdot 10^{-5}, \pi \cdot 10^{-4}, \pi \cdot 10^{-3}, \pi \cdot 10^{-2}, \pi \cdot 10^{-1}$).

Les deux processus sont répétés pour 1000 suites et à chaque fois on calcule l'erreur $\text{LOG}(\text{NMSE1})$, puis, l'erreur moyenne $\text{LOG}(\text{NMSE})$ définie pour 1000 suites.

III.4 ETUDE DE L'EFFET DU BRUIT LORS DE LA RECONSTRUCTION

III.4.1 ETUDE DE ALGO-DIR EN CONTEXTE BRUITE

Quand la phase de la transformée de Fourier est dégradée par un bruit additif, la relation (I-22) du chap.1 s'écrit comme suit:

$$\sum_{n=1}^{N-1} \hat{x}(n) \sin(Phx(\mathbf{w}) + br + n\mathbf{w}) = -x(0) \sin(Phx(\mathbf{w}) + br) \quad (\text{III-3})$$

Où

br : représente le bruit rajouté à la phase

$\hat{x}(n)$: la séquence reconstruite à partir de la phase dégradée par un bruit 'br' non nul.

L'objectif de cette expérience, est étudier expérimentalement l'effet du bruit 'br' sur l'erreur $\text{LOG}(\text{NMSE})$ entre la suite d'origine $x(n)$ et celle en cours de reconstruction $\hat{x}(n)$ par le développement d'une série d'expériences avec une discussion des résultats.

Notre propos est de vérifier les résultats obtenus par ESPY et LIM [3], pour cela, on a utilisé une phase dégradée pour reconstruire le signal d'origine « $x(n)$ » par Algo-Dir, puis, discuter l'erreur $\text{LOG}(\text{NMSE})$ entre la suite reconstruite « $\hat{x}(n)$ » et la suite d'origine « $x(n)$ ».

Ces expériences ont été faites à quatre reprises, d'où on a relevé deux, représentées dans la figure III.2 (a- pour un premier essai et b- pour un deuxième essai). On a pris des suites de longueurs différentes

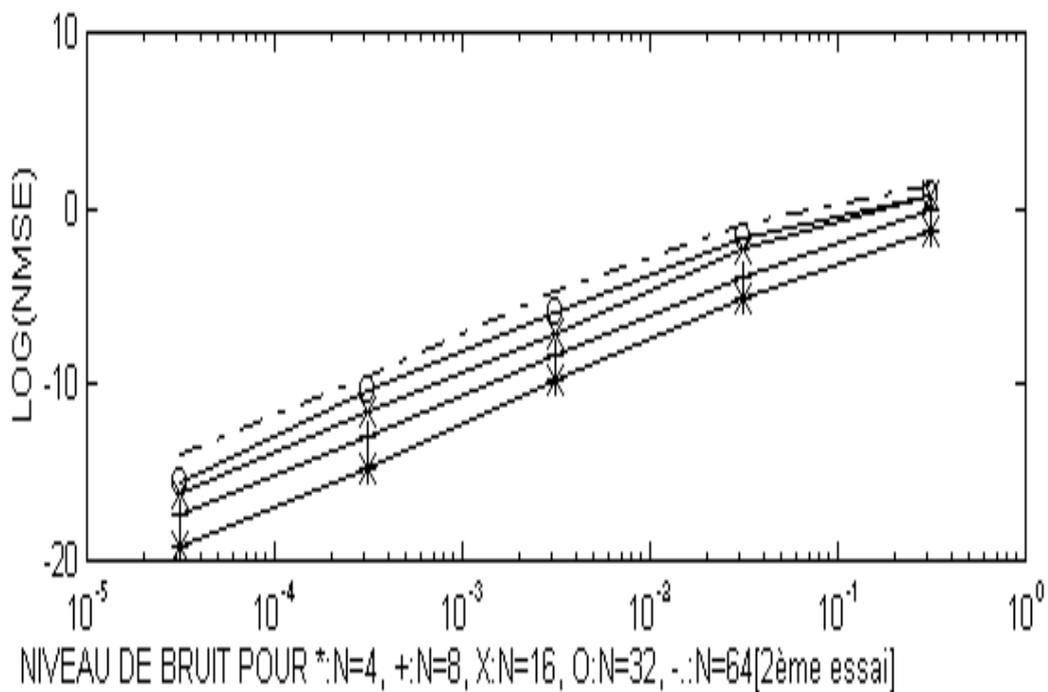
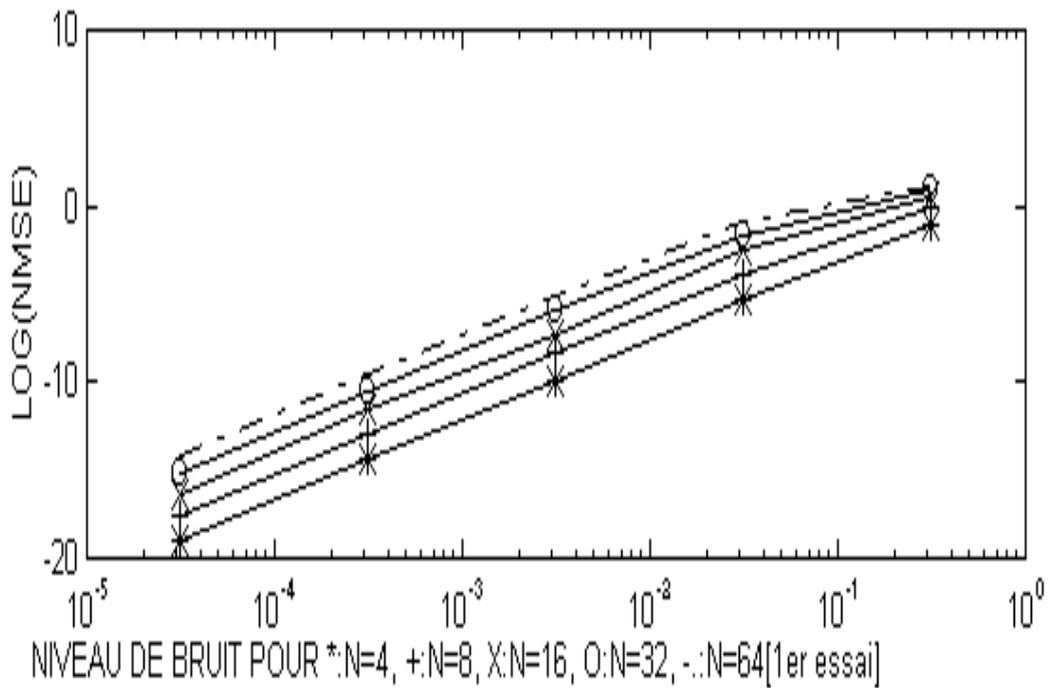
« N=4, N=8, N=16, N=32, N=64 » et bruité à chaque fois la phase spectrale avec un bruit dont le niveau appartient à l'intervalle [$\pi \cdot 10^{-5}$, $\pi \cdot 10^{-4}$, $\pi \cdot 10^{-3}$, $\pi \cdot 10^{-2}$ et $\pi \cdot 10^{-1}$], puis, on a calculé à chaque fois le LOG(NMSE) correspondant à la moyenne des erreurs pour chaque niveau de bruit.

L'erreur moyenne LOG(NMSE) est représentée sur une échelle linéaire, tandis que le bruit est représenté sur une échelle logarithmique.

Les suites « $\hat{x}(n)$ » ont été construites à partir de 1000 phases différentes, chacune correspond à une suite d'origine « $x(n)$ ».

Nous remarquons que les deux figures III.2 a et b sont équivalentes, ce qui nous assure que 1000 suites sont suffisantes pour notre étude statistique, et pour une représentation significative.

(a)



(b)

Figure III.2: Représentation de l'erreur LOG(NMSE) en fonction du niveau de bruit(br) pour plusieurs longueurs de suites ' $N=4, N=8, N=16, N=32, N=64$ ' (Algo-Dir).

Nous observons sur ces deux figures que la moyenne LOG(NMSE) augmente avec le niveau de bruit et la longueur de la séquence N. Elle diminue avec 'N'.

Le tableau III.3. suivant donne les valeurs numériques que nous avons obtenus et représentés dans la figure III.2.a.

N \ br	0	$\pi \cdot 10^{-5}$	$\pi \cdot 10^{-4}$	$\pi \cdot 10^{-3}$	$\pi \cdot 10^{-2}$	$\pi \cdot 10^{-1}$
4	-70.54	-19.07	-14.4553	-9.8473	-5.3349	-1.1510
8	-68.37	-17.5506	-13.001	-8.3275	-3.9302	-0.0903
16	-65.76	-16.3619	-11.5395	-7.2257	-2.5414	-0.5584
32	-62.66	-15.1067	-10.5704	-5.9995	-1.7031	-0.9820
64	-60.26	-14.1653	-9.5313	-5.0301	-0.8504	-1.0865

tableau III.3: Variation de l'erreur LOG(NMSE) en fonction de N et br(ALGO-DIR).

Si on considère les résultats du tableau III.3., on remarque que la moyenne LOG(NMSE) est pratiquement faible pour un niveau de bruit 'br' nul. La moyenne LOG(NMSE) dépend de la longueur de la suite 'N'.

Une étude comparative des résultats que nous venons d'obtenir figure III.2. et tableau III.3. avec ceux obtenus par ESPY et LIM [3] montrent qu'il y a une concordance des résultats. De même, la plus faible erreur calculée par ESPY et LIM [3] est 'LOG(NMSE)=-9' pour une suite de longueur 'N=4' à un bruit $br = \pi \cdot 10^{-5}$, tandis que le tableau III.3. montre que la plus petite erreur est LOG(NMSE)= -19.07 pour N= 4 et $br = \pi \cdot 10^{-5}$.

Alors, notre programme a prouvé de meilleures améliorations que celui de ESPY et LIM [3].

III.4.2 ETUDE DE ALGO-DIR-S DANS UN CONTEXTE BRUITE

Nous nous proposons d'étudier l'effet du bruit sur la phase en utilisant l'Algo-Dir-S. Le processus utilisé est schématisé dans la figure III.1.b. La figure III.4. présente la moyenne LOG(NMSE) pour une suite d'origine 'x(n)' de longueur 'N=64' avec des segments de longueurs NS=2,4,8,16 et 32.

Nous observons que l'erreur 'LOG(NMSE)' augmente avec la longueur du segment 'NS' (et non plus avec la longueur de la suite d'origine 'N') et le niveau du bruit 'br'.

Cette erreur pour une segmentation par 'NS' est cohérente à celle pour une suite de longueur 'N=NS'. En effet, en considérant le tableau III.3. et III.5., on remarque que l'erreur de reconstruction LOG(NMSE) (Algo-Dir) pour une suite de longueur 'N' est approximativement celle obtenue pour une suite de longueur 'N1' ($N1 \gg N$), mais, segmentée en sous-suites de longueur 'NS'. Donc, nous retrouvons les mêmes résultats obtenus dans un contexte non bruité: l'erreur de reconstruction en utilisant un algorithme avec segmentation dépend principalement de la longueur du segment indépendamment de la longueur 'N'. Une étude comparative des résultats obtenus avec Algo-Dir dans un contexte bruité (tableau III.3.) et ceux obtenus avec Algo-Dir-S dans un milieu bruité (tableau III-5), met en relief que l'erreur LOG(NMSE) pour Algo-Dir-S dépendant principalement de 'NS', est pratiquement équivalente à celle obtenue par Algo-Dir pour une suite de longueur 'N = NS'.

Une remarque importante en ressort, c'est que pour l'Algo-Dir, l'erreur peut être contrôlée à l'aide de la longueur du segment 'NS' quelque soit la longueur du signal.

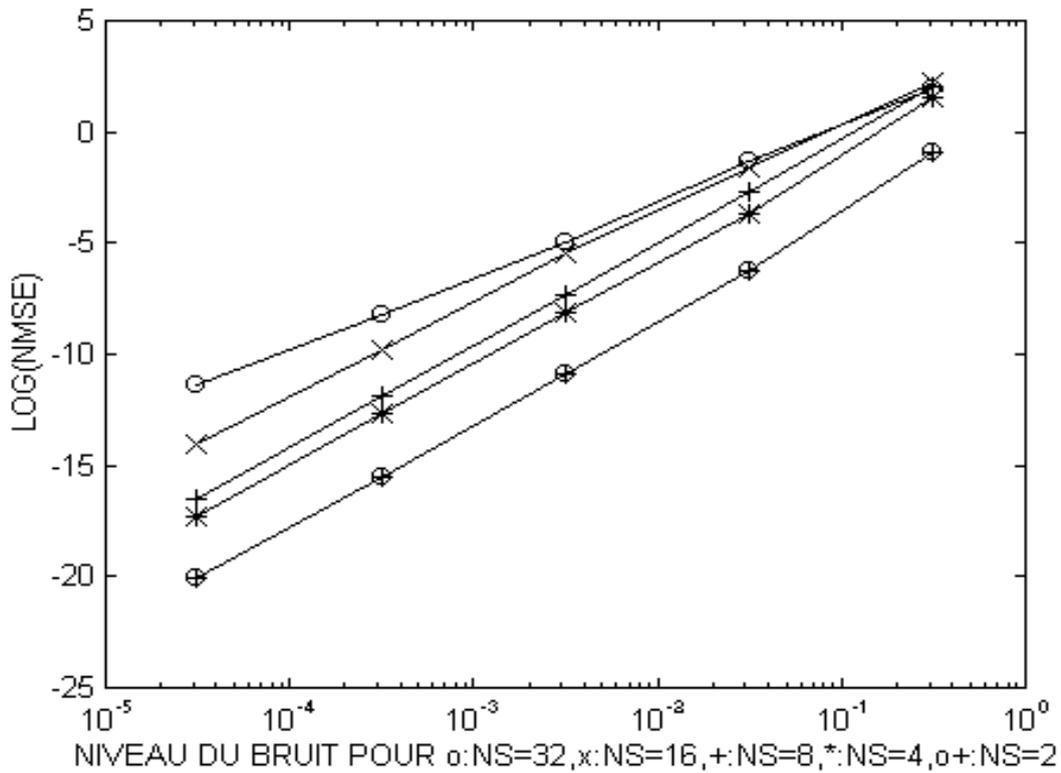


Figure III.4: Représentation de l'erreur LOG(NMSE) (ALGO-DIR-S) en fonction du niveau de bruit pour $NS=[2, 4, 8, 16 \text{ et } 32]$.

La figure III.4 montre que la moyenne LOG(NMSE) augmente avec le niveau du bruit et la longueur du segment NS. On remarque aussi que, quelque soit le niveau de bruit br, la moyenne LOG(NMSE) diminue lorsque la longueur NS diminue.

NS \ br	0	$\pi \cdot 10^{-5}$	$\pi \cdot 10^{-4}$	$\pi \cdot 10^{-3}$	$\pi \cdot 10^{-2}$	$\pi \cdot 10^{-1}$
2	-71.5074	-20.0846	-15.4896	-10.8819	-6.2710	-0.9616
4	-68.5419	-17.2937	-12.6506	-8.1543	-3.6422	1.5416
8	-66.1842	-16.5297	-11.8895	-7.3018	-2.7030	2.0868
16	-62.7269	-14.0658	-9.8303	-5.4933	-1.6238	2.1997
32	-61.4538	-11.3749	-8.1761	-4.9740	-1.3134	1.9722

tableau III.5.: Variation de l'erreur LOG(NMSE) en fonction de NS et br(ALGO-DIR-S).

III.4.3 ETUDE DE ALGO-ITER EN CONTEXTE BRUITE

La même étude expérimentale a été entreprise pour l'étude de Algo-Iter en milieu bruité. Les résultats sont représentés dans la figure III.6. et le tableau III.7.

Mêmes observations, nous remarquons que l'erreur augmente avec le bruit 'br' et avec la longueur 'N' de la suite en cours de reconstruction.

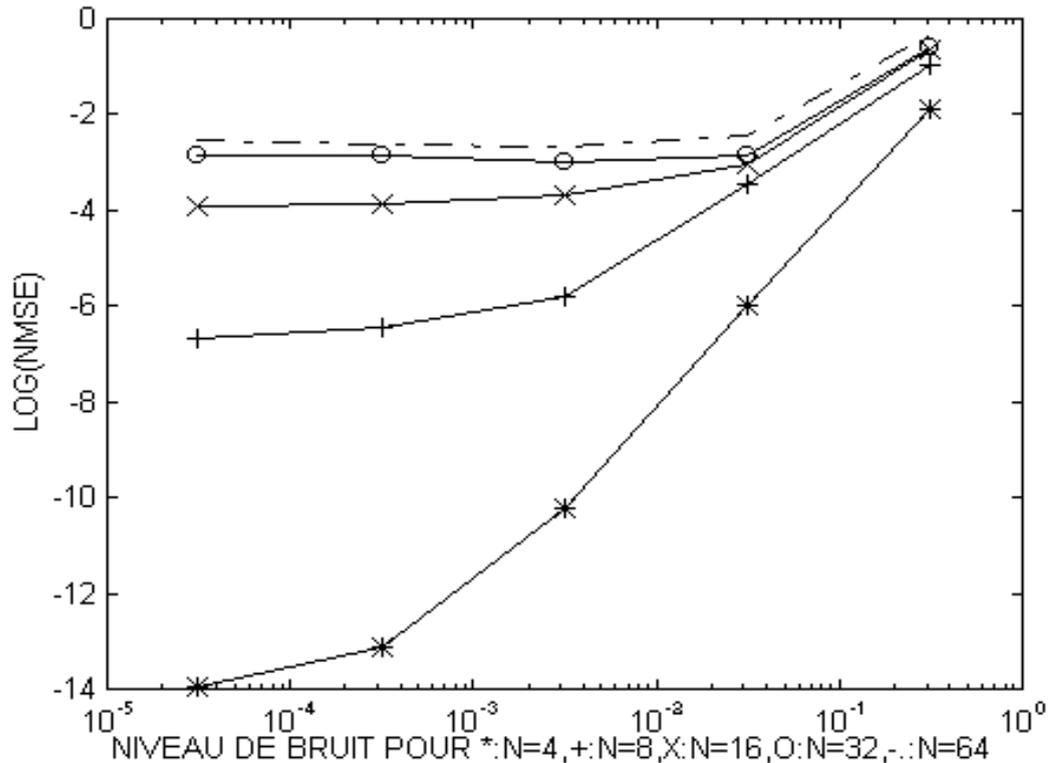


Figure III.6. : Représentation de l'erreur LOG(NMSE) en fonction du niveau de bruit pour plusieurs longueurs de suites (ALGO-ITER).

On a pris des suites de longueurs différentes «N=4, N=8, N=16, N=32, N=64 » et bruité à chaque fois la phase spectrale avec un bruit blanc de densité uniforme dont le niveau appartient à l'intervalle $[\pi \cdot 10^{-5}, \pi \cdot 10^{-4}, \pi \cdot 10^{-3}, \pi \cdot 10^{-2} \text{ et } \pi \cdot 10^{-1}]$, puis, on a calculé le LOG(NMSE) correspondant à chaque niveau. Pour la reconstruction de chaque séquence le nombre d'itérations utilisé est N-iter=100.

N	br					
	0	$\pi \cdot 10^{-5}$	$\pi \cdot 10^{-4}$	$\pi \cdot 10^{-3}$	$\pi \cdot 10^{-2}$	$\pi \cdot 10^{-1}$
4	-28.6064	-14.5112	-11.6833	-9.4073	-5.5307	-1.6796
8	-5.5271	-6.9094	-6.4664	-5.8486	-4.1774	-0.7932
16	-3.8981	-3.9424	-4.0445	-3.7312	-3.3220	-0.8123
32	-3.0742	-2.9544	-2.8863	-2.8543	-2.7503	-0.8358
64	-2.5555	-2.6214	-2.6127	-2.6186	-2.6553	-0.4651

tableau III.7. : Variation de l'erreur LOG(NMSE) en fonction de N et br(ALGO-ITER).

On remarque que :

- L'erreur augmente avec le niveau du bruit, cependant, la figure III.6. et le tableau III.7. montrent que pour des suites de longueur supérieure ou égale à '16', l'erreur de reconstruction n'est pas sensible à un bruit d'amplitude comprise dans l'intervalle $[\pi \cdot 10^{-5}, \pi \cdot 10^{-2}]$
- L'erreur augmente avec la longueur de la séquence, quelque soit le bruit.

III.4.4 ETUDE DE ALGO-ITER-S DANS UN CONTEXTE BRUITE

L'étude statistique que nous allons réaliser se fera seulement sur 200 suites. Ceci pour des raisons de temps de calcul. Cependant, nous avons vérifié, que 200 suites au lieu de 1000, comme dans les expériences précédentes, permettent de donner des résultats statistiques représentatifs. De ce fait, on a implémenté Algo-Iter-S (figure III.8.) afin de minimiser l'erreur, en segmentant une grande suite en plusieurs petites sous-suites.

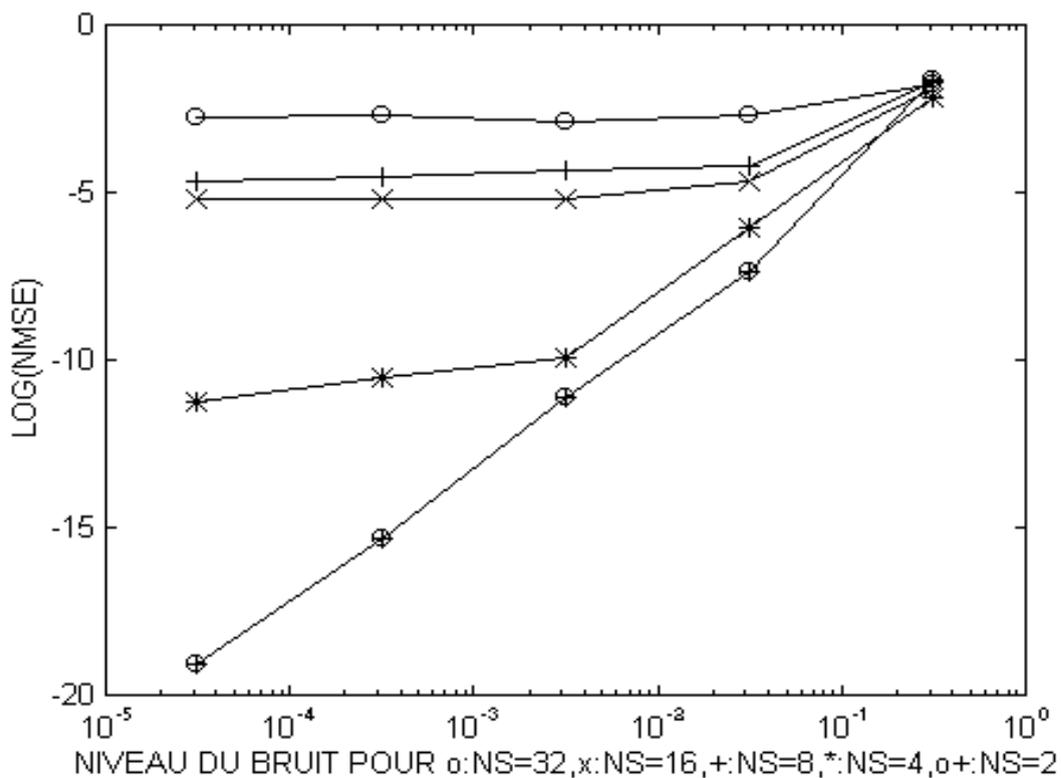


Figure III.8 : Représentation de l'erreur de reconstruction en fonction du niveau de bruit pour différentes longueurs de segments 'NS'. (ALGO-ITER-S)

L'algorithme a été implémenté pour 200 suites. Tous les résultats ont été obtenus après un nombre d'itérations (N-Iter=100).

La figure III.8. représente la variation de l'erreur de reconstruction d'une séquence de longueur '64' segmentée par « 2, 4, 8, 16 et 32 » en fonction du bruit à l'aide de Algo-Iter-S.

br					
NS	0	$\pi \cdot 10^{-5}$	$\pi \cdot 10^{-4}$	$\pi \cdot 10^{-3}$	$\pi \cdot 10^{-2}$

2	-71.3672	-21.4945	-15.2351	-10.5711	-6.3693	-2.2417
4	-3.9854	-10.5340	-10.7384	-8.4373	-5.7706	-2.7427
8	-3.4891	-4.5719	-4.3164	-4.4119	-4.5395	-1.6045
16	-2.9089	-5.1693	-5.1721	-5.2109	-4.6982	-1.8776
32	-2.3891	-2.5856	-2.8356	-2.5771	-2.4211	-1.6872

tableau III.9: variation de l'erreur LOG(NMSE) en fonction de NS et br (ALGO-ITER-S).

L'étude de la figure III.8. et le tableau III.9. nous amène à formuler les remarques suivantes :

- L'erreur LOG(NMSE) augmente avec la longueur du segment 'NS'.
- L'erreur LOG(NMSE) augmente avec le niveau du bruit 'br'.
- Comme pour l'algorithme Algo-Dir-S, l'erreur de reconstruction de Algo-Iter-S pour un segment 'NS' correspond approximativement à l'erreur obtenue pour une suite 'N=NS' pour Algo-Iter.

III.5 COMPARAISON DES RESULTATS OBTENUS POUR LES DIFFERENTS ALGORITHMES TRAITES

La figure III.10.a. représente, en pointillé, l'erreur LOG(NMSE) obtenue à partir de Algo-Dir pour $N=64$ et, en traits, l'erreur LOG(NMSE) pour différentes valeurs de 'NS' et 'br' à l'aide de Algo-Dir-S.

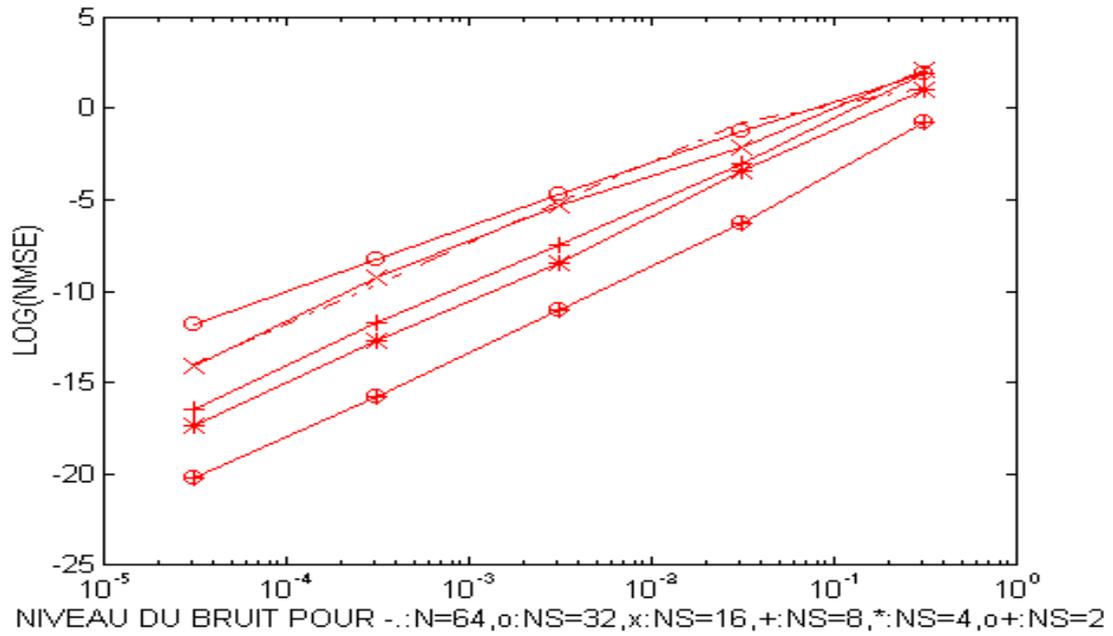
La figure III.10.b. compare les résultats de Algo-Iter (en pointillé) pour 'N=64' et ceux obtenus pour Algo-Iter-S pour les différentes valeurs de 'NS' et 'br' pour une suite d'origine de longueur 'N=64'.

La comparaison des résultats obtenus dans un milieu bruité par Algo-Dir et Algo-Dir-S nous fait remarquer que la segmentation a permis de réduire d'une manière appréciable l'erreur de reconstruction.

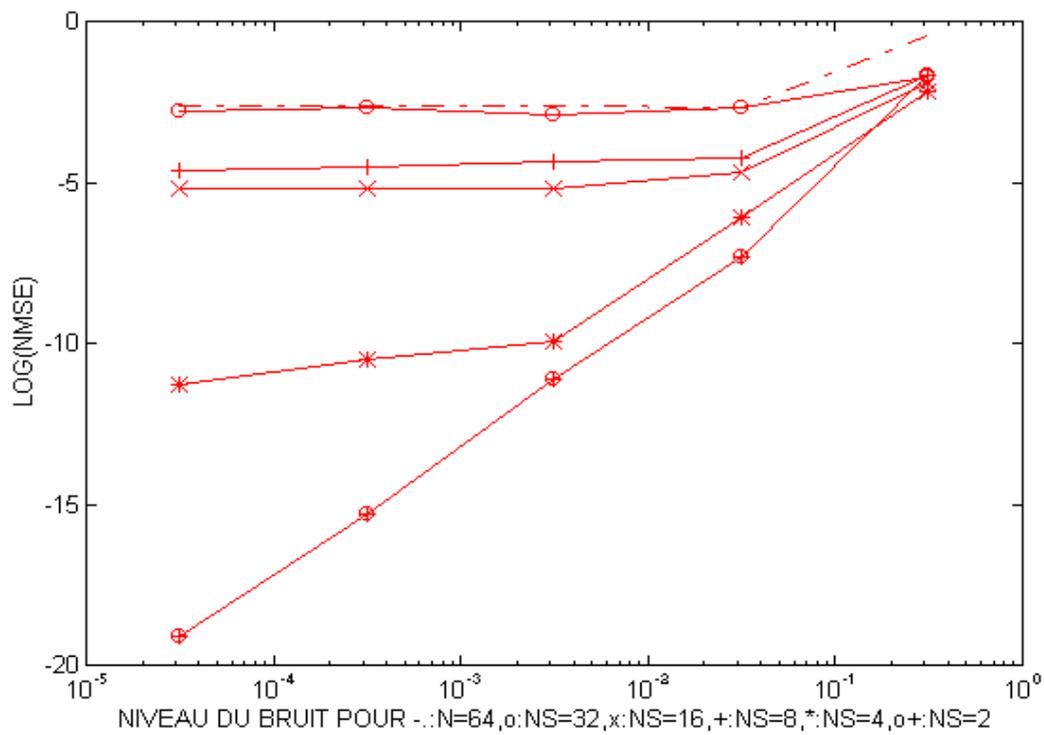
En effet, pour une suite 'N = 64' (Algo-Dir) l'erreur LOG(NMSE) = -14.16 (tableau III.3) alors que pour 'N = 64' et 'NS = 2' par exemple l'erreur LOG(NMSE) = -20.08 (tableau III.5).

La même remarque est valable si on remarque l'algorithme Algo-iter et Algo-Iter-S à partir des tableaux III.7. et III.9. L'erreur LOG(NMSE) = -2.62 pour 'N = 64' et 'br = $\pi \cdot 10^{-5}$ ' contre une erreur de -21.49 pour Algo-Iter-S pour 'NS=2'.

L'étude des résultats fait ressortir que l'algorithme Algo-Dir-S est plus sensible au bruit que Algo-Iter-S. Cependant, celui nécessite un temps de calcul plus grand.



(a) « Algo-Dir »



(b) « ALGO-ITER »

figure III.10: Etude comparative de ALGO-DIR(en pointillé) et ALGO-DIR-S en a) et en b) comparaison des résultats obtenus par ALGO-ITER (en pointillé) et ALGO-ITER-S pour différentes valeurs de 'br' et 'NS' pour 200 suites 'N=64'.

III.6 EXEMPLE

Un exemple illustratif, basé sur la reconstruction d'une image à partir de la phase dégradée par le bruit. L'image originale utilisée «LENNA» est de taille de (220X221) pixels, représentée dans la figure III.11. Cet exemple est élaboré afin de faire ressortir et prouver toutes les remarques faites précédemment, puis, sentir l'influence du bruit sur la phase dans la reconstruction de l'image au moyen des deux algorithmes (Algo-Dir-S et Algo-Iter-S).

Algo-Dir et Algo-Iter ne peuvent pas être utilisés à cause de la grande taille de l'image qui est de '48620' pixels.



figure III.11: Image d'origine de taille 221X220 pixels.

La figure III.12. représente les images reconstruites pour une longueur de segment 'NS=4'.

- a) et b) représentent les images reconstruites pour des bruits $br = \pi \cdot 10^{-5}$ et $\pi \cdot 10^{-1}$ respectivement pour Algo-Iter-S.
- c) et d) représentent les images reconstruites pour des bruits $br = \pi \cdot 10^{-5}$ et $\pi \cdot 10^{-1}$ respectivement pour Algo-Dir-S.

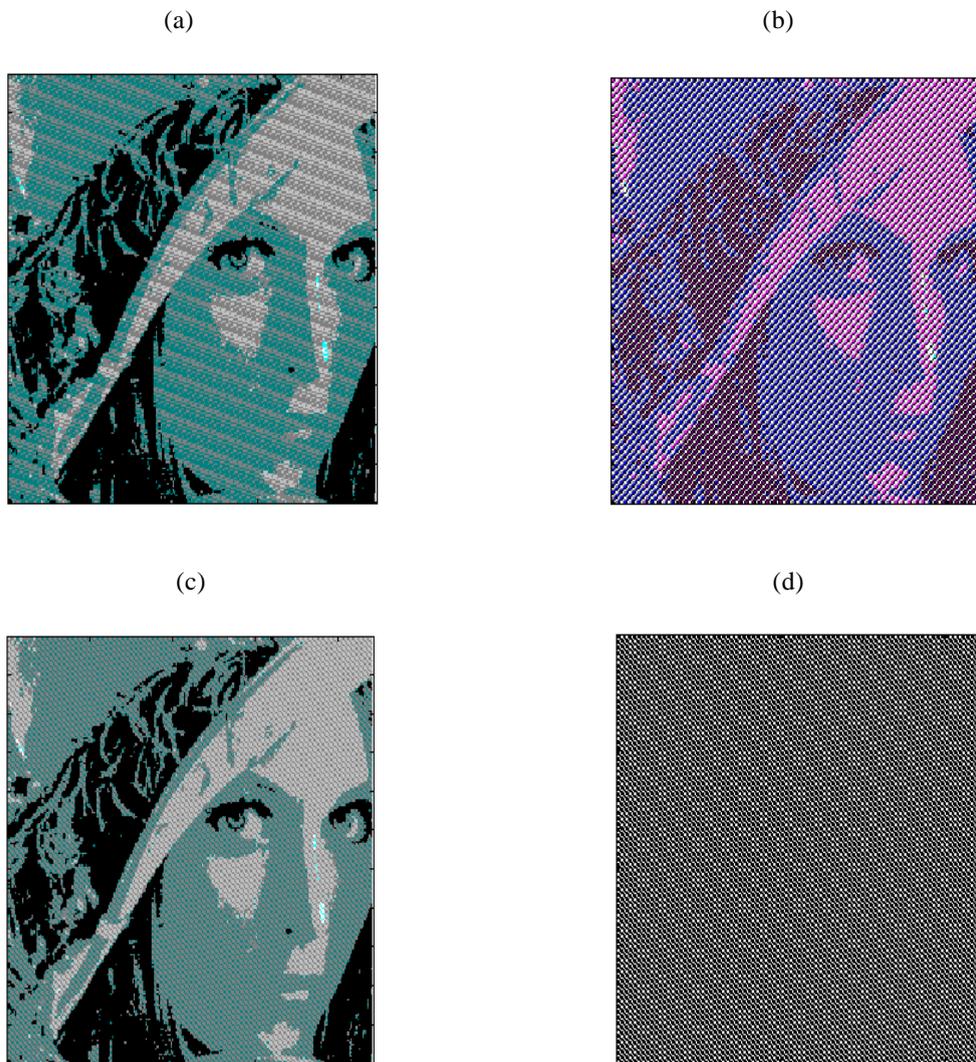


Figure III.12.: Image reconstruite en a) et b) par ALGO-ITER-S pour $br = \pi \cdot 10^{-5}$ et $\pi \cdot 10^{-1}$ respectivement ($NS=4$ et N -Iter=10) en c) et d) images reconstruite par ALGO-DIR-S pour $br = \pi \cdot 10^{-5}$ et $\pi \cdot 10^{-1}$ respectivement ($NS=4$).

La figure III.13. représente les images reconstruites pour une longueur de segment 'NS=32'.

- a) et b) représentent les images reconstruites pour des bruits $br = \pi \cdot 10^{-5}$ et $\pi \cdot 10^{-1}$ respectivement pour Algo-Iter-S.
- c) et d) représentent les images reconstruites pour des bruits $br = \pi \cdot 10^{-5}$ et $\pi \cdot 10^{-1}$ respectivement pour Algo-Dir-S.

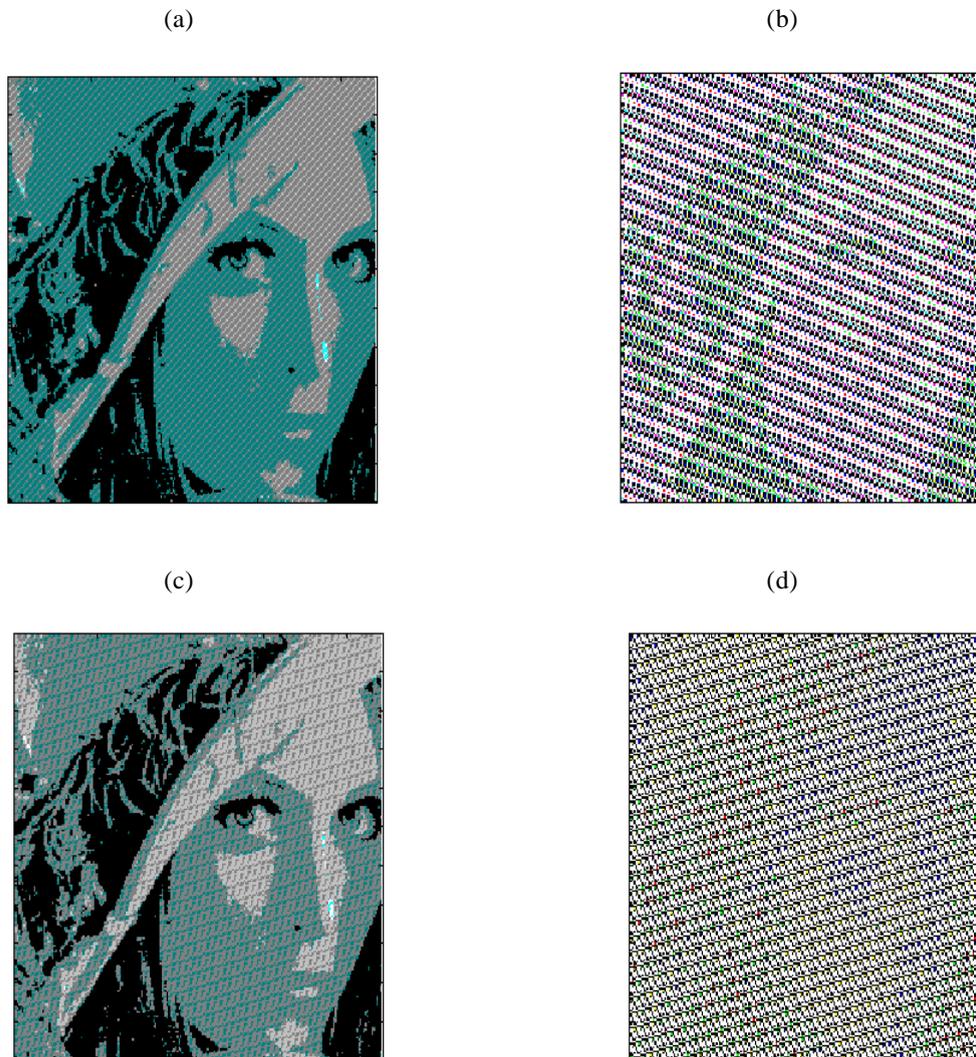


Figure III.13: Image reconstruite en a) et b) par ALGO-ITER-S pour $br = \pi \cdot 10^{-5}$ et $\pi \cdot 10^{-1}$ respectivement ($NS=32$ et $N-Iter=10$) en c) et d) images reconstruite par ALGO-DIR-S pour $br = \pi \cdot 10^{-5}$ et $\pi \cdot 10^{-1}$ respectivement ($NS=32$).

On remarque que pour un bruit ' $br = \pi \cdot 10^{-5}$ ', l'image de Algo-Dir-S est meilleure que celle de Algo-Iter-S, tandis que pour un ' $br = \pi \cdot 10^{-1}$ ', l'image de Algo-Iter-S est nettement plus différenciable du bruit que celle de Algo-Dir-S.

On en déduit que l'algorithme direct est très performant dans la reconstruction des signaux à partir d'une phase bruitée à faible niveau. Cependant, pour des niveaux de bruit élevés, l'erreur de reconstruction devient plus grande que celle obtenue par Algo-Iter-S.

CONCLUSION

Dans ce chapitre, nous avons étudié les algorithmes standards « Algo-Dir » et « Algo-Iter » dans un milieu bruité (figure III.2 et III.6 et tableau III.3 et III.7). Par la suite, nous avons effectué la même étude en utilisant cette fois-ci les algorithmes utilisant une segmentation.

L'étude comparative des résultats a montré que ces derniers présentent de réels avantages. En effet, en plus du fait que ceux-ci nécessitent une quantité de calcul plus petite, et donc, des temps machines moins prohibitifs, ils engendrent une erreur de reconstruction plus petite. En effet, nous avons remarqué que l'erreur due au bruit pour une suite de longueur 'N' en utilisant 'Algo-Dir' ou 'Algo-Iter', est équivalente à celle d'une suite de longueur 'M' quelconque ($M \gg N$) reconstruite à l'aide de 'Algo-Dir-S' et 'Algo-Iter-S' avec une segmentation, avec 'NS=N' (figures III.4 et III.8 et tableaux III.5 et III.9).

CONCLUSION

On peut subdiviser les travaux contenus dans ce mémoire en deux parties: l'une concerne l'étude des algorithmes standards (Algo-Dir et Algo-Iter (CHAP.I)) et les algorithmes utilisant une segmentation (Algo-Dir-S et Algo-Iter-S (CHAP.II), l'autre partie concerne l'étude de ces algorithmes en présence du bruit blanc (CHAP.III). Dans la première nous avons mis en relief que les algorithmes standards (Algo-Dir et Algo-Iter) étaient limités lorsqu'on voulait reconstruire des signaux de taille importante.

Les algorithmes utilisant une segmentation possèdent des performances plus intéressantes tant en erreur de reconstruction qu'en temps de calcul nécessaire.

Dans la deuxième partie qui concerne l'étude de ces algorithmes en milieu bruité, nous avons retrouvé une partie des résultats de la première partie. En effet l'étude expérimentale avec un bruit blanc additif montrée que la reconstruction était d'autant meilleure que la longueur NS du segment était plus petite, on observe d'ailleurs que l'erreur de reconstruction pour une suite de longueur M quelconque segmentée avec NS est équivalente à celle obtenue avec un algorithme sans segmentation pour une suite de longueur N=NS. Ceci explique l'intérêt de l'approche de la reconstruction en utilisant une segmentation. Nous dirons aussi que l'étude expérimentale à montrée (CHAP III) que l'algorithme itératif avec segmentation (Algo-Iter-S) est plus robuste que l'algorithme direct avec segmentation(Algo-Dir-S).

BIBLIOGRAPHIE

- [1] A. V. OPPENHEIM and J. S. LIM, « The Importance of Phase in Signals », Processings of the IEEE, VOL. 69, No. 5, May 1981.
- [2] M. H. HAYES, J. S. Lim and A. V. OPPENHEIM, «Signal Reconstruction from Phase or Magnitude», IEEE Trans. On Acoustics, Speech and Signal Processing, VOL. ASSP-28, No. 6, December 1980.
- [3] C. Y. ESPY and J. S. Lim, «Effects of Additive Noise on Signal Reconstruction from Fourier Tranform Phase », IEEE Trans. On Acoustics, Speech and Signal Processing, VOL. ASSP-31 , No. 4 , August 1983.
- [4] B. V. K. V. KUMAR , «Effect of Signal Bandwidth on the Accuracy of Signal Reconstruction from its Phase », IEEE Trans. On Acoustics, Speech and Signal Processing, VOL. ASSP-32, No. 6, December 1984.
- [5] M. H. HAYES, «The Reconstruction of Multidimensional Sequences from Phase or Magnitude of its Fourier Transform», IEEE Trans. On Acoustics, Speech and Signal Processing, VOL. ASSP-30, No. 2, December 1982.
- [6] L. R. RABINER and B. GOLD, « Digital Signal Processing », Prentice Hall, 1975.
- [7] A. V. OPPENHEIM & R. W. SCHAFER, « Digital Signal Processing », Prentice Hall, 1975
- [8] F. DE COULON, « Théorie et Traitement des Signaux », Traité d'Electricité VOL. VI. Edition Georgi, 1984.
- [9] J. MAX et Collaborateurs, « Méthodes et Techniques de Traitement de Signal et Application aux Mesures Physiques », TOME 1. MASSON, 1981.
- [10] M. KUNT, « Traitement Numérique des Signaux ». Presses Polytechniques Normandes, LAUSANNE, 1980.

APPENDICE

INTRODUCTION AU TRAITEMENT DE SIGNAL

INTRODUCTION

Dans ce chapitre , on a essayer de tracer les grandes lignes du traitement de signal afin de donner au lecteur les concepts de base enveloppant ce domaine , et dont cette thèse en prendra compte .

1- NOTIONS PRELIMINAIRES DES SIGNAUX

1-1- DEFINITION D'UN SIGNAL

Le signal est défini comme étant le support physique d'une information .
Mathématiquement , les signaux sont représentés par une fonction d'une ou plusieurs variables . La majorité des signaux sont fonction d'une variable , cette variable est généralement , le temps . L'information transportée par un signal se manifeste alors , par une variable au cours du temps [4] .

On pourrait dire que le traitement de signal intervient chaque fois que l'on a besoin de séparer un message du bruit qui le dégrade , on pourra ensuite retrouver au mieux le signal original , malgré la grandeur d'influence qui lui a été superposée . Le traitement de signal s'appuie sur plusieurs branches des mathématiques comme de la physique pour ses fondements théoriques ; pour la réalisation de ses différentes fonctions il fait appel à de nombreuses branches de la technologie, et ses domaines d'application sont de plus en plus nombreux car , la vie elle-même semble faite de traitement de signal[2] .

1-2- LES SIGNAUX DETERMINISTES

Ce sont des signaux où l'amplitude et la fréquence du signal obéissent à une loi connue, et que la connaissance des amplitudes implique la connaissance des fréquences et

réciproquement [3] .

1-2-1-LES SIGNAUX PERIODIQUES

Ce sont les plus simples, des fonctions sinusoïdales, créneaux, dents de scie, etc..., ainsi que toute fonction possédant une période de répétition T_0 finie . On remarquera que la représentation d'un bruit blanc « déterminé périodique » dans sa bande de fréquence , utilisé couramment pour la génération des signaux en étude de comportement vibratoire des spécimens [3] .

1-2-2- LES SIGNAUX NON PERIODIQUES

Ce sont ceux qui ne se retrouvent jamais égaux à eux-mêmes , quelle que soit leur durée , signaux que l'on rencontre aussi fréquemment en essai que les signaux périodiques , seront considérés pour des raisons pratiques liées à leur interprétation , comme des signaux périodiques de période t_0 , t_0 étant alors égale à la durée de la mesure . Cela entraîne une limitation sur la connaissance de ce signal , et engendre des erreurs [3] .

1-2-3- LES SIGNAUX TRANSITOIRES

Ils ont une durée limitée et s'expriment par :

$$x(t) = 0 \text{ pour } t < t_1$$

$$x(t) = 0 \text{ pour } t > t_2$$

La durée comprise entre t_1 et t_2 étant la durée d'existence du signal . Physiquement, il est important de connaître l'origine des temps de tels signaux , qui ne sont pas périodiques et de durée limitée , de façon à pouvoir les saisir plus facilement . En pratique , l'origine des temps est liée à la valeur d'un seuil fixé arbitrairement . Les chocs produisent des fonctions transitoires [3] .

1-3- LES SIGNAUX ALEATOIRES

Ils sont ceux dont la décomposition en amplitude et en fréquence varie à chaque instant . On ne peut pas les décrire à l'aide d'une expression analytique . On les observe à partir d'échantillons de durée finie , en recherchant des fonctions et des valeurs caractéristiques au moyen d'opérations telles que : densité de probabilité , moyennes statistiques , lois de répartition

Les signaux déterministes étant insuffisants pour décrire l'ensemble des phénomènes physiques , on a eu recours à l'étude des signaux aléatoires et les modèles probabilistes [3] .

1-3-1-LES SIGNAUX STATIONNAIRES

Un système est dit stationnaire si les moyennes ne dépendent pas de la période au cours de laquelle est faite la mesure [6] . Supposons par exemple , que l'on s'intéresse au bruit de fond d'un amplificateur placé dans une enceinte à 25°C avec entrée court-circuitée . Tout d'abord , en admettant que le vieillissement est nul , on peut

faire l'hypothèse de stationnarité . Il en résulte de cette définition que si $x(t)$ est un processus stationnaire :

$$E[x(t_1)] = E[x(t_2)] = m \text{ (fonction constante)}$$

et

$$E[x^\circ(t_1)] = E[x^\circ(t_2)] = v = \text{constante} . \quad [1]$$

avec $x^\circ(t) = (x(t) - m(t))^2$

1-3-2-LES SIGNAUX ERGODIQUES

Un système est ergodique si sa moyenne statistique et sa moyenne temporelle sont identiques [6] . L'ergodicité est une propriété extrêmement intéressante : les caractéristiques d'un processus aléatoire s'obtiennent d'un seul enregistrement , suffisamment long , au lieu d'exiger la connaissance d'un très grand nombre d'échantillons . L'ergodicité n'entraîne pas la stationnarité et réciproquement , ces notions sont indépendantes [1] .

1-4-LES SIGNAUX MIXTES

Ce sont des signaux composés de signaux déterministes et de signaux aléatoires superposés . C'est le cas le plus général d'existence des signaux . Par exemple , signal périodique correspondant à la mesure du phénomène physique auquel se superpose un bruit aléatoire [3] .

2-PROPRIETES

2-1-LA CONVOLUTION

2-1-1-DEFINITION

La convolution est l'action de perturber un système pour obtenir un autre système . Elle est l'effet que produit un instrument de mesure qui donne d'un phénomène non pas une image nette , mais une image un peu floue . L'image d'un point dans un instrument d'optique n'est jamais réellement un point mais une tâche . Cette tâche sera plus ou moins large selon la qualité de l'appareil [2] .

La réponse impulsionnelle est l'empreinte digitale du système . Si on connaît la réponse impulsionnelle d'un système supposé linéaire et le signal d'entrée on aura un problème de calcul pour retrouver le signal de sortie . La solution de ce problème est 'la convolution' .

2-1-2-EQUATIONS ET REPRESENTATION PHYSIQUE DE LA CONVOLUTION

Soit $e(t)$ un signal d'entrée quelconque , il nous est toujours possible de décomposer ce signal $e(t)$ en une suite d'impulsions de largeur Δt . Soit $h(t)$ la réponse du système à une impulsion de largeur Δt et de hauteur $1/\Delta t$; $h(t)\Delta t$ sera la réponse à une impulsion de largeur Δt et de hauteur unité . à $t = \Delta t$ on aura un signal de sortie $S(\Delta t) = e(\Delta t) h(t-\Delta t)\Delta t$

La sortie formera une suite et puisque le système est linéaire , on peut lui appliquer le théorème de superposition et la sortie sera la somme des contributions de chaque $e(k\Delta t)$ jusqu'à l'instant considéré :

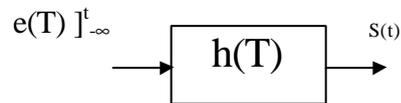
$$S(t) = \sum_{k=0}^l e(k \Delta t) h(t-k\Delta t)\Delta t$$

avec $l =$ partie entière de $t/\Delta t$

Si $h_1(t)$ est la réponse du système physique donc $h(t-k\Delta t) \equiv 0$ pour $t < k\Delta t$, car l'effet ne peut

précéder la cause .

*) Le signal de sortie à l'instant « t » ne dépend pas de la seule valeur de h(t) à l'instant « t » et de la seule valeur du signal d'entrée e(t) à l'instant « t » . Le signal de sortie à l'instant « t » dépend théoriquement de toute l'histoire du signal e(t) .



**) En fait , la réponse impulsionnelle h(T) peut être considérée comme à support borné , c'est à dire de durée finie , c'est à dire , nulle en dehors d'un intervalle (θ1,θ2) . Alors on aura :

$$S(t) = \sum_{k=t-q_2}^{t+q_1} e(k\Delta t)h(t-k\Delta t)\Delta t .$$

puisque pour les filtres physiques h(T) est nulle pour T < 0 donc l'équation de convolution devient :

$$S(t) = \sum_{k=t-q}^t e(k\Delta t)h(t-k\Delta t)\Delta t \quad \text{avec } \theta = \theta_2 - \theta_1 [2] .$$

2-1-3- CONVOLUTION ET DECONVOLUTION

Connaissant un système linéaire « S » de réponse impulsionnelle h(t) et l'histoire d'un signal « e » , introduit à l'entrée du système « S » , on pourra calculer le signal de sortie S(t) par l'équation de convolution

$$S(t) = e(t) \otimes h(t)$$

mais l'on se trouve souvent affronté à deux classes de problèmes.

*) Connaissant le signal d'entrée et le signal de sortie d'un système linéaire , peut-on déduire sa réponse impulsionnelle ?

c'est le problème de l'identification de processus .

**) Connaissant la réponse impulsionnelle d'un système linéaire et le signal de sortie , peut-on obtenir le signal d'entrée ?

C'est le problème de la déconvolution qui permettrait de corriger l'effet de convolution de tout appareil de mesure .

Ces deux problèmes reviennent à résoudre l'équation inverse de convolution . Un tel problème n'a en général pas de solution , autrement dit , sauf cas particulier , il n'existe pas de réponse impulsionnelle qui à un signal « e(t) » donné fasse correspondre un signal « S(t) » donné . Il y a là quelque chose d'extrêmement désagréable , car l'essentiel des problèmes d'identification de processus ou de mesure réside dans cette opération de déconvolution .

Si : S(t) = e(t) ⊗ h(t) a une solution (convolution) .

alors : h(t) = S(t) ⊗⊗ e(t) et e(t) = S(t) ⊗⊗ h(t) (déconvolution)

soit n'a pas toujours de solution ou a une infinité de solution .

***) En cas des systèmes réels physiques , on pourrait supposer qu'en enregistrant simultanément

le signal d'entrée $e(t)$ et le signal de sortie $S(t)$ réels d'un système réel, on puisse par déconvolution, en tirer $h(t)$, puisque le système existant, sa réponse impulsionnelle existe aussi ! Or, cette opération n'est, en général, pas réalisable, même dans le cas du système linéaire, car on ne mesure jamais exactement $e(t)$ entaché d'une $\alpha(t)$; $e'(t) = e(t) + \alpha(t)$.

De même, on ne mesure pas $S(t)$, mais $S(t) + \beta(t)$ et l'équation $h(t) = [S(t) + \beta(t)] \otimes [e(t) + \alpha(t)]$ n'est pas forcément soluble.

Sur calculateur, la solution de cette opération ne converge pas. Le problème reste actuellement entier à résoudre [2].

2-1-4-FONCTIONS PROPRES DE L'OPERATEUR CONVOLUTION

Les fonctions propres de l'opérateur convolution sont des fonctions telles que l'une d'entre elles, convoluées avec elle-même reste inchangée.

$x(t) \otimes x(t) = kx(t)$ avec k coefficient constant.

Par transformation de fourrier on aura $X(f) X(f) = kx(t)$. donc les solutions sont :

$X_1(f) = 0$ et $X_2(f) = x$ et toute combinaison linéaire des deux.

A cette fonction $X(f)$ correspond $x(t) = \sin(2\Pi.f_0.t)/2\Pi.f_0.t$

Donc, les fonctions du type « $\sin(x)/x$ » sont des fonctions propres de l'opérateur convolution.

La fonction sinus(ou cosinus) est aussi fonction propre de l'opérateur corrélation [2].

2-2- L'ECHANTILLONNAGE :

2-2-1-GENERALITES ET THEOREMES

Dans de nombreux cas, on ne traite pas directement les signaux analogiques (donc continu) fournis par les capteurs de mesure, mais on les échantillonne, c-à-d que l'on observe ces signaux non pas d'une manière continue mais à certains instants seulement. Généralement ces observations sont périodiques. La démonstration des théorèmes d'échantillonnage est rendue très simple par l'utilisation des représentations temporelle et fréquentielle des signaux.

2-2-1-1-ECHANTILLONNAGE IDEAL

Les durées d'observation sont infiniment brèves.

$$x^{\wedge}(t) = X(f) \otimes F_e \sum_{n=-\infty}^{+\infty} d(f - nfe)$$

« Pour que la répétition périodique du spectre ne déforme pas le motif répété, il faut et il suffit que la fréquence de répétition 'fe', qui est la fréquence d'échantillonnage soit égale ou supérieure au spectre de $x(t)$ à $2f_m$ » c'est le théorème de shannon $F_e \geq 2f_m$.

« On peut reconstituer une fonction à partir de ses échantillons, pourvu que la fréquence d'échantillonnage soit supérieure ou égale à deux fois la plus haute fréquence contenue dans le spectre. » c'est le théorème de reconstitution de shannon [2].

$$x(t) = \sum_{k=-\infty}^{+\infty} x(k / fe) \sin(p . fe(t - k / fe)) / p . fe(t - k / fe)$$

2-2-1-2-ECHANTILLONNAGE PAR SUITE D'IMPULSIONS PERIODIQUES DE FORME

QUELCONQUE

On réalise l'échantillonnage par un dispositif dont la réponse impulsionnelle n'est plus une distribution de Dirac mais une impulsion de durée finie ' θ '. On considère pour valeur de l'échantillon x_{kT_e} [2].

2-2-1-3-ECHANTILLONNEUR MOYENNEUR

On ne réalise plus l'échantillonnage avec impulsion infiniment brève, mais on considère au contraire une impulsion de largeur finie et on considère la valeur moyenne de $x(t)$ pendant la durée de l'impulsion [2].

2-2-1-4-ECHANTILLONNEUR BLOQUEUR

A l'instant ' kT_e ', on échantillonne $x(t)$ et on met en mémoire cette valeur pour avoir le temps, par exemple, de faire la conversion analogique-numérique. La mise en mémoire peut être très rapide, d'une durée ' θ ' beaucoup plus courte que la durée θ considérée dans le cas précédent. L'impulsion d'échantillonnage est donc une porte de durée θ [2].

2-2-2-CADENCE PRATIQUE DE L'ECHANTILLONNAGE

Pour savoir comment il convient, en pratique d'échantillonner un signal, il faut savoir ce que l'on veut faire ensuite :

*) Echantillonnage d'un signal en vue de calculs ultérieurs; pour faire des calculs sur une fonction $x(r)$ de la variable « r », il suffit des échantillons $x(kT_e)$ pourvu que le pas d'échantillonnage soit conforme au théorème de Shannon.

*) Echantillonnage du signal en vue de sa reconstitution, par exemple pour une visualisation ultérieure. Il est important de connaître la distance minimale qui doit séparer dans le temps deux échantillons successifs pour que, à partir de ces échantillons on puisse reconstituer le signal au moyen d'une interpolation relativement simple, si cette reconstitution est nécessaire. L'interpolation de Shannon, facile à réaliser, consiste en une interpolation linéaire accompagnée d'un filtrage passe-bas destiné à lisser les points anguleux à l'intersection de deux segments de droite successifs.

La cadence d'échantillonnage va donc dépendre de l'erreur que l'on tolère sur le signal reconstitué à partir de ses échantillons et, bien sûr, de la forme du signal. L'erreur dont il est question ici est une erreur déterministe et non une erreur statistique, nous le définirons donc à la manière habituelle [2].

2-2-3-LE SOUS-ECHANTILLONNAGE

Le théorème de Shannon, sous sa forme habituelle exige que l'on choisisse une fréquence d'échantillonnage $f_e > 2(f_c + \beta)$ avec β : la bande du signal. Mais, compte tenu de ce que la densité spectrale du signal considéré est nulle en dehors des bandes définies ci-dessus, on va voir qu'il est possible d'utiliser des fréquences d'échantillonnages « f_s » très inférieures à « f_e » définie par le théorème de Shannon. On peut tolérer des translatées du spectre dans l'intervalle $(0, f_c - \beta)$, et dans l'intervalle $(f_c + \beta, +\infty)$ pourvu qu'il n'y ait pas chevauchement, même partiel, avec le spectre « $S^+(f)$ » du signal avant l'échantillonnage. La translatée immédiatement à droite de « $S^+(f)$ » provient de « $S^-(f)$ » après $k+1$ translations de f_s ; celle immédiatement à gauche de $S^+(f)$

provient de $S^-(f)$ après k translations de f_s . Si aucun chevauchement ne se produit entre les diverses translations, le spectre $S^+(f)$ est conservé, ainsi que $S^-(f)$, et pour retrouver le spectre initial dans le spectre du signal échantillonné « $S(f)$ », il suffit de multiplier « $S(f)$ » par une somme de deux portes, de largeur 2β centrées sur $-f_c$ et $+f_c$. et k : l'ordre de sous-échantillonnage.

Pour que le sous-échantillonnage soit possible il faut que l'on ait : $k < (f_c - \beta)/2\beta$

On a intérêt à accroître le plus possible l'ordre de sous-échantillonnage, car de ce fait on réduit le nombre de mots du signal à traiter.

2-3-NOTIONS DE CORRELATION ET DE L'ERREUR

2-3-1-DEFINITION DE LA CORRELATION

En traitement des signaux, il est souvent nécessaire de comparer deux signaux. Ceci peut se faire de plusieurs manières. Une méthode possible, dont on fait usage, est de décaler l'un des signaux par rapport à l'autre, et de mesurer leur similitude en fonction du décalage. Mathématiquement, cette opération s'exprime par :

$$\varphi_{xy}(k) = \sum_{l=-\infty}^{+\infty} x(l)y(l+k) = \sum_{l=-\infty}^{+\infty} x(l-k)y(l) = \varphi_{yx}(-k)$$

où au moins l'un des deux signaux supposés réels est à énergie finie. Le signal $\varphi_{xy}(k)$ est appelé fonction d'intercorrélation de $x(k)$ et de $y(k)$. Si ces deux signaux sont identiques, le signal $\varphi_x(k)$ est appelé fonction d'autocorrélation de $x(k)$ [4].

La corrélation est l'opération qui consiste à vouloir relier entre elles deux fonctions, résultat de mesures de l'évolution de $x = f(t)$ et $y = f'(t)$ à partir du temps t_0 , et à rechercher leur dépendance, leur ressemblance, afin de mieux les connaître. Elle mesure l'interaction entre les deux signaux [3].

Puisque $\varphi_{xy}(k)$ est une mesure de la similitude entre $x(l)$ et $y(l)$, elle atteint son maximum pour une valeur de k , lorsque la similitude est la plus grande. Les différentes étapes et opérations intervenant dans le calcul d'une fonction de corrélation sont :

- *) Le signal $y(l)$ est décalé d'une certaine quantité k ;
- **) Le produit $x(l)y(l+k)$ est effectué échantillon par échantillon pour toutes les valeurs de l ;
- ***) Les valeurs ainsi obtenues sont additionnées pour obtenir une valeur $\varphi_{xy}(k)$.

Ces étapes sont répétées autant de fois que nécessaire [4].

Le coefficient de corrélation linéaire

$$\rho = \text{cov}(x,y) / \sigma_x \sigma_y$$

mesure la ressemblance entre les variables aléatoires x et y . Il varie entre 0 et 1.

Si $\rho = 0 \Rightarrow$ pas de relation au sens d'une moyenne entre x et y [3].

2-3-2-COEFFICIENT DE CORRELATION ET THEORIE DE L'INFORMATION

Le coefficient ou la fonction de corrélation est liée à l'énergie mutuelle d'interaction des processus décrits par les deux fonctions $x(t)$ et $y(t')$, t et t' différant d'une valeur positive, négative ou nulle et les fonctions $x(t)$ et $y(t')$ pouvant être identiques (cas de l'autocorrélation). Cette notion d'énergie mutuelle d'interaction est beaucoup plus valable que les mots ambigus ou imprécis de relation ou de dépendance. L'histoire d'un

phénomène est $x = f_1(t)$ et d'un autre $y = f_2(t)$ entre 0 et T . L'élimination de t entre ces deux équations va conduire à une relation entre x et y , il serait donc absurde de dire que x et y sont indépendants , mais si le produit moyen est nul on peut affirmer qu'il n'y a pas globalement échange d'énergie , donc l'énergie mutuelle d'interaction est nulle ; l'énergie qui passe d'un processus à l'autre pendant un intervalle « dt » est compensée en moyenne par l'énergie échangée dans l'autre sens pendant un autre intervalle « dt ». Il peut y avoir échange d'information sans transfert d'énergie; ce transfert d'énergie peut être extrêmement petit , mais il existe . Ceci signifie que si deux processus n'ont pas d'énergie mutuelle d'interaction , toute connaissance sur l'un d'eux ne peut rien nous rapporter sur la connaissance de l'autre . Donc , si deux processus ont un coefficient de corrélation nul , cela ne signifie pas qu'ils soient indépendants ,mais qu'en moyenne l'information qu'ils échangent est nulle . La quantité d'information relative à un processus « x » contenue dans un processus « y » est liée au coefficient d'intercorrélation de ces deux processus [2].

2-3-3-LES ERREURS EN TRAITEMENT DE SIGNAL

IL faut conserver à l'esprit les limitations que les méthodes d'analyse possèdent intrinsèquement . Il convient aussi de savoir que pour une large part , ces limitations ont donné lieu à des adaptations qui en réduisent les effets . La maîtrise complète des conditions de mesure requiert une parfaite connaissance des caractéristiques des instruments d'analyse utilisées , ainsi que des autres limitations dues au spécimen par lui-même et dues au condition d'exploitation . Si les sources d'erreurs dégradent l'information initiale , il faut noter que lorsque les données numérisées sont traitées par un calculateur(16 bits) possédant une meilleure définition de calcul que les convertisseurs (typiquement 12 bits) , un gain d'information apparaît , appelé gain de calcul qui permet de gagner systématiquement $\frac{1}{2}$ bit dans la définition des fonctions moyennées . Il y a aussi des erreurs liées à l'échantillonnage du signal surtout pour l'analyseur non équipé de filtres analogiques antirepliement . Lors de l'analyse des constituantes d'un signal sinusoïdal , si durant l'acquisition , on peut disposer d'un nombre entier de périodes , la raie observée sur le spectre ne sera pas entachée de l'erreur de troncature aux bords de cette fenêtre d'observation . Si par contre , ce nombre de périodes n'est pas entier , la raie observée peut être entachée d'erreurs pouvant atteindre 36% . Pour atténuer ces effets , on applique au signal acquis une pondération . Les courbes de pondération sont aussi connues sous le nom de fenêtres de pondération . Une fenêtre de HANNING permet de réduire l'erreur maximale d'estimation d'amplitude à 15.2% . L'application d'une double fenêtre de HANNING réduit l'erreur d'estimation à 9% [3] .

3-1-TRANSFORMATION DE FOURIER_ [1]

Lorsque l'on considère un signal quelconque , il est indispensable d'avoir présent à l'esprit deux représentations possibles de ce signal , une représentation-temps , c'est-à-dire une représentation de la forme $y=f(t)$ dans laquelle la variable indépendante est la durée qui s'écoule , et une représentation-fréquence de la forme $Y=F(f)$ dans laquelle la variable indépendante est la fréquence f (dont la dimension est l'inverse d'un temps) .

Ces deux représentations sont reliées entre elles par la transformation de Fourier et il est indispensable , pour pouvoir utiliser les méthodes de traitement du signal , de savoir manipuler ces deux représentations complémentaires .

Ceci explique l'ampleur donnée à ce chapitre sur la transformation de Fourier , outil d'une importance capitale .

Notons également que cette transformation définie ici dans le domaine temps-fréquence peut s'étendre à tous les domaines . Si la variable indépendante est une longueur , dans le domaine associé , ce sera l'inverse d'une longueur , etc .

3-1-1-Transformation de Fourier des fonctions périodiques

Soit $x(t)$ une fonction de t périodique , de période T .

On montre que $x(t)$ peut s'écrire :

$$x(t) = a_0 / 2 + \sum_{n=1}^{\infty} [a_n * \cos((2 * \Pi / T) * n * t) + b_n * \sin((2 * \Pi / T) * n * t)]$$

$$x(t) = a_0 / 2 + \sum_{n=1}^{\infty} [a_n * \cos((2 * \Pi * f_0) * n * t) + b_n * \sin((2 * \Pi * f_0) * n * t)]$$

avec

$$f_0 = 1 / T.$$

Les coefficients a_n et b_n sont donnés par :

$$a_n = 2 / T \int_{-T/2}^{T/2} x(t) * \cos(2 * \Pi * f_0 * n * t) dt$$

$$b_n = 2 / T \int_{-T/2}^{T/2} x(t) * \sin(2 * \Pi * f_0 * n * t) dt$$

Si l'on pose

$$X(n * f_0) = 1 / 2(a_n - j b_n)$$

On a

$$X(n * f_0) = 1 / T \int_{-T/2}^{T/2} x(t) \exp(-2 * \Pi * j * n * f_0 * t) dt.$$

$X(n * f_0)$ est le spectre de fréquence , grandeur complexe , qui peut se décomposer en :
spectre d'amplitude :

$$|X(n * f_0)| = 1 / 2 \sqrt{a_n^2 + b_n^2}$$

spectre de phases :

$$\phi(n * f_0) = \text{tg}^{-1}(-b_n / a_n).$$

$$X(f) = \sum_{n=-\infty}^{\infty} |X(nf_0)| * \exp(-j * f * (nf_0))$$

$$x(t) = \sum_{n=-\infty}^{\infty} \exp(2\pi j n f_0 t) * 1/T \int_{-T/2}^{T/2} x(s) * \exp(-2\pi j f_0 n s) ds$$

Les valeurs négatives de n sont introduites pour rendre les équations plus simples ; on vérifie aisément que si x(t) est réel :

$$a_{-n}=a_n \text{ et } b_{-n}=-b_n .$$

Il est important de remarquer que le spectre d'une fonction périodique de période T est composé de raies dont l'écart minimum est , sur l'axe des fréquences : $f_0=1/T$.

Le spectre d'une fonction périodique est donc essentiellement discontinu , il n'existe que pour les valeurs de la fréquence multiples de $f_0=1/T$.

3-1-2-Transformation de Fourier des fonctions non périodiques

On peut considérer cette non-périodicité comme résultant d'une extension à l'infini de la période T . L'intervalle de fréquences $f_0=1/T$ tend alors vers zéro et le spectre devient alors une fonction qui peut être continue .

On a :

$$x(t) = \int_{-\infty}^{\infty} \exp(2\pi j f t) df * \int_{-\infty}^{\infty} x(s) \exp(-2\pi j s) ds$$

$$X(f) = \int_{-\infty}^{\infty} x(t) \exp(-2\pi j f t) dt$$

est appelée la transformée de Fourier de x(t) .

On écrit habituellement :

$$x(t) \leftrightarrow X(f)$$

X(f) est une fonction de f , en général complexe , qui comprend donc une partie réelle (ou en phase) et une partie imaginaire (ou en quadrature) .

$$\Re[X(f)] = \int_{-\infty}^{\infty} x(t) \cos(2\pi f t) dt$$

$$\Im[X(f)] = \int_{-\infty}^{\infty} x(t) \sin(2\pi f t) dt .$$

Le spectre d'amplitude est :

$$|X(f)| = \sqrt{\{\Re[X(f)]\}^2 + \{\Im[X(f)]\}^2} .$$

Le spectre de phase est :

$$f(f) = tg^{-1}\left(-\frac{\Im[X(f)]}{\Re[X(f)]}\right).$$

3-1-3-Signification physique de la transformation de Fourier

Lorsque l'on écrit l'équation de la transformée de Fourier :

$$X(f) = \int_{-\infty}^{\infty} x(t) \exp(-2\pi jft) dt$$

$X(f)$ et $x(t)$ représentent la même grandeur physique, mais dans une représentation différente. Si l'on considère $x(t)$, le point représentatif se déplace dans le domaine (amplitude-temps). Si l'on considère $X(f)$, le point représentatif se déplace dans le domaine (amplitude-fréquence).

Lorsque l'on cherche la valeur de $X(f)$ pour une valeur f_i de f , cela signifie que l'on cherche dans toute l'histoire (et tout le futur) de $x(t)$ ce qui correspond à la fréquence f_i . Ceci correspond à un filtrage infiniment sélectif. Un tel filtrage n'est pas réalisable physiquement.

On ne peut donc pas connaître $X(f)$ avec une localisation parfaite sur l'axe des fréquences.

De même, si l'on veut retrouver $x(t)$ à partir de $X(f)$, il faut connaître le spectre pour toutes les fréquences jusqu'à l'infini et la formule montre que c'est la même opération de filtrage infiniment sélectif qui intervient, les variables temps et fréquences étant permutées.

Cela signifie que pour connaître parfaitement la valeur de $x(t)$ à un instant t , il faut disposer d'une bande passante infinie.

3-1-4-Conditions d'existence de la transformation de Fourier

On démontre que, pour qu'une fonction $f(t)$ ait une transformée de Fourier, il faut et il suffit que : 1- La fonction $f(t)$ soit bornée (pas de valeurs infinies) ;

2- L'intégrale de $f(t)$ entre $-\infty$ et $+\infty$ ait une valeur finie ;

3- Les discontinuités de $f(t)$ ainsi que ses maxima et minima soient en nombre fini.

Mais ceci n'entraîne pas nécessairement que la réciproque soit vraie. Pour que la transformée de Fourier de $f(t)$ existe et soit réciproque, il suffit que $f(t)$ appartienne à l'espace L^2 ou espace des fonctions de carré sommable. En d'autres termes, cela signifie que ainsi que sa transformée de Fourier sont à énergie finie.

On voit, que toutes les fonctions existant physiquement et que l'on peut être amené à traiter, vérifient ces conditions, essentiellement parce qu'on les observe sur un temps fini.

Donc, nous aurons à calculer la transformée de Fourier d'un signal physique, nous n'aurons pas à nous poser le problème de l'existence de cette transformée.

3-1-5-Quelques propriétés de la transformée de Fourier

3-1-5-1-Propriétés de linéarité

La transformée de Fourier est une opération linéaire.

$$f(t) \leftrightarrow F(f)$$

$$g(t) \leftrightarrow G(f)$$

entraîne :

$$a*f(t)+b*g(t) \leftrightarrow a*F(f)+b*G(f)$$

3-5-2-Propriétés de parité

Le tableau ci-dessous les résume .

$x(t)$	$X(f)$
Réelle et paire	Réelle et paire
Réelle et impair	Imaginaire et impaire
Imaginaire et paire	Imaginaire et paire
Imaginaire et impair	Réelle et impaire
Complexe et paire	Complexe et paire
Complexe et impair	Complexe et impaire
Réelle quelconque	Partie réelle paire Partie imaginaire impaire
Imaginaire quelconque	Partie imaginaire impaire Partie réelle impaire
Partie réelle paire Partie imaginaire impaire	Réelle
Partie réelle impair Partie imaginaire paire	Imaginaire

Si $x(t)$ est quelconque (donc pouvant être complexe) , on a les relations :

$$x(t) \leftrightarrow X(f)$$

$$x^*(t) \leftrightarrow X^*(-f)$$

L'astérisque signifiant quantité complexe conjuguée .

3-1-5-3- Propriétés de similitude

L'équation

$$f(t) \leftrightarrow F(f)$$

$$f(a*t) \leftrightarrow \frac{1}{|a|} * F\left(\frac{f}{a}\right).$$

entraîne

Autrement dit , un étalement de l'échelle des temps conduit à une « contraction » de l'échelle des fréquences et inversement .

3-1-5-4-Propriétés de translation

L'équation :

$$f(t) \leftrightarrow F(f)$$

entraîne :

$$\begin{aligned} f(t-a) &\leftrightarrow \exp(-2\pi jaf) * F(f) \\ f(t-a) &\leftrightarrow \cos(2\pi fa) * F(f) - \sin(2\pi fa) F(f) . \end{aligned}$$

Ceci implique que les transformées de Fourier de $f(t)$ et de $f(t-a)$ ont le même module mais la transformée de $f(t)$ et de $f(t-a)$ subit une rotation de phase supplémentaire de $2\pi af$.
Cette propriété de translation est évidemment réciproque c'est-à-dire que :

$$F(f) \leftrightarrow f(t)$$

entraîne :

$$F(f-a) \leftrightarrow f(t) * \exp(2\pi jat) .$$

3-1-6-5-Dérivation

Si :

$$x(t) \leftrightarrow X(f)$$

$$\frac{dx(t)}{dt} \leftrightarrow (2\pi jf) * X(f)$$

$$\frac{d^n x(t)}{dt^n} \leftrightarrow (2\pi jf)^n * X(f) .$$

3-1-6-6-Les cas particuliers

a).Formule de Poisson. Si :

$$x(t) \leftrightarrow X(f)$$

$$\sum_{n=-\infty}^{+\infty} x(n) = \sum_{K=-\infty}^{+\infty} X(K) .$$

b).Distribution de Dirac. On sait que :

- $\delta(t)$ est :-une impulsion centrée sur $t=0$;
- de largeur infiniment étroite ;
- et de surface 1 .

A pour transformée de Fourier 1

$$\delta(t) \leftrightarrow 1$$

D'ou

$$\delta(t-t_0) \leftrightarrow \exp(-2\pi jft_0) .$$

3-1-7-Pourquoi la transformation de Fourier

Pourquoi avoir choisi la transformation de Fourier, c'est-à-dire une décomposition en éléments sinusoïdaux (ou cosinusoidaux ou sinusoïdaux) pour la représentation fréquentielle des signaux ? N'aurait-on pu choisir une autre décomposition, en signaux carrés par exemple ? La réponse est simple : On a choisi une décomposition en élément sinusoïdaux propres du filtre linéaire ; c'est-à-dire qu'une telle forme n'est pas déformée par passage dans un filtre linéaire . Ceci explique l'intérêt d'une telle décomposition .

3-2-TRANSFORMATION DE LAPLACE

3-2-1-Définition

La transformée de Fourier de $f(t)$ est :

$$F(f) = \int_{-\infty}^{\infty} f(t) \exp(-2\pi jft) dt$$

$F(f)$ n'existe que si cette intégrale est convergente . C'est-à-dire a une valeur finie .

S'il n'en est pas ainsi , on la rend convergente en multipliant $f(t)$ par $\exp(-\delta_0 t)$, δ_0 étant une grandeur réelle positive , appelée abscisse de convergence .

Cela revient à remplacer $2\pi jf$ par p avec :

$$p = \delta_0 + 2\pi jf$$

p est appelée fréquence complexe .

Si , de plus , la fonction $f(t)$ est identiquement nulle pour $t < 0$, on a la transformation de Laplace :

$$TL[f(t)] = \int_0^{\infty} f(t) \exp(-pt) dt.$$

La transformée de Laplace trouve son intérêt dans l'étude des régimes transitoires qui , justement , sont toujours nuls pour $t < 0$ en vertu du principe de causalité (l'instant zéro étant l'instant où apparaît l'excitation qui génère le transitoire) qui veut que l'effet ne puisse exister avant la cause .

La transformation de Laplace est un outil bien connu des servomécaniciens et des physiciens .

Un autre intérêt de la transformation de Laplace est que , par des passages à la limite portant sur l'abscisse de convergence , on arrive à la transformée de Fourier de fonctions qui n'y sont pas accessibles directement .

Rappelons aussi que la transformée de Laplace de la réponse impulsionnelle d'un système linéaire en est la fonction de transfert .

L'équation de convolution :

$$s(t) = e(t) \otimes h(t)$$

S'écrit , par transformation de Laplace :

$$W_s(p) = W_e(p) * W_h(p).$$

3-3-TRANSFORMEE DE FOURIER DES SIGNAUX DISCRETS

L'idée d'exprimer une fonction de forme compliquée, comme une combinaison linéaire de fonctions élémentaires de forme simple, est bien connue. Par exemple, dans un intervalle $[u_1, u_2]$, une fonction $\theta(u)$ peut être représentée par :

$$q(u) = \sum_{i=0}^{\infty} a_i * y_i(u)$$

Où les fonctions $\Psi_i(u)$ forment un ensemble de fonctions élémentaires de forme simple. Si ces fonctions sont orthogonales, alors les coefficients α_i sont indépendants les uns des autres. On parle, dans ce cas, du développement en série de fonctions orthogonales.

Comme mathématiquement les signaux sont représentés par des fonctions, ceci est évidemment valable pour les signaux aussi bien numériques qu'analogique. D'une manière tout à fait conforme à la définition d'un signal numérique, on peut interpréter la suite des coefficients α_i comme un signal numérique représentant le signal analogique $\theta(u)$.

De tous les développements en série de fonctions orthogonales, la série de Fourier est sans doute celle dont on fait le plus grand usage en traitement des signaux.

Comme on le sait, on peut généraliser la notion de série de Fourier pour représenter une fonction dans un intervalle infini. Cette généralisation aboutit à la transformation intégrale de Fourier, dont la définition est rappelée ci-dessous:

$$X(f) = \int_{-\infty}^{+\infty} x(t) \exp(-2\pi jft) dt$$

La transformée inverse est donnée par:

$$x(t) = \int_{-\infty}^{+\infty} X(f) \exp(2\pi jft) df$$

3-3-1-Definition

La transformée de Fourier des signaux numériques est définie par:

$$X(f) = \sum_{n=-\infty}^{+\infty} x(n) \exp(-2\pi jfn)$$

D'après cette définition, la fonction $X(f)$ est généralement une fonction complexe de la variable réelle continue f .

3-3-2- Existence de la transformée de Fourier des signaux discrets

La transformée de Fourier $X(f)$ existe si le membre de la relation précédente est fini, c'est-à-dire si la série converge. Comme le module du terme $\exp(-2\pi jn)$ est toujours égal à 1, la condition suffisante pour que $X(f)$ existe est:

$$\sum_{n=-\infty}^{+\infty} |x(n)| < \infty \quad (5-5)$$

Si cette condition est satisfaite, alors la série converge absolument vers une fonction continue de f . La classe des signaux qui satisfont la condition (5-5) est celle des signaux qui sont aussi à énergie finie. Ceci découle directement de l'inégalité:

$$\sum_{n=-\infty}^{+\infty} |x(n)| \delta \leq \left[\sum_{n=-\infty}^{+\infty} |x(n)| \right] \delta \quad (5-6)$$

Le membre de gauche de cette inégalité est par définition l'énergie du signal $x(n)$. Donc, si la condition (5-5) est : (5-6) l'est aussi. Les signaux satisfaisant (5-5) sont appelés signaux absolument sommable.

Par contre, un signal à énergie finie, n'est pas nécessairement absolument sommable.

La condition (5-5) est une condition suffisante pour l'existence de la transformée $X(f)$. C'est pourquoi elle est sé condition nécessaire est sujette à de nombreuses définitions et interprétation. Nous admettrons, sans démonstrati

4-NOTIONS PHYSIQUES DES PHENOMENES ALEATOIRES « BRUITS »

4-1-INTRODUCTION

Le principal champ d'application du traitement de signal et des méthodes de corrélation étant les phénomènes aléatoires, il paraît indispensable de tenter de préciser ces notions physiques de phénomènes aléatoires. Il existe de nombreux phénomènes physiques pour lesquels les grandeurs que l'on observe sont reliées aux conditions extérieures par des lois facilement décelables. Par exemple, le mouvement d'une particule soumise à une force aisément calculable et peut être entièrement « prédit » par la connaissance des conditions initiales et des « forces motrices ». Considérons, par exemple, le bruit dans un circuit électronique : si nous essayons de relier la valeur instantanée de la tension de ce bruit au mouvement des électrons, phénomène qui engendre ce bruit, il apparaît immédiatement que nous nous trouvons en présence d'un problème actuellement insoluble, ceci pour des raisons à la fois théoriques et pratiques. Si nous voulons, néanmoins, pouvoir faire intervenir ce bruit quantitativement, il nous faudra le caractériser par des concepts autres que sa valeur instantanées ; Ce seront ses caractéristiques statistiques. Un phénomène cesse d'être aléatoire dès lors que nous maîtrisons les lois qui le régissent. La très grande puissance de la théorie des fonctions aléatoires nous donne le moyen d'approcher la connaissance de ces phénomènes, dont les lois exactes nous échappent, et nous permet de les faire intervenir dans le calcul de leurs conséquences.

Tout signal peut, selon le point de vue envisagé, être considéré comme aléatoire, alors qu'un bruit peut être riche d'information [3].

4-2-LES LOIS DE PROBABILITES SUR LA VARIABLE ALEATOIRE

4-2-1-LA LOI UNIFORME

Soit « A » une partie quelconque de l'ensemble R des nombres réels, de longueur $v(A)$; on dit qu'une variable aléatoire « X » est de loi uniforme sur « A », ou encore équirépartie sur « A », si elle est continue, avec la densité de probabilité $f(x)$ suivante :

$$f(x) = 0 \quad \text{si } x \notin A$$
$$f(x) = 1/v(A) \quad \text{si } x \in A$$

par exemple, si A est l'intervalle (a,b) avec $a < b$,
 $f(x) = 1/(b-a)$ pour $x \in A$ [5]

4-2-2-LA LOI DE POISSON

Une variable aléatoire « X » est dite obéir à la loi de POISSON de paramètre m , si elle est discontinue , ses valeurs possibles $x_k(k=0,1,2,...)$ étant les entiers non-négatifs $x_k = k$; et si l'on a :

$$P_k = \Pr(X = k) = \exp(-m) * m^k / k! \quad (k = 0,1,2,...)$$

m: est la moyenne de X .

Il y a donc une infinité dénombrable de valeurs possibles pour X ; le paramètre m peut avoir une valeur quelconque , mais toutefois supérieure à zéro ; $m < 0$ n'est pas admissible , car cela donnerait de $P_k < 0$ pour k impair ; quant à la valeur $m=0$, elle est admissible , mais alors $\Pr(X=0)=1$; autrement dit , X se réduit au nombre presque-certain 0 ; nous dirons alors qu'il s'agit d'une loi de POISSON dégénérée . Dans le cas non dégénéré $m > 0$ et $P_k > 0$ on a :

$$\sum_{k=0}^{+\infty} P_k = \sum_{k=0}^{+\infty} \exp(-m) * m^k / k!$$

et

$$E(X) = \sum_{k=0}^{+\infty} k * \exp(-m) * m^k / k!$$

puisque le premier terme de cette série (k=0) est nul alors on a :

$$E(X) = \exp(-m) * \sum_{k=1}^{+\infty} m^k / (k-1)! = \exp(-m) * m * \sum_{k=1}^{+\infty} m^{(k-1)} / (k-1)!$$

or:

$$\sum_{k=1}^{+\infty} m^{(k-1)} / (k-1)! = \exp(-m) ;$$

d'où $E(X) = m$ et de même on peut calculer

$$E(X^2) = m^2 + m \quad [5].$$

On trouve pour cette loi des tables de probabilités individuelles ainsi que des tables de probabilités cumulées ; on entre dans ces tables par les valeurs de « m » et par celles de k . L'extension de ces tables croît donc en même temps que les valeurs de m que l'on désire atteindre . Deux propriétés permettent cependant de limiter cette extension :

*) Lorsque m augmente , la loi de POISSON tend vers une loi normale de moyenne « m » et d'écart type « $m^{1/2}$ » . Il suffit alors de passer en variable réduite : $u = (k-m)/m^{1/2}$

et de rechercher la valeur de « u » dans les tableaux de la fonction de répartition de la loi normale centrée réduite , tout en tenant compte du caractère discontinu de la loi de POISSON .

**) En passant par l'intermédiaire d'une fonction auxiliaire qui suit une loi de « χ^2 » à « 2l » degrés de liberté que pour $l > 15$ on a :

$$\text{Probabilité } [k > l] = F(2 * m^{1/2} - (4 * l - 1)^{1/2}) \text{ et}$$

$$\text{Probabilité } [k < l] = 1 - F(2 * m^{1/2} - (4 * l + 3)^{1/2}).$$

où « F » est la fonction de répartition de la loi normale centrée réduite . La loi de POISSON est une approximation de la loi binomiale .

L'usage de la loi de POISSON est lorsque « $P < 0.1$ et $n \gg$ de l'ordre de 50 à 100 » , la simplification des calculs donne son intérêt à l'emploi de la loi de POISSON dans ces circonstances . Un autre emploi de cette loi est les comptages de radioactivité d'une source , mesurés dans des intervalles égaux , sont distribués suivant la loi de POISSON de variance empirique connue , puisqu'elle est égale à la racine carrée du nombre d'impulsions de comptage pendant l'intervalle de temps en cause [3].

4-2-3-LA LOI NORMALE, LOI DE LAPLACE-GAUSS OU LOI DE GAUSS

Une variable aléatoire « X » obéit à la loi de GAUSS , si X est continu , de densité de probabilité :

$$f(x) = (1/\sigma*(2*\Pi)^{1/2})*\exp(-(x-m)^2/2*\sigma^2) \quad \sigma : \text{est l'écart type}$$

$$\sigma = \lim_{n \rightarrow \infty} (\sum (x_i - m)^2/n)^{1/2} .$$

La loi normale centrée réduite correspond à une moyenne nulle et une variance unité .Sa valeur numérique peut être trouvée dans les tables classiques . Toutes les lois normales peuvent s'en déduire grâce au changement de variable : $u = (x-m)/\sigma$. [3]

4-2-4-LA FORMULE DE BAYES

Supposons que la variance conditionnelle (v) ne comporte qu'un nombre fini ou une infinité dénombrable d'éléments « v » , soient les :v1,v2,v3,...vk,...;

$$v(X/v) = E(X^2/v) - [E(X/v)]^2 .$$

alors pour tout événement « E » et pour tout « h » :

$$\Pr(vh \cap E) = \Pr(vh)*\Pr(E/vh); \text{ et aussi } \Pr(vh \cap E) = \Pr(E)*\Pr(vh/E) ;$$

$$\text{or : } \Pr(E) = \sum_k \Pr(vk) * \Pr(E / vk);$$

$$\text{on en tire que : } \Pr(vh/E) = (\Pr(vh)*\Pr(E/vh))/ \sum_k \Pr(vk) * \Pr(E / vk)$$

qui est la célèbre formule de BAYES ,ou plutôt une forme élémentaire du théorème de BAYES .

Mais supposons maintenant que v est quelconque , donc on a , en désignant par A l'événement quelconque ($A \subset v$ et $A \in \Sigma$, d'où Σ la famille des événements de A)

$$\Pr(A/E) = \frac{\int \Pr(E / v) * q(dv)}{\int \Pr(E / v) * q(dv)} \quad q(A) = \Pr(A) \text{ et } q(dv) \text{ est la loi de probabilité de } V \text{ d'où } V \text{ est}$$

l'élément aléatoire dans v . Telle est la forme générale de la formule de BAYES [5] .

4-2-5-LA LOI DES GRANDS NOMBRES

Soit {Xk } (k=1,2,3,...) une suite indéfinie de variables aléatoires ;

posons $S_k = X_1 + X_2 + X_3 + \dots + X_k$, $Y_k = S_k / k = 1/k(X_1 + X_2 + \dots + X_k)$;

On appelle loi des grands nombres tout énoncé établissant , naturellement sous certaines conditions , que lorsque $k \rightarrow + \infty$ la moyenne arithmétique Y_k des k premiers « Xj » tend stochastiquement vers une variable aléatoire limite « Y » .

Supposons que pour tout « j » , « Xj » est une variable aléatoire du second ordre posons :

$$E(|X_j|^2) = \sigma_j^2 , (j = 1,2,3,...);$$

$$a_{hk}^2 = E(| \sum_{j=h+1}^{h+k} X_j |^2) \text{ avec } (h= 0,1,2,3,...; k=1,2,3,...);$$

$$\text{on a } E(|Y_k|^2) = a_{0k}^2/k^2 [5] .$$

4-2-6-LA LOI DE PEARSON OU LOI DU X²

Soit v variables aléatoires normales centrées, réduites, indépendantes, X_1, X_2, \dots, X_v . Considérons la somme : $X^2 = X_1^2 + X_2^2 + \dots + X_v^2$

La variable aléatoire X^2 suit ce que l'on appelle une loi de X^2 à v degrés de liberté. C'est une loi continue, si on remplace les « X_i » par les « Y_i » tels que :

$Y_i = (X_i - \mu_i) / \sigma_i$ alors la somme des carrés des Y_i suivra la loi du X^2 .

Il arrive aussi que la représentation d'une variable aléatoire par une loi de X^2 provient d'une comparaison purement empirique réalisée à l'aide de certains tests ; si la comparaison est jugée satisfaisante, la distribution expérimentale pourra bénéficier des propriétés de la loi de X^2 : tables de valeur, estimations de la moyenne et de la variance, et autres [3].

4-2-7-LA LOI DE STUDENT, LOI DE FISHER-STUDENT OU LOI DE « t »

C'est la loi de probabilité de la variable d'expression :

$t = x / (X^2/v)^{1/2}$ ou x : variable aléatoire normale réduite

X^2 : variable aléatoire suivant la loi de X^2 et v degrés de liberté

Le calcul à partir de l'expression algébrique de la densité de probabilité de la loi de « t » à v degrés de liberté, donne pour sa variance :

$\sigma_s^2 = v / (v-2)$ (n'est valable que pour $v > 2$).

Les principaux usages de la loi de « t » sont la comparaison des moyennes et le calcul de l'intervalle de confiance d'une moyenne [3].

4-2-8-LA LOI DE FISHER-SNEDECOR OU DE SNEDECOR

C'est la loi de probabilité de la variable aléatoire définie par l'expression

$F = (X_1^2/v_1) / (X_2^2/v_2)$

ou X_1^2 et X_2^2 : variables aléatoires suivant la loi de X^2 respectivement à v_1 et v_2 degrés de liberté.

La variable « F » est une variable continue ne pouvant prendre que des valeurs positives ou nulles. Elle ne présente pas de symétrie ; pour des valeurs données de v_1 et de v_2 elle a la même allure générale que la courbe de la densité de X^2 .

Le principal usage de la loi de SNEDECOR est la comparaison des variances empiriques obtenues expérimentalement par deux processus différents [3].

4-2-9-LA LOI BINOMIALE

C'est une loi de variable discontinue qui concerne les problèmes où celle-ci ne peut prendre que deux valeurs distinctes A et B. On désigne par « P » la probabilité pour trouver A, et par « Q=1-P » celle de trouver B. On répète n fois l'essai, la variable aléatoire est le nombre de fois « r » que le résultat A est apparu. L'expression de la loi binomiale est :

$P[X=r] = C_n^r * P^r * Q^{n-r}$ avec $C_n^r = n! / r! * (n-r)!$

en supposant que « P » n'a pas été modifiée par les essais antérieurs. Donc, la moyenne

$m = n * P$ et la variance $\sigma^2 = n * P * Q$.

Un des principaux emplois de la loi binomiale concerne les prélèvements de contrôle de réception, avec acceptation ou refus de la pièce extraite. Elle sert également à suivre la régularité de la qualité d'une fabrication dans le cadre d'un plan d'expérience [3].

conclusion

A la fin de ce petit résumé , on pense qu'avec la bonne compréhension de ces principes , on pourra décoller dans le domaine du traitement du signal y compris la reconstitution , avec beaucoup de confiance et sans complexe .