

REPUBLIQUE ALGERIENNE DEMOCRATIQUE & POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Institut D'Aéronautique De Blida
Département De Navigation Aérienne



Projet de fin d'études
en vue de l'obtention du diplôme d'ingénieur d'état en aéronautique

Option : Installations

*Elaboration d'un Editeur sous Matlab
pour le traitement numérique des signaux*

Réalisé par :

Dirigé par : M^{ieur} M.HELLAL

MECHTOUF Toufik
SINGARE Mohamadou

-----Promotion 2002/2003-----

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Dédicaces

Je dédie ce travail à :

- Ma chère maman et mon cher papa pour leurs bénédictions et leur soutien tant moral que matériel ;
- Ma charmante épouse qui m'a soutenu et encouragé tout au long de mon cursus ;
- Mes frères et sœurs qui m'ont épaulé pendant mes moments difficiles ;
- Mes amis qui m'ont beaucoup aidé à travers leurs conseils ;
- Mes beaux parents pour leurs bénédictions et leur encouragement.

SINGARE Mohamadou

Je dédie ce travail à :

- Ma chère mère et feu mon père (paix à son âme) pour leurs encouragements et leur soutien tant moral que matériel ;
- Ma petite famille sans oublier ma petite sœur RANYA et mes frères MEIIDI et ROCIIDI ;
- Ma grande famille, la famille MECHTOUF et KHAZRI ;
- Mes beaux frères ;
- Mes amis sans exception ;
- tous les gens de SKIKDA et surtout les habitants de la cité BENI MALEK;
- tous ce qui sont proches de mon cœur.

MECHTOUF Toufik

Remerciements

Nous tenons tout d'abord à remercier notre promoteur M^r HELAL, son soutien sans faille, ses encouragements et ses conseils nous ont permis de mener à bien ce travail. Nous avons toujours apprécié la confiance et la liberté qu'il nous a accordées.

Nous désirons vivement remercier les membres du jury ainsi que tout le corps professoral de l'institut d'aéronautique de Blida.

Nos remerciements vont également à l'endroit de nos camarades de la promotion 2002 2003 et en particulier à STAMBOULI Zinédine pour son aide à la réalisation de ce logiciel.

Enfin, merci à nos proches, familles et copains qui ont également participé à leur manière à la réalisation de ce Mémoire.

RESUME

L'objectif principal de ce projet est d'établir le traitement numérique des signaux. Pour cela, nous avons élaboré un éditeur de signal sous MATLAB en raison des performances graphiques et de la convivialité qu'offre ce langage. Cet éditeur permet de traiter beaucoup de sortes de signaux.

Pour les signaux simples, il permet de visualiser les spectres d'amplitude et de phase par la FFT, de calculer la fréquence d'échantillonnage, de zoomer et d'écouter le signal.

Pour les signaux de paroles, en plus des traitements cités pour les signaux simples, l'éditeur permet la visualisation du sonagramme (c'est-à-dire le graphique représentant les composantes acoustiques à savoir la durée, la fréquence et l'intensité de la voix), du cepstre, la détection du pitch (fondamental) et de la fréquence des formants.

SUMMARY:

The principal objective of this project is to establish the digital processing of the signals. For that, we worked out an editor of signal under MATLAB because of the graphic performances and the user-friendliness which this language offers. This editor allows to treat many kinds of signals.

For the simple signals, it makes it possible to visualize the spectra of amplitude and phase by the FFT, to calculate the zoomer, sampling rate and to listen to the signal.

For the signals of words, in addition to the treatments quoted for the simple signals, the editor allows the visualization of the sonogram (i.e. the graph representing the acoustic components with knowing the duration, the frequency and the intensity of the voice), of the cepstre, the detection of the pitch (fundamental) and the frequency of the formants.

الهدف الرئيسي من هذا المشروع، هو تحقيق المعالجة الرقمية للإشارات بأدوات الحاسوب و لهذا قد قمنا بإنجاز برنامج يظهر الطيف بالمرشحات الرقمية، تكبير و تصغير و اشتقاق و تكامل الإشارة و الاستماع إليها

لقد تعاملنا من لغة برمجة خاصة هي " المطلباب "، هذه المذكرة تعد خطوة أساسية في مجال التحسس الصناعي للإشارة الصوتية.

SOMMAIRE

INTRODUCTION GENERALE	1
I. GENERALITES SUR LE MATLAB:	2
I-1-Introduction	2
I-2-Organisation du logiciel Matlab	2
I-3-Réalisation d'interfaces graphiques	3
1. Les contrôles.....	3
1.1. Les types de contrôles	3
1.2. Création de contrôles	5
1.3. Propriétés des contrôles.....	5
2. Les menus.....	8
2.1. Création de menus	8
2.2. Les propriétés des menus.....	8
II. SYNTHESE DES SIGNAUX SIMPLES	9
II.1 Introduction.....	9
II.2 Description	9
II.3 Procédures et fonctions.....	9
II.3.1 Procédure d'entrer en matière « affichage ».....	10
II.3.2 Procédure choix du type de signal « choix ».....	10
II.3.3 Procédure du tracé des fonctions	10
II.3.3.1 Procédure de tracé de l'impulsion rectangulaire.....	13
II.3.3.2 Procédure de tracé de l'impulsion triangulaire	13
II.3.3.3 Procédure de tracé de l'impulsion exponentielle.....	13
II.3.3.4 Procédure de tracé de la fonction DDS (Dents De Scie).....	14
II.3.3.5 Procédure de tracé de la fonction somme de sinusoïdes.....	18
II.3.3.6 Procédure de tracé de l'impulsion unité.....	19
II.3.3.7 Procédure de tracé du signal carré	19
II.3.3.8 Procédure de tracé du bruit.....	19
III- EDITEUR DE SIGNAL.....	22
III-1 Introduction	22
III-2 Description.....	22
III-3 Procédures d'édition du signal.....	23
III-3-1 Procédure de visualisation du signal.....	23
III-3-2 Procédure de dilatation du signal (zoom).....	28
III-3-3 Spectre du signal	30
III-3-3-1 Echantillonnage d'un signal analogique.....	30
III-3-3-2 Fenêtres de pondération (Troncature).....	32
III-3-3-3 Représentation graphique du spectre.....	37
III-3-4 Procédure de visualisation de la primitive.....	43
III-3-5 Procédure de visualisation de la dérivée du signal.....	44
III-3-6 Procédure d'écoute du signal.....	45
III-3-7 Procédure Clear.....	45
III-3-8 Procédure Fin.....	45

III-3-9 Procédure d'analyse cepstrale.....	46
III-3-9-1. Estimation de la trajectoire des formants.....	46
III-3-9-2 Procédure de préaccentuation du signal.....	49
III-3-10 Procédure spectrogramme du signal.....	51
IV. MODE D'EMPLOI DE L'EDITEUR DE SIGNAL.....	52
IV.1 – Introduction.....	52
IV. 2 - Mode d'emploi du synthétiseur des signaux	52
IV.3 - Mode d'emploi de l'éditeur de signal	55
V. EXEMPLES D'APPLICATION.....	61
V.1.Comparaison des différentes fenêtres de pondération.....	61
V.2 Variation de la fréquence des signaux	63
V.3 Zoom de quelques signaux	66
V.4 Exemple de visualisation d'un signal de parole.....	67
CONCLUSION GENERALE.....	71

Introduction générale

INTRODUCTION GENERALE

Le traitement numérique du signal consiste en un ensemble de théories et de méthodes, relativement indépendantes du signal traité, permettant de créer, d'analyser, de modifier, de classer, et finalement de reconnaître les signaux. Il s'agit donc d'une science appliquée, puisque le signal numérique n'existe pour ainsi dire pas dans la nature. Il est une invention de l'homme, qui a pour but principal de permettre une manipulation aisée de signaux analogiques à l'aide de calculateurs numériques. Ses applications sont nombreuses dans des domaines aussi variés que les télécommunications, le traitement du son musical, le traitement de la parole, le radar, le sonar, l'étude des vibrations sismiques, le contrôle non destructif, l'ingénierie bio-médicale, l'imagerie, l'économie (avec l'étude des séries chronologiques), etc.

On peut estimer que cette science de l'ingénieur a réellement vu le jour dans les années 60, avec la découverte d'algorithmes rapides de calcul de la transformée de Fourier discrète par Cooley et Tukey. Les ouvrages de base ont été publiés entre la fin des années 70 et le début des années 90. Ce n'est que récemment, cependant, que les applications pratiques se sont multipliées, avec l'avènement des technologies numériques (RNIS-CD-GSM-DVD-MP3, etc.) et des processeurs spécialisés en traitement du signal (DSP : digital signal processor).

Afin de pouvoir à long terme synthétiser la parole et comprendre le mécanisme de perception de son signal, il nous a été proposé dans le cadre de ce projet de fin d'études, de développer un logiciel d'édition de signal et ce dans un but d'analyse de type numérique de la parole.

Puisque le traitement de ces signaux fait appel à des équipements adéquats tels que le microphone, un enregistreur FM, une carte d'acquisition et de restitution du signal (carte OROS), il nous a semblé dans un premier temps et à défaut de ce matériel généré des signaux sinusoïdaux, rectangulaires, triangulaires, impulsionnels, en dent de scie, carré et bruit. Ainsi ces signaux seront traités par l'éditeur de signal.

Le premier chapitre sera consacré à des généralités sur le MATLAB, le deuxième chapitre à la synthèse des formes d'ondes simples. Dans le troisième chapitre sera décrit le programme « Editeur de signal »,

Par contre dans le quatrième chapitre, nous allons proposer un mode d'emploi du logiciel accompagné d'illustrations et menus facilitant l'utilisation du programme. Enfin, dans le cinquième et dernier chapitre, nous allons présenter sous forme graphique, l'application du logiciel aux différents types de signaux.

En conclusion, nous allons faire une analyse des résultats obtenus, dégager l'intérêt d'un tel logiciel et faire des propositions pour la poursuite de cette étude.

Chapitre I

Généralités sur le matlab

Références [1]

I- GENERALITES SUR LE MATLAB

I.1- Introduction

MATLAB dont le nom provient de MATrix LABoratory est un système interactif et convivial de calcul numérique et de visualisation graphique basé sur le type de variable matricielle. Il possède un langage de programmation à la fois puissant et simple d'utilisation. Il permet d'exprimer les problèmes et solutions d'une façon aisée, contrairement aux autres langages de programmation.

MATLAB intègre des fonctions d'analyse numérique, de calcul matriciel, de traitement de signal, de visualisation graphique 2D et 3D, etc. Il peut être utilisé de façon interactive ou en mode programmation. En mode interactif, l'utilisateur a la possibilité de réaliser rapidement des calculs sophistiqués et d'en présenter les résultats sous formes numériques ou graphiques. En mode programmation, il est possible d'écrire des scripts (programmes) comme avec d'autres langages. L'utilisateur peut aussi créer ses propres fonctions pouvant être appelées de façon interactive ou par les scripts. Ces fonctions fournissent à MATLAB un atout inégalable : son extensibilité. Ainsi, l'environnement MATLAB peut être facilement étendu.

MATLAB s'impose dans les mondes universitaire et industriel comme un outil puissant de simulation et de visualisation de problèmes numériques. Dans le monde universitaire, MATLAB est utilisé pour l'enseignement de l'algèbre linéaire, le traitement du signal, l'automatique ainsi que les recherches scientifiques. Dans le domaine industriel, il est utilisé pour la résolution et la simulation de problèmes pratiques d'ingénierie et de prototypage.

A l'origine, MATLAB était écrit en FORTRAN, par CLEVE MOLER. La version actuelle est écrite en C par The MathWorks. Sa disponibilité est assurée sur plusieurs plates-formes : Sun, Bull, HP, IBM, compatibles PC, Macintosh et plusieurs machines parallèles.

MATLAB est conforté par une multitude de boîtes à outils (toolboxes) spécifiques à des domaines variés. Un autre atout de MATLAB, est sa portabilité ; la même portion de code peut être utilisée sur différentes plates-formes sans modification.

En complément de MATLAB, l'outil additionnel SIMULINK est proposé pour la modélisation et la simulation de systèmes dynamiques en utilisant une représentation de type schémas blocs.

En plus des caractéristiques exceptionnelles de MATLAB pour le calcul scientifique et technique, l'utilisateur peut créer pour ses applications, ses propres interfaces graphiques (GUI : Graphical User interface).

En effet, MATLAB prévoit un ensemble de commandes et fonctions dont l'utilisation est relativement aisée pour la création et la manipulation d'objets graphiques (fenêtres, menus, boutons de commandes, cases à cocher, etc.)

C'est cette partie que nous allons développer car elle permet non seulement de visionner les signaux, mais aussi et surtout de les traiter.

I.2- Organisation du logiciel Matlab

Le logiciel Matlab se compose d'un noyau et d'un ensemble de programmes auxiliaires habituellement regroupés dans des dossiers différents. Le dossier j:\Matlab\Bin renferme le noyau de Matlab dont le document principal, Matlab.exe, contient les primitives de calcul et les interfaces permettant l'accès à des documents de textes ou de données

1.3- Réalisation d'interfaces graphiques

L'élément de base d'une interface graphique est la fenêtre. Elle permet de grouper des outils graphiques dans un même cadre dans un but de clarté et de manipulation facile. A une fenêtre, sont associées des propriétés modifiables telles que la taille, le nom, la position, le titre, la couleur, etc.

1. Les contrôles

Les contrôles sont des objets graphiques qui réagissent et provoquent une action lorsqu'ils sont manipulés par la souris ou le clavier.

Sous une interface graphique, la communication avec l'utilisateur est généralement établie à l'aide des « contrôles » tels que les groupes d'option, cases à cocher, boutons de commande, textes statiques et textes modifiables. Ainsi, les programmes offrent une meilleure convivialité, car il suffit d'un simple clic avec le pointeur de la souris pour définir, par exemple, de nouvelles valeurs.

1.1 Les types de contrôles

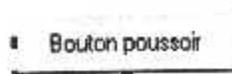
Dans MATLAB, on trouve différents types de contrôles. Leur allure dépend de la plateforme sur laquelle vous travaillez (Unix, MS Windows, Macintosh, etc.) Nous allons travailler dans l'environnement MS Windows.

Liste des contrôles présents dans MATLAB :

Contrôles	Description
Bouton-poussoir (ou bouton de commande)	Réalise une action
Bouton radio (ou bouton d'option)	Un groupe de boutons radio permet une sélection exclusive (un seul bouton activé à la fois. Lorsqu'on en a un seul, il permet la permutation entre deux états).
Case à cocher	Un groupe de cases à cocher permet une sélection non exclusive (plusieurs cases peuvent être cochées à la fois). Lorsqu'on en a une seule, elle permet la permutation entre deux états.
Texte statique	Affiche une ligne de texte d'information.
Texte éditable	Zone de texte éditable, permettant à l'utilisateur d'entrer une chaîne de caractères.
Menu	Permet de choisir un élément dans une liste.
Cadre (frame)	Affiche une bordure rectangulaire autour d'un ou plusieurs contrôles pour former un groupe logique.
Barre de défilement (slider)	Permet à l'utilisateur de choisir une valeur dans un intervalle donné (en glissant un curseur).

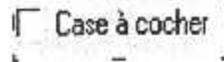
Bouton poussoir

Le bouton-poussoir est aussi connu sous l'appellation de bouton de commande. Il contient généralement le texte « OK » ou « annuler ». Une action précise s'exécute dès que l'utilisateur effectue un clic sur ce bouton.



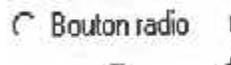
Case à cocher

Les cases à cocher permettent à l'utilisateur de sélectionner une ou plusieurs options parmi plusieurs alternatives. Ces cases à cocher se comportent comme des interrupteurs, indiquant s'ils sont actionnés ou non. Par convention, ces cases à cocher sont indépendantes les unes des autres. On peut en cocher n'importe quel nombre, l'état de l'une d'entre elles ne doit pas influencer sur celui des autres.



Bouton radio

Les boutons radio ou boutons d'options offrent à l'utilisateur la possibilité de choisir entre plusieurs alternatives exclusives. Comme les cases à cocher, les boutons radio se comportent comme des interrupteurs indiquant leur état d'activation. La seule différence est qu'ils doivent être programmés comme étant mutuellement exclusifs. On ne peut en activer qu'un seul parmi un ensemble de boutons proposés.



Les barres de défilement (Slider)

Les barres de défilement permettent à l'utilisateur de choisir une valeur appartenant à un intervalle donné. Ce sont des objets graphiques disposant d'un indicateur mobile qui représente la position relative de la valeur courante dans l'intervalle autorisé. La propriété « valeur de la barre de défilement » peut être modifiée soit par déplacement de l'indicateur mobile, soit par les flèches disposées à chaque extrémité de l'objet graphique.

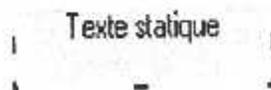


Les zones de textes

On distingue deux types de zones de texte : une zone de texte statique et une zone de texte éditable.

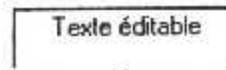
Zone de texte statique

Une désignation, que l'utilisateur ne peut modifier, est utilisée pour afficher un texte. Il s'agit normalement d'un titre ou d'une brève explication. La valeur de cette zone peut être modifiée par programmation.



Zone de texte éditable (ou champ d'édition)

C'est une zone de texte que l'utilisateur peut modifier. Elle représente un véritable éditeur de texte. Sa valeur est une chaîne de caractères (d'une ou plusieurs lignes) qui sera utilisée par l'application.



Les cadres ou frames

Ce sont des rectangles que l'on dessine autour de différents contrôles pour les répartir en groupes logiques dans un but d'esthétique ou de fonctionnalité.

1.2. Création de contrôles

Pour créer des contrôles, on utilisera la fonction `uicontrol` en respectant la syntaxe suivante :
`h = uicontrol(hfig, 'Nompropriété', 'valeurpropriété', ...)`

`h` : « handle » sur le contrôle créé, utile pour la manipulation et la modification de ce contrôle

`hfig` : correspond au « handle » de la fenêtre sur laquelle on veut créer contrôle (`gcf` : correspond à la figure courante)

`NomPropriété` : correspond au nom de la propriété (style, position, string, callback, etc), peut être indiqué en minuscule ou en majuscule,

`ValeurPropriété` : peut être un nombre, un tableau, une chaîne de caractères, le type dépend de la propriété.

L'action que MATLAB réalise lorsque l'utilisateur active le contrôle est indiquée dans la propriété « callback ».

1.3 Propriétés des contrôles

Tous les objets graphiques de MATLAB ont des propriétés qui contrôlent la façon dont ils sont affichés à l'écran. Ces propriétés contiennent des informations générales (le type de l'objet, l'objet parent, les objets enfants, etc.) et des informations spécifiques au type d'objet considéré (les échelles des axes, les données utilisées pour le tracé des graphes, etc). A chaque création d'objet, un « handle » lui est associé, ainsi que des valeurs par défaut aux différentes propriétés générales et particulières de l'objet.

On peut obtenir des informations sur les valeurs courantes des propriétés d'un objet par la commande `get`. Il est aussi possible de modifier les valeurs de certaines propriétés à l'aide de la commande `set` (certaines propriétés ne sont pas modifiables par l'utilisateur). Un certain ensemble de propriétés est disponible pour tous les objets.

Pour consulter les valeurs des propriétés d'un objet ayant pour « handle » `h`, on utilisera `get(h)`. Pour obtenir la liste des propriétés modifiables d'un objet de « handle » `h`, on utilisera la commande `set(h)`. Cette même commande permettra de fixer les valeurs des propriétés d'un objet. Sa syntaxe est la suivante :

`Set(handle, 'propriétéP1', valeurP1, 'propriétéP2', valeurP2, ...)`

Appliquée à un vecteur de « handles », la commande `set` aura pour résultat la modification des propriétés spécifiées pour tous les objets dont on a indiqué les pointeurs.

Pour récupérer la valeur d'une propriété, on utilisera la fonction `get`.

`VP=get(handle, 'propriété')`

Pour supprimer un objet graphique, on utilisera la commande `delete` en spécifiant le « handle » de l'objet.

`Delete(handle)`

Style

Cette commande identifie le type de contrôle utilisé. Les valeurs possibles sont données dans le tableau qui suit :

Contrôle	Valeur
Bouton poussoir	pushbutton
Bouton radio	radiobutton
Case à cocher	checkbox
Texte statique	text
Texte éditable	edit
Menu popup	popupmenu
Cadre	frame
barre de défilement	slider

String

Représente :

- Le texte qui apparaît sur le contrôle dans le cas des boutons-poussoirs,
- Le texte à côté des boutons radio et des cases à cocher,
- La valeur par défaut dans le cas d'un texte éditable,
- La liste des choix pour les menus popup,
- Les textes apparaissant sur le contrôle pour le texte statique.

Cette propriété n'est pas utilisée pour les cadres et les barres de défilement.

Value

Représente la valeur courante du contrôle.

Position

La propriété `position` permet de spécifier la position d'un contrôle ainsi que sa taille. La valeur de cette propriété est un tableau à quatre éléments (gauche, bas, hauteur, largeur).

- **Gauche** : distance du coin inférieur gauche de la fenêtre au côté gauche du contrôle,
- **Bas** : distance du côté bas de la fenêtre au côté bas du contrôle,
- **Hauteur** : hauteur du contrôle,
- **Largeur** : largeur du contrôle.

Callback

Les actions réalisées par un contrôle, lorsqu'il est activé, sont données par la propriété callback. La valeur de cette propriété est une chaîne de caractères pouvant être une commande MATLAB, le nom d'une variable ou un fichier M.

Cette chaîne est évaluée par la formation eval lorsque le contrôle est sélectionné.

Dans le cas d'actions complexes, il est conseillé d'utiliser une fonction pour la propriété callback.

Visible

Cette propriété détermine si l'objet graphique apparaît sur la figure. Les valeurs possibles sont : on et off. L'intérêt de cette propriété est d'afficher seulement les contrôles nécessaires à un moment donné de l'exécution de l'application.

Min et Max

Valeurs minimale et maximale de l'intervalle de défilement pour un contrôle de type barre de défilement. Ces valeurs permettent aussi de spécifier si une zone de texte éditable contient une ou plusieurs lignes.

Liste des propriétés courantes et leurs significations

Propriété	Description
backgroundColor	couleur du fond de contrôle
callback	action à exécuter lorsque le contrôle est activé
foregroundColor	spécifie la couleur du texte ou du tracé pour le contrôle
horizontalalignment	position du texte dans un contrôle
max	valeur max que peut prendre le contrôle
min	valeur min que peut prendre le contrôle
position	position et taille du contrôle
string	définit le texte qui apparaît sur le contrôle
style	définit le type du contrôle (boutons poussoirs, texte, etc.)
type	type de l'objet, « uicontrol » pour les contrôles
units	unité de mesure du contrôle (pour la position et la taille)
user data	matrice associée à l'objet
value	valeur courante du contrôle
visible	indique si l'objet apparaît ou non à l'écran
parent	représente le « handle » sur l'objet parent
interruptible	indique si le callback peut être interrompu ou non

2. Les menus

La fenêtre principale de la majorité des applications graphiques (windows, Mac, Window, etc.) possède une barre de menus. La sélection de l'un de ces éléments fait apparaître un menu déroulant présentant une liste d'options. Chacune de ces options peut conduire à un autre sous menu, ou être un élément terminal. Les menus présentent ainsi une structure arborescente. Leur allure dépend de la plate-forme utilisée.

Une commande d'un menu peut être sélectionnée à l'aide du clavier ou de la souris. Plusieurs menus peuvent être définis simultanément, chacun d'eux doit posséder un nom qui lui est propre.

2.1. Création de menus

Pour créer des menus on utilisera la fonction `uimenu` en respectant la syntaxe suivante :

`Hmenu = uimenu (hfig, « NomPropriété », « ValeurPropriété », ...)`

`Hmenu` : « handle » sur le menu créé, utile pour la manipulation et la modification des propriétés,

`Hfig` : correspond au « handle » de la fenêtre (figure) sur laquelle on veut créer le menu.

La création des éléments d'un menu se fera avec la même commande en spécifiant le « handle » du menu père.

`mnuitem1 = uimenu (hmenu, « NomPropriété », « ValeurPropriété », ...)`

`hmenu` : « handle » sur le menu père,

`mnuitem1` : « handle » sur l'élément de menu créé

2.2. Les propriétés des menus

Nous présentons ci-dessous un tableau récapitulatif des propriétés essentielles d'un menu.

Propriété	Description
Accelerator	précise l'équivalent clavier pour l'accès à un élément d'un menu
backgroundColor	couleur de fond du menu
foregroundColor	couleur du texte du menu
callback	script, fonction ou commande à exécuter lors de l'activation d'un élément du menu
checked	indique si un élément de menu est sélectionné par une marque
children	liste des « handle » sur les éléments d'un menu donné
label	définit le titre du menu, ou le nom de l'option du menu
parent	« handle » sur le menu ou l'objet parent
position	position du menu dans la barre des menus ou celle d'une option dans un menu.
separator	ligne horizontale de séparation entre les éléments d'un menu.
visible	indique si le menu et ses objets enfants sont visibles sur la fenêtre graphique.

Chapitre II

Synthèse des signaux simples

Références [2], [3]

II. Synthèse Des Signaux Simples

II.1 Introduction

Dans l'état actuel, nous ne disposons pas dans notre laboratoire d'équipements adéquats pour l'acquisition et la restitution d'un signal vocal. Nous sommes orientés, à défaut d'un signal numérisé de parole, à la synthèse des fonctions simples. Il s'agit de construire dans le domaine temporel le signal voulu à l'aide de fonctions temporelles ayant des caractéristiques spectrales intéressantes.

Nous pouvons constituer un signal synthétique grâce à une formule mathématique simple.

II.2 Description

Notre « générateur de signaux » est entièrement élaboré en MATLAB, exécutable sur IBM-PC et PC compatibles. Ce programme permet de générer des signaux rectangulaires, triangulaires, exponentiels, dents de scie, sinusoïdaux, impulsionnels, carrés, bruits de fréquence de durée et d'amplitude déterminées.

Par analogie au signal de parole, la bande de fréquence audible varie de 20Hz à 20KHz. Le signal est ensuite échantillonné à une fréquence déterminée pour chaque signal. Le signal généré est représenté par un bouton dont un simple clic permet à l'utilisateur d'introduire les différents paramètres pour son analyse.

II.3 Procédures et fonctions :

Le programme principal dénommé (signal.m) (générateur de signaux) comporte trois procédures : affichage , choix , calcul et tracé.

L'organigramme de cette partie principale est représenté à la figure II – 1

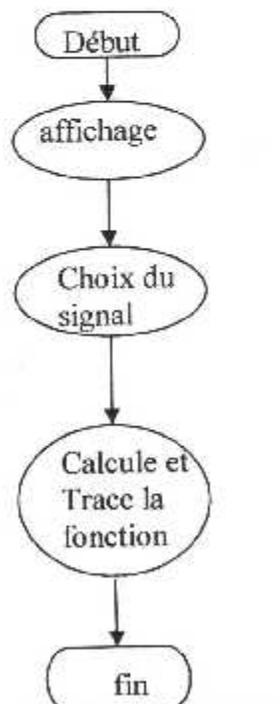


Fig II.1 : Organigramme de la procédure « signal.m »

II.3.1 Procédure d'entrer en matière « affichage »

La procédure "affichage" permet d'afficher l'interface. En utilisant les instructions « Windows » de Matlab, cette procédure affiche l'interface comprenant les boutons des différents signaux et les fenêtres. La figure II-2 montre l'organigramme de la procédure « affichage »

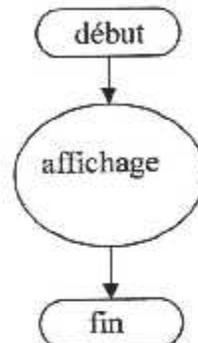


Fig II.2 : Organigramme de la procédure « affichage »

II.3.2 Procédure choix du type de signal « choix »

Cette procédure commence par l'affichage de l'interface. Cette dernière présente les différents signaux à étudier. L'utilisateur porte son choix sur un type de signal. Un clic sur son bouton permet à l'instruction « case » de faire un test, chaque signal étant affecté d'un numéro de test. L'instruction « callback » permet alors d'appeler le signal choisi.

La figure II-3 montre l'organigramme de la procédure choix du type de signal.

II.3.3 Procédure du tracé des fonctions

Cette procédure comprend dix routines associées aux dix types de signaux au moyen de l'instruction « case » et selon le type de signal choisi auparavant. Cette procédure englobe l'introduction des paramètres. Il y a au préalable les paramètres par défaut avec lesquels le signal choisi est tracé. Nous pouvons personnaliser ces paramètres par l'introduction d'autres paramètres et un clic sur le bouton « tracer » permet de les initialiser afin de tracer le signal.

Les neuf types de signaux étudiés au sein de ce programme sont successivement :

1. Impulsion rectangulaire
2. Impulsion triangulaire
3. Impulsion exponentielle
4. Signal en dents de scie
5. somme de sinusoides
6. Impulsion unité
7. Signal carré
8. Le bruit
9. Signal *.Mat
10. Autre signal

L'organigramme de la procédure « tracé » est donné par la figure II-4

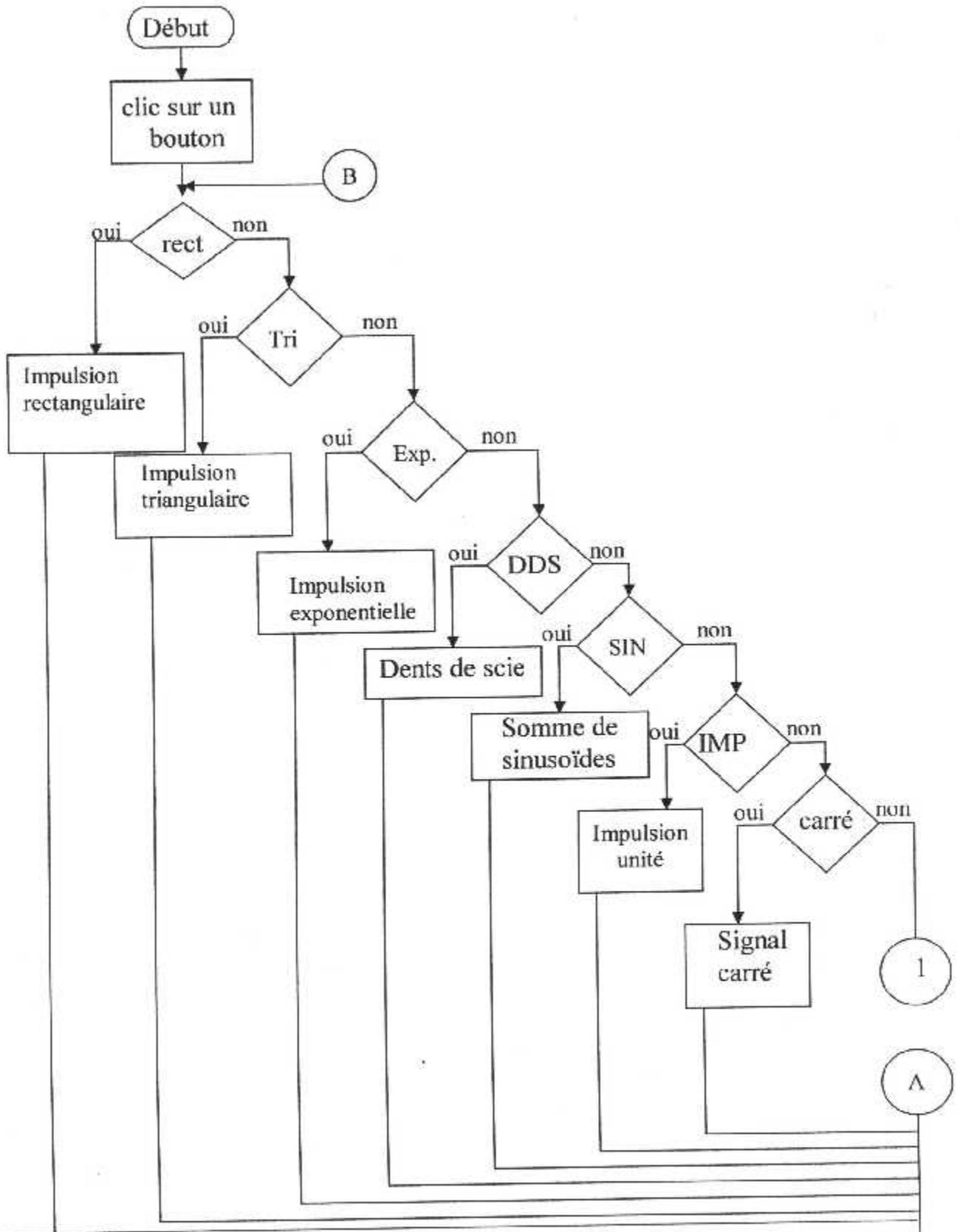
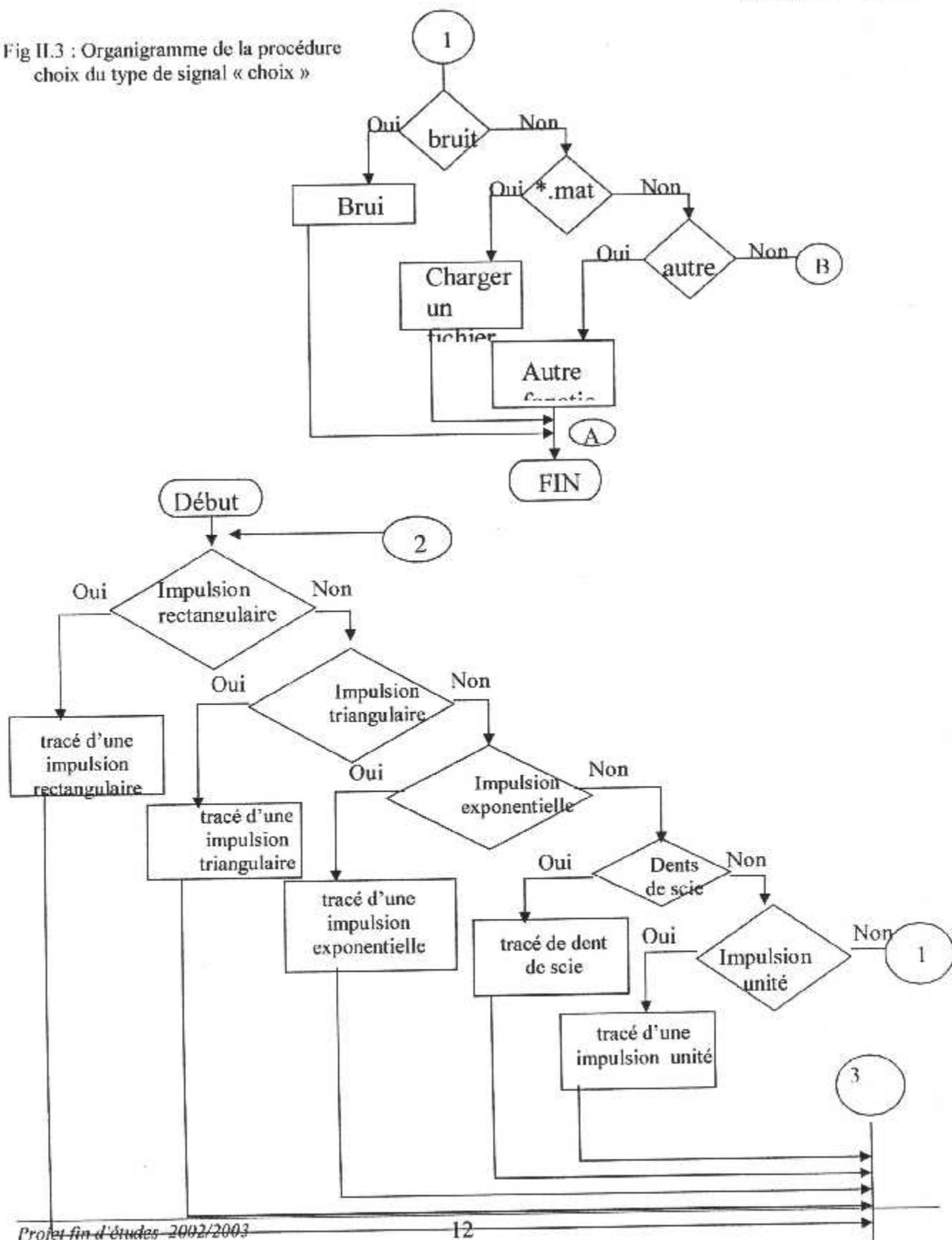


Fig II.3 : Organigramme de la procédure choix du type de signal « choix »



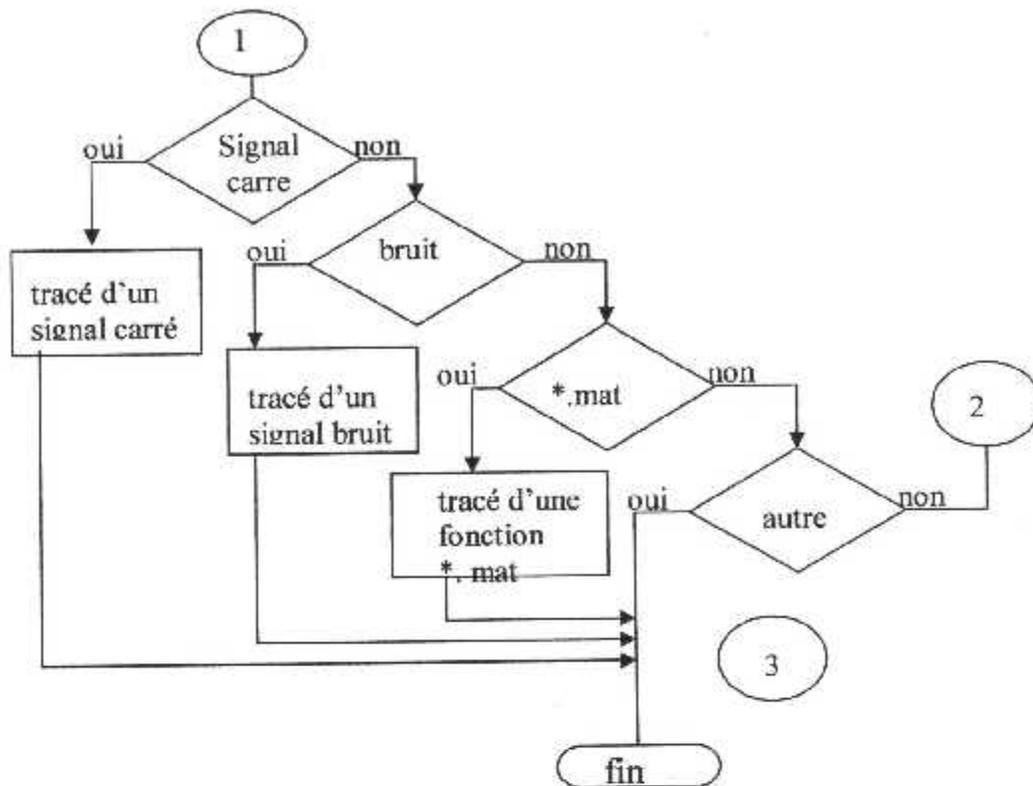


Fig II.4 : Organigramme de la procédure « tracé »

II.3.3.1 Procédure de tracé de l'impulsion rectangulaire

On calcule dans cette procédure l'amplitude des échantillons pendant une période du signal rectangulaire. On utilise pour cela la fonction :

$$Y = A * \text{RECTPULS} [t , \text{long}]$$

Où A représente l'amplitude, t le temps et long le nombre de périodes du signal.

La figure II-5 montre l'organigramme de la procédure de tracé de l'impulsion rectangulaire.

II.3.3.2 Procédure de tracé de l'impulsion triangulaire

Le calcul de l'amplitude des échantillons du signal triangulaire est donné par l'expression suivante :

$$Y = A * \text{TRIPULS}[t, \text{long}]$$

L'organigramme de cette procédure est donné par la figure II-6

II.3.3.3 Procédure de tracé de l'impulsion exponentielle

Cette procédure permet de calculer l'amplitude de chaque échantillon du signal exponentiel. On utilise la fonction : $Y = A * [0.*[t \leq \text{lim}] + \exp[-t].*[t > \text{lim}]]$

L'organigramme de la procédure est donné par la figure II-7

II.3.3.4 Procédure de tracé de la fonction DDS (Dents De Scie)

Cette procédure permet de calculer l'amplitude de chaque échantillon du signal Dents De Scie. On utilise la fonction : $Y=A*\text{sawtooth}[w*t]$

L'organigramme de la procédure est donné par la figure II-8

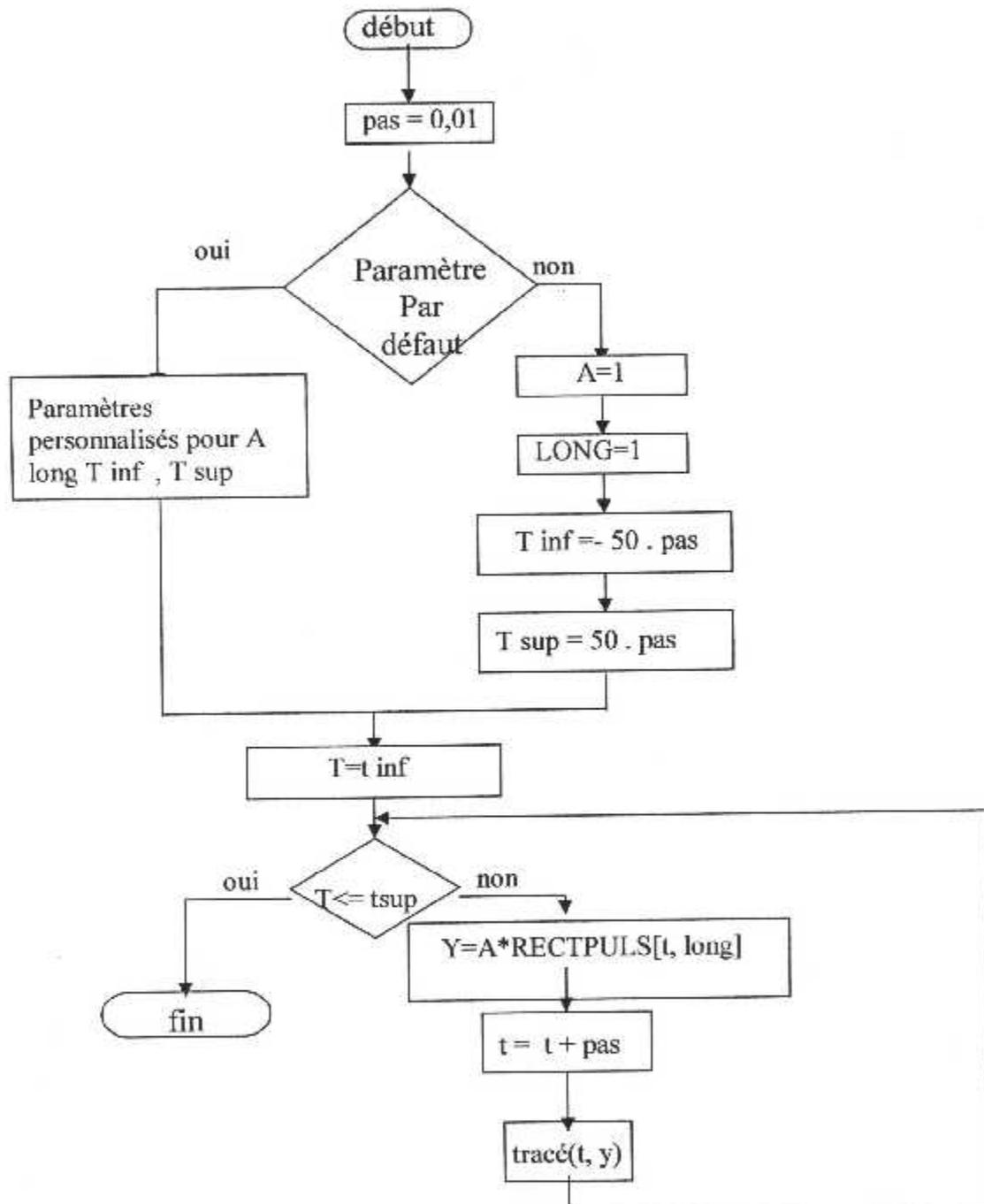


FIG II.5 : Organigramme de la procédure de tracé de l'impulsion rectangulaire

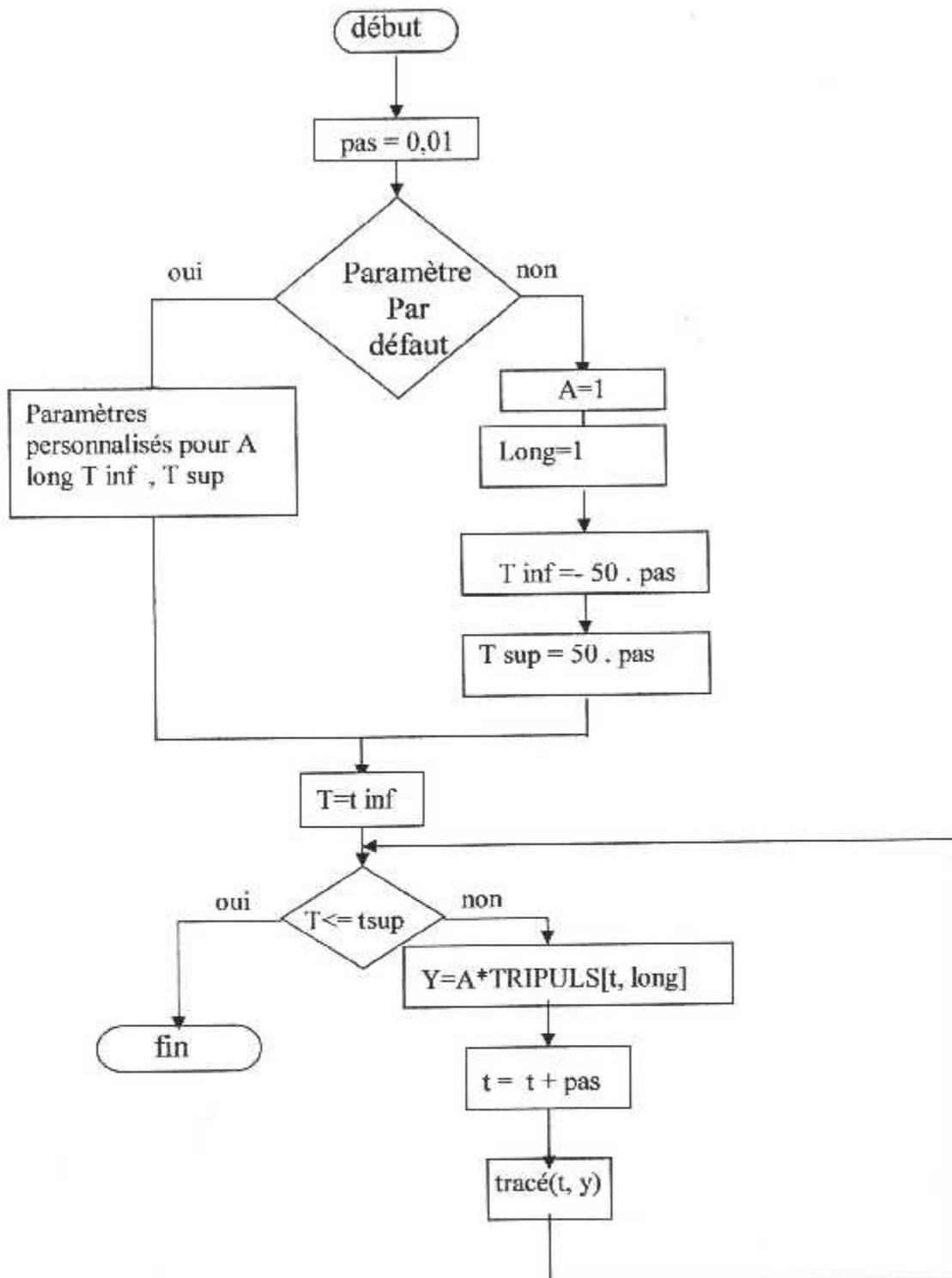


Fig II-6 . Organigramme de la procédure de tracé de l'impulsion triangulaire

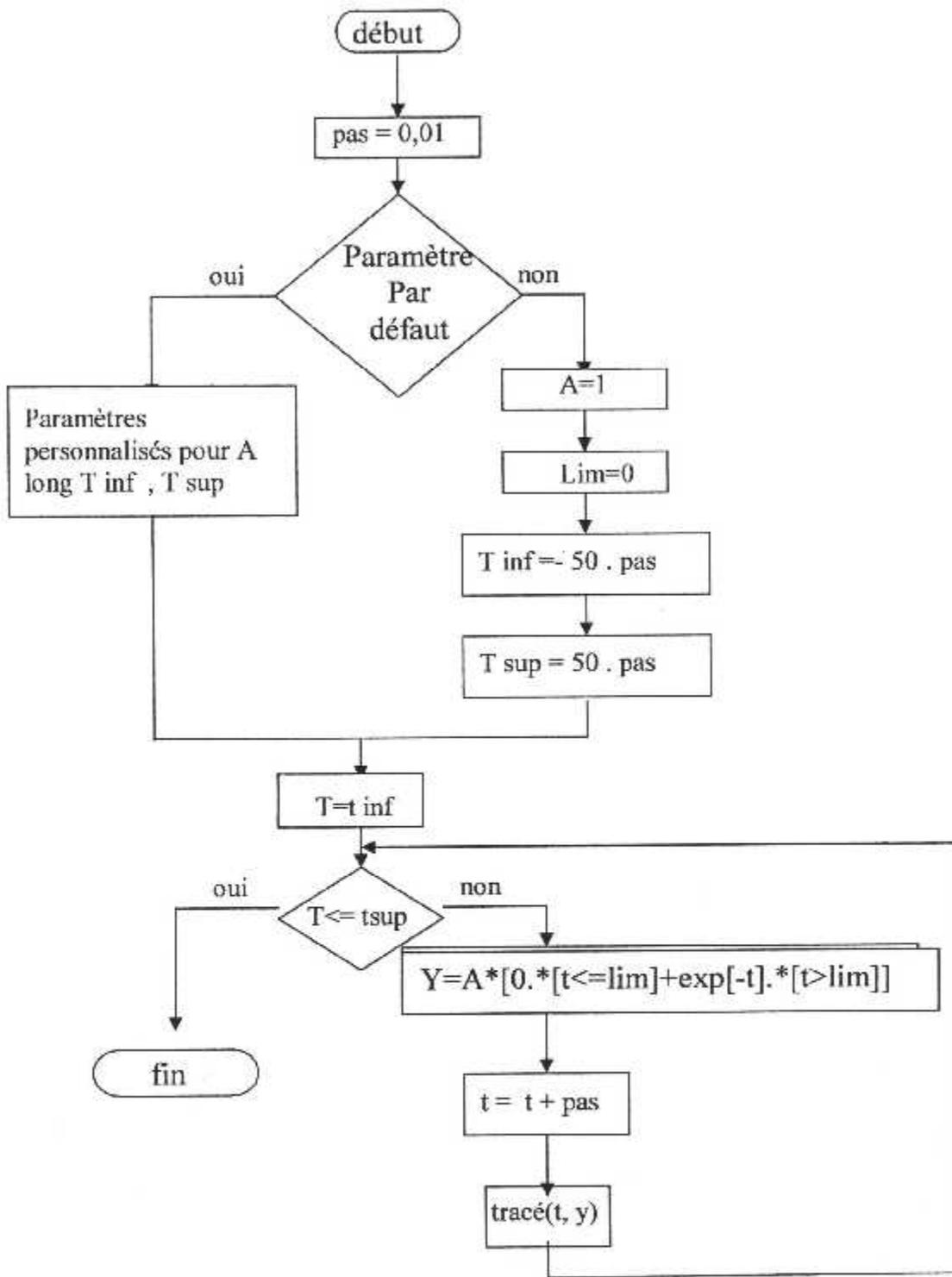


Fig II-7 Organigramme de la procédure de tracé de l'impulsion exponentielle

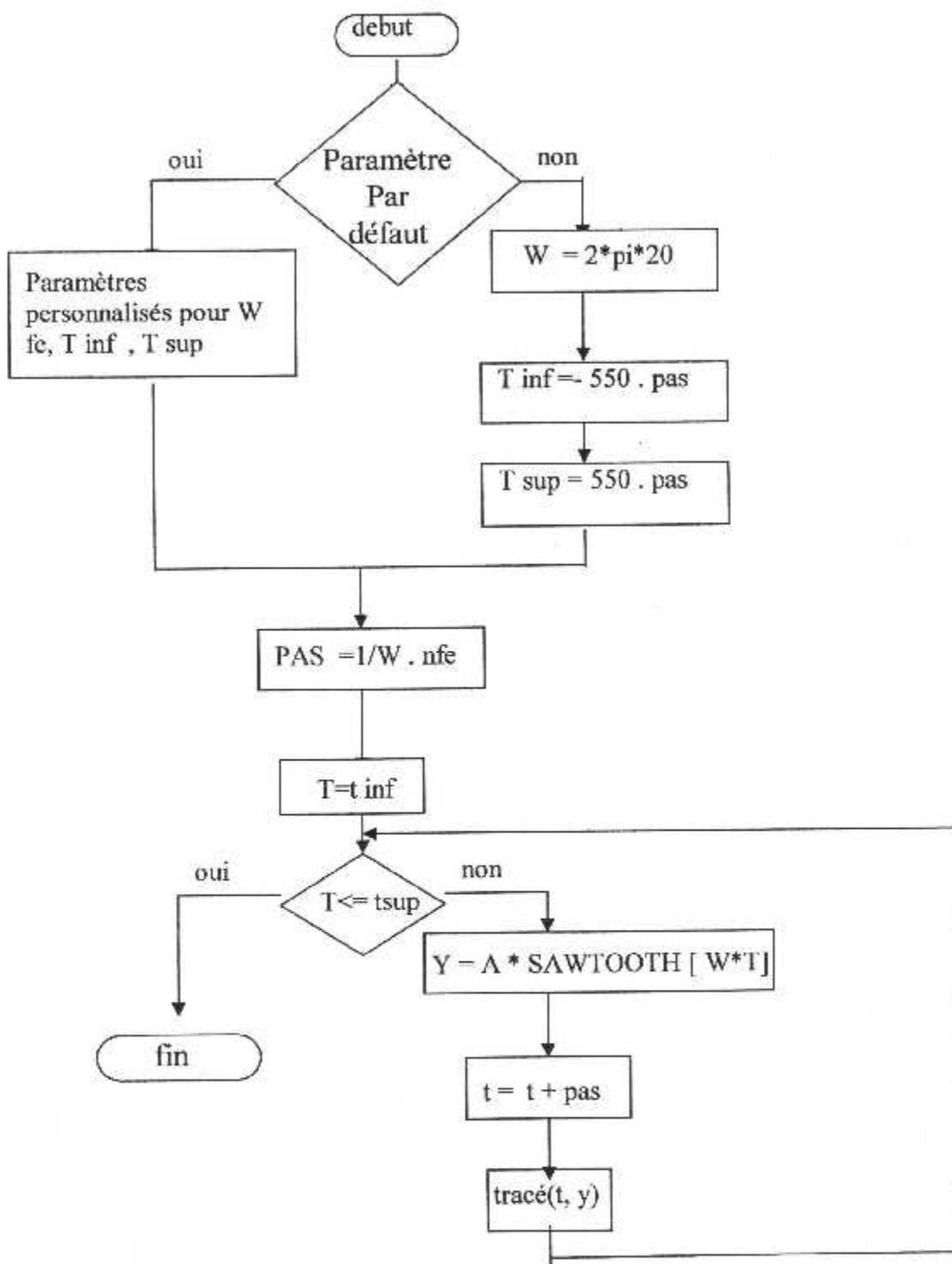


Fig II-8 Organigramme de la procédure de tracé du signal DDS

II-3-3-5 Procédure de tracé de la fonction somme de sinusoides

Cette procédure permet de calculer l'amplitude de chaque échantillon de la fonction somme de sinusoides. On utilise la fonction :

$$Y = A_1 \cos(w_1 t + \phi_{i1}) + A_2 \sin(w_2 t + \phi_{i2})$$

L'organigramme de la procédure est donné par la figure II-9

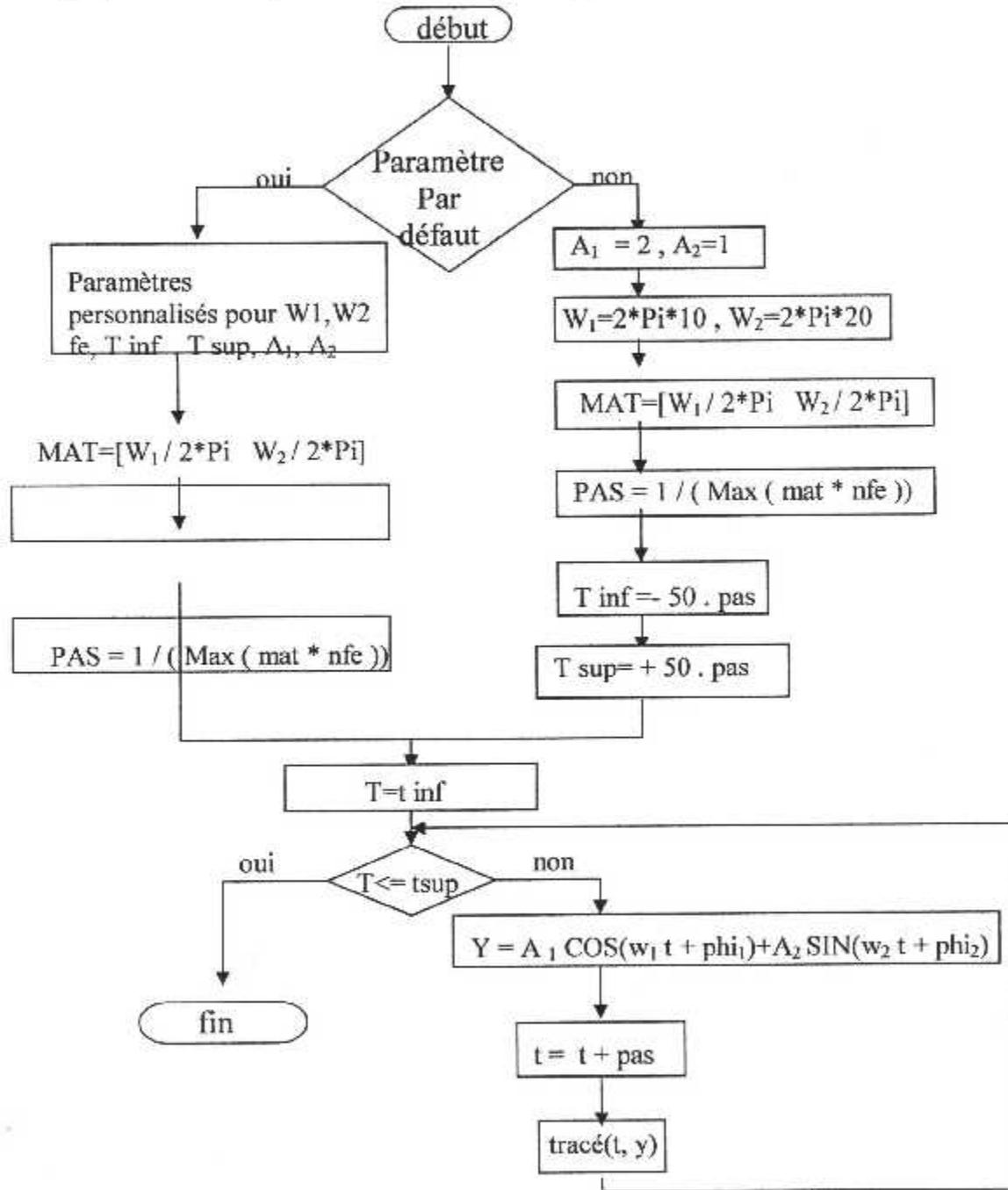


Fig II-9 Organigramme de la procédure tracé de somme de sinusoides

II-3-3-6 Procédure de tracé de l'impulsion unité

On calcule dans cette procédure l'amplitude des échantillons pendant une période du signal de l'impulsion unité. On utilise pour cela la fonction : $Y = A * \text{écheleon} [t + \text{retard}]$. C'est une fonction propre du MATLAB. L'organigramme de cette procédure est représenté par la figure II-10

II.3.3.7 Procédure de tracé du signal carré

On calcule dans cette procédure l'amplitude des échantillons pendant une période du signal carré. On utilise pour cela la fonction : $Y = A * \text{square} [w.t , n]$. C'est une fonction propre du MATLAB.

La figure II-11 représente l'organigramme de cette procédure.

II.3.3.8 Procédure de tracé du bruit

le calcul de l'amplitude des échantillons du bruit est donné par l'expression suivante :

$Y = A * \text{RANDN} (\text{size} (t))$. C'est une fonction propre du MATLAB.

L'organigramme de cette procédure est donné par la figure II-12

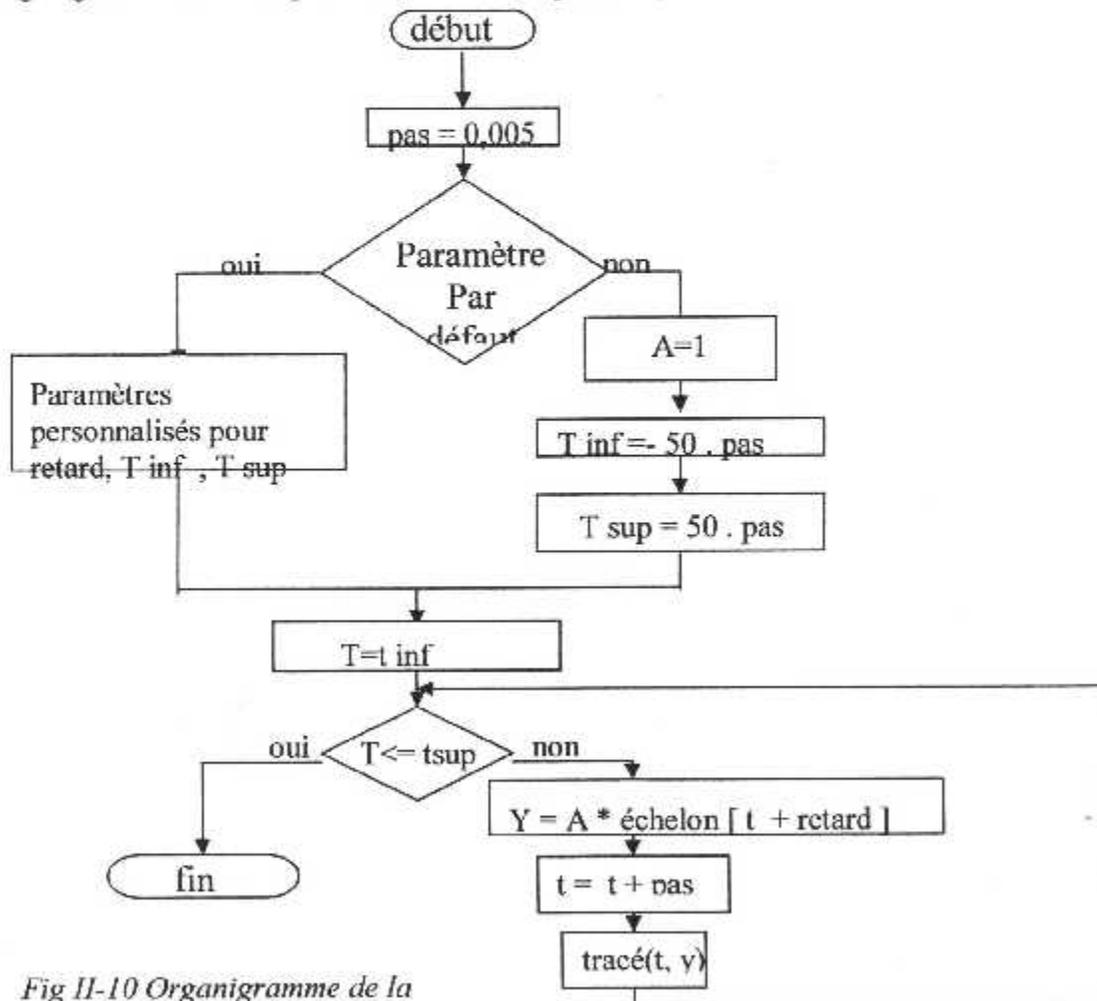


Fig II-10 Organigramme de la procédure tracé de l'impulsion

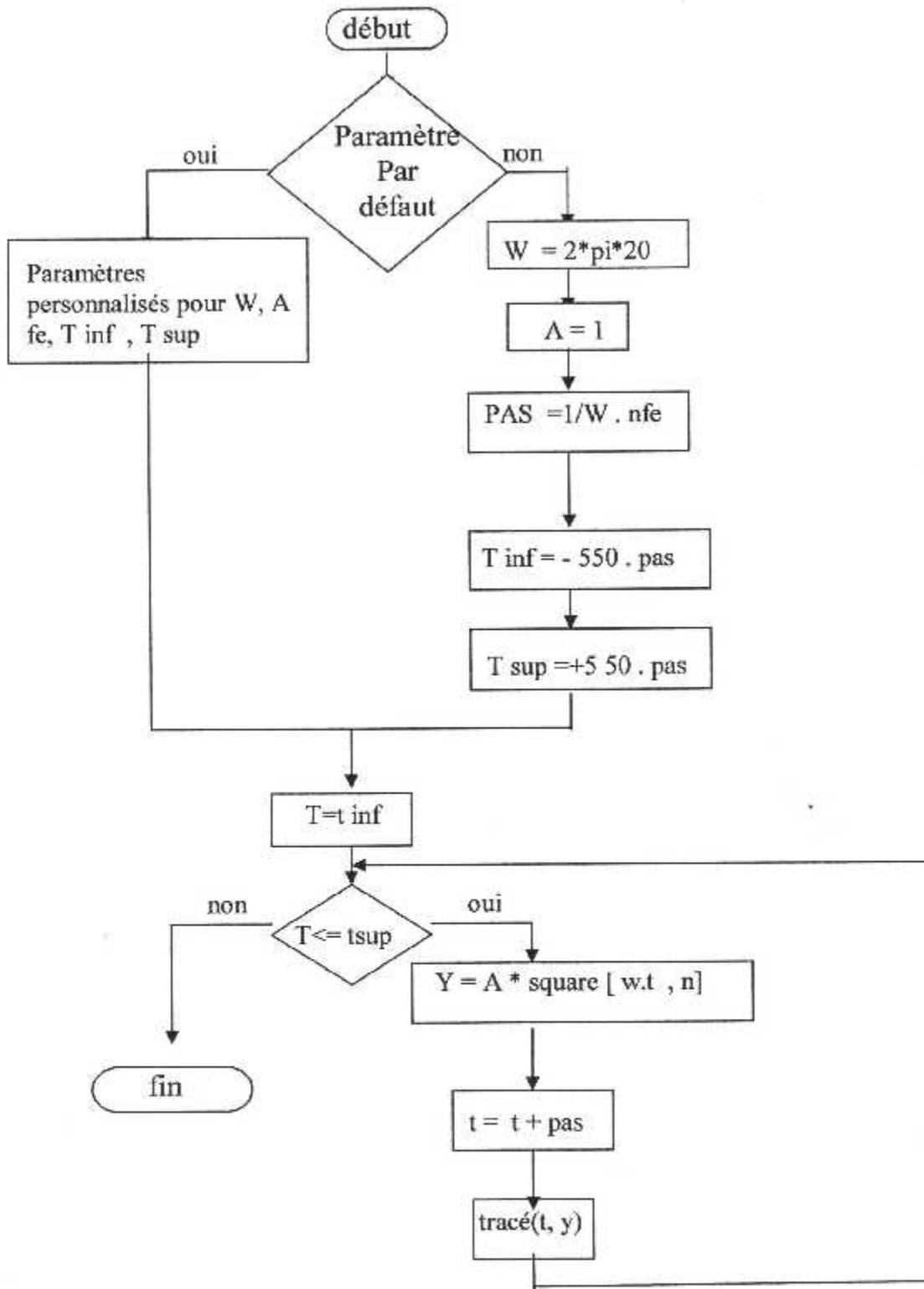


Fig II-11 Organigramme de la procédure de tracé du signal carré

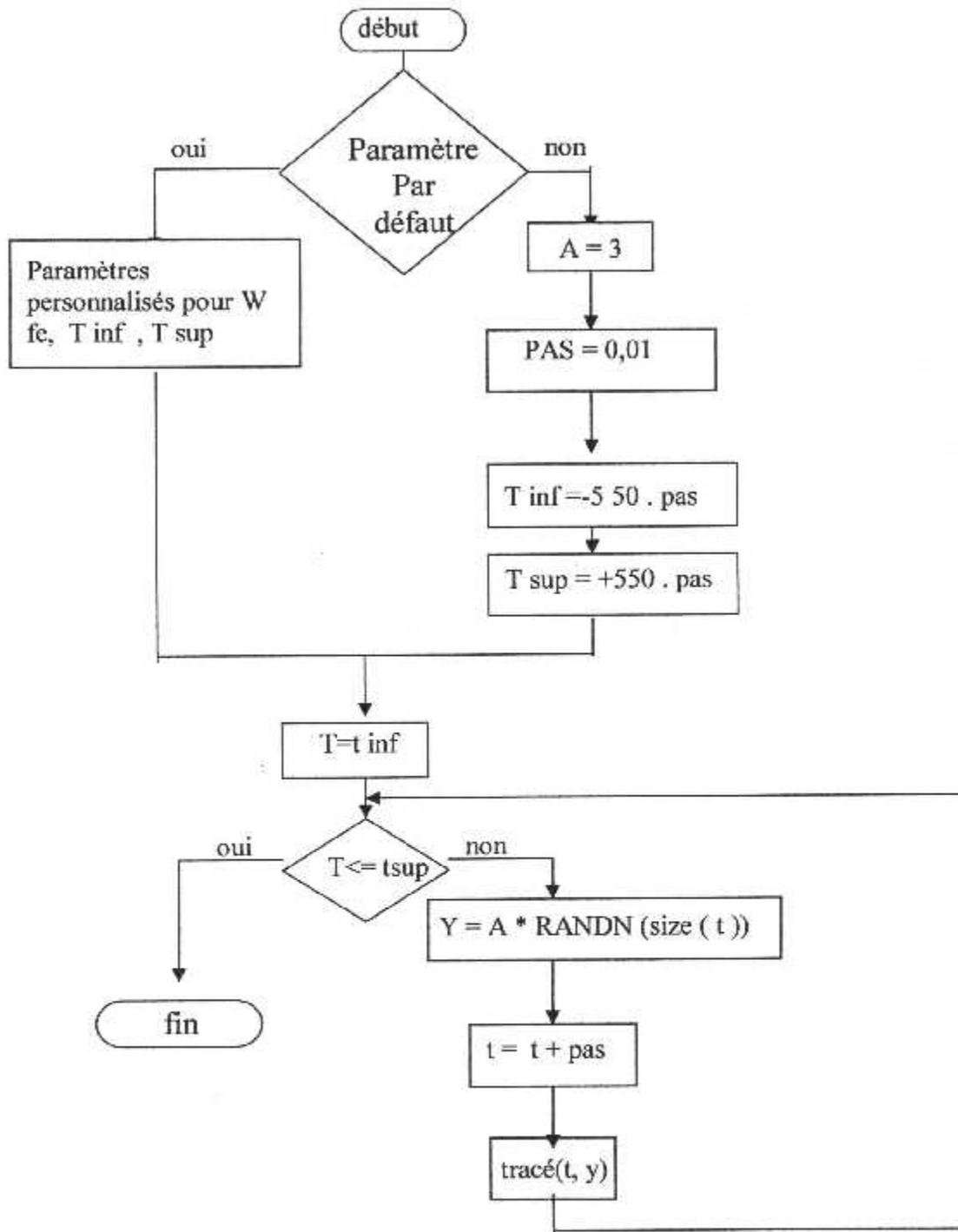


Fig II-12 Organigramme de la procédure tracé du bruit

Chapitre III

Éditeur de signal

Références

[2], [4], [5], [6], [7], [8]

III- EDITEUR DE SIGNAL

III-1 Introduction

Il s'agit d'un logiciel essentiellement graphique permettant d'éditer un signal de parole. Il offre la possibilité à tous ceux qui sont intéressés par la parole de mieux connaître, visualiser, comparer, analyser le matériel acoustique sur lequel ils travaillent préalablement indispensable à toute étude sur la communication Homme machine.

A partir du signal de parole numérisé, la première opération à faire est de passer à un premier traitement de la parole :

L'éditeur de signal a pour but de visualiser le signal de parole pour en faire des traitements. Le signal peut être zoomé, écouté, découpé.

La figure III-1 montre l'enregistrement d'un son *.wav.

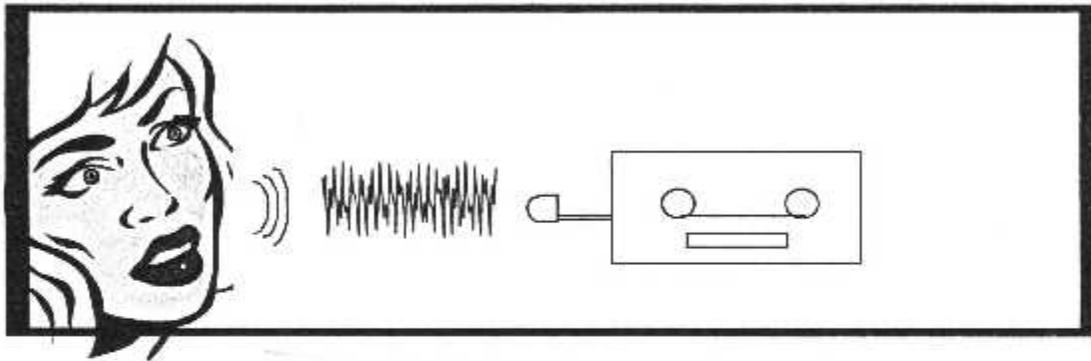


Fig III.1 enregistrement d'un son *.wav

Donc pour pouvoir exploiter au mieux ce logiciel, il faut disposer d'un signal de parole et le numériser. C'est pourquoi, nous y avons introduit une liste de voyelles orales.

La nécessité de vérifier la validité de notre éditeur de signal nous a conduit à établir un synthétiseur de signaux simples.

Sa fonction est d'abord de générer différents types de signaux ensuite de les stocker dans un fichier de nom *.wav.

III-2 Description

Les fonctions principales permettant une analyse fondamentale du signal de parole numérisé sont les suivantes :

- 1- Visualisation du signal de nom de fichier « *.wav ». Chaque signal est de fréquence et de durée déterminées.
- 2- Visualisation « à la loupe » de toute portion du signal déterminée par le zoom.
- 3- Visualisation des spectres d'amplitude et de phase par la FFT autour d'un échantillon précis. Possibilité de choix entre huit fenêtres de pondération et trois types de tracé.
- 4- Visualisation de la primitive du signal
- 5- Visualisation de la dérivée du signal.
- 6- Ecoute du signal entre n'importe quelle limite déterminée par le zoom.
- 7- Calcul du cepstre pour la détection du pitch (fondamental) et de la fréquence des formants. Précision du cepstre.
- 8- Visualisation du sonagramme (spectrogramme). Sa préaccentuation.

Dans les paragraphes qui suivent, nous allons montrer progressivement les différentes étapes suivies pour aboutir au logiciel d'édition du signal.

III-3 Procédures d'édition du signal

Disposons d'un fichier de signal *.wav, l'éditeur appelle le fichier désiré pour y lire les données propres au signal qui lui sont associées.

Dans la suite du programme, les données seront traitées, analysées de manière à aboutir à la relation de la fonction désirée.

Les fichiers « *.wav » se situent dans la liste des signaux de l'interface « parole ». Dans le cas où l'utilisateur veut traiter un signal stocké dans un fichier de chemin différent, par exemple « A:\ », il n'aura qu'à rentrer dans « Browse » pour sélectionner le fichier se trouvant sur la disquette.

III-3-1 Procédure de visualisation du signal

Cette procédure permet de visualiser un signal temporel dans une fenêtre graphique. Celle-ci ainsi ouverte, se situe en haut à droite de l'interface.

La procédure se déroule en deux étapes :

1^{ère} Etape : Visualisation de l'interface

Cette procédure permet l'apparition de l'interface.

Soit « ccc » les paramètres de l'écran et « position » la position de l'interface sur l'écran.

Position = position * c où c est un coefficient pris selon la résolution de l'écran.

2^{ème} Etape : création des axes et établissement de l'échelle.

Cette procédure permet de créer les axes et d'établir l'échelle pour le tracé du signal. La figure III-2 nous montre un exemple de création des axes. Dans cette figure, nous choisissons les axes selon la partie dont nous avons besoin.

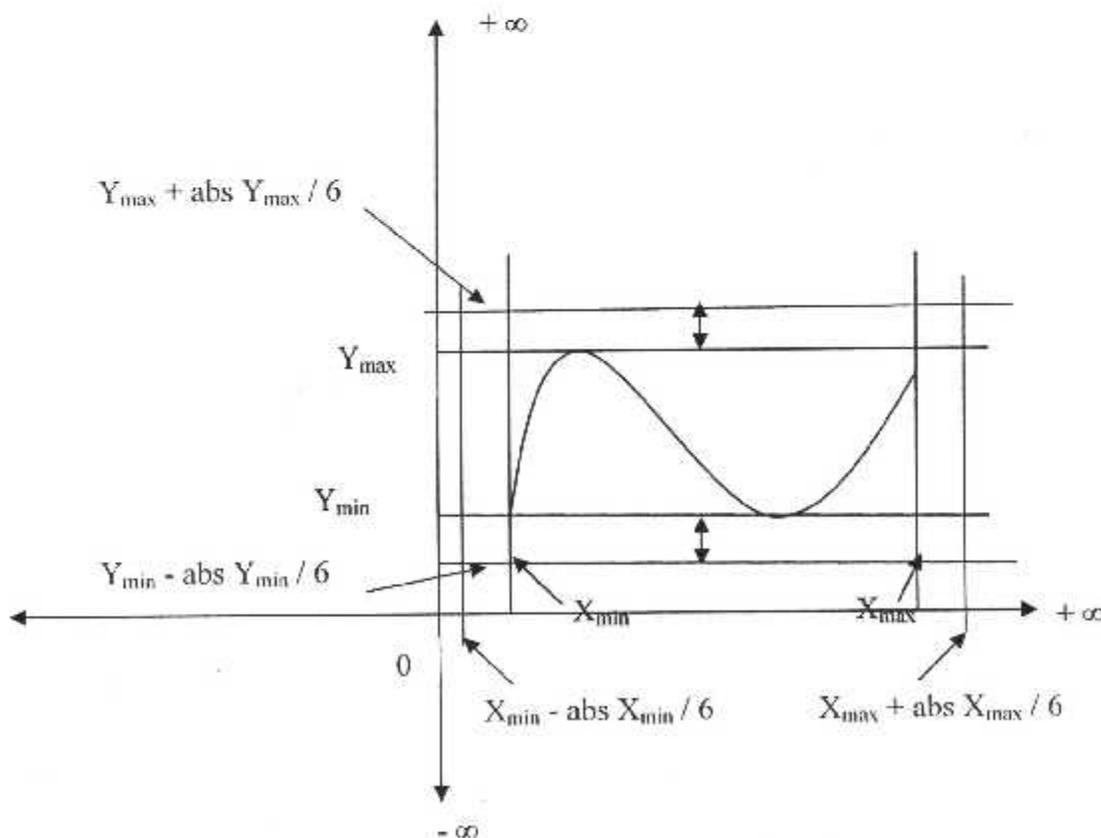


Fig III-2 Exemple de création des axes

(xmin, ymin) sont les coordonnées du coin inférieur gauche, tandis que (xmax, ymax), celles du coin supérieur droit.

Les figures III-3, III-4 et III-5 représentent respectivement les organigrammes de la visualisation de l'interface de la création des axes et de la visualisation du signal.

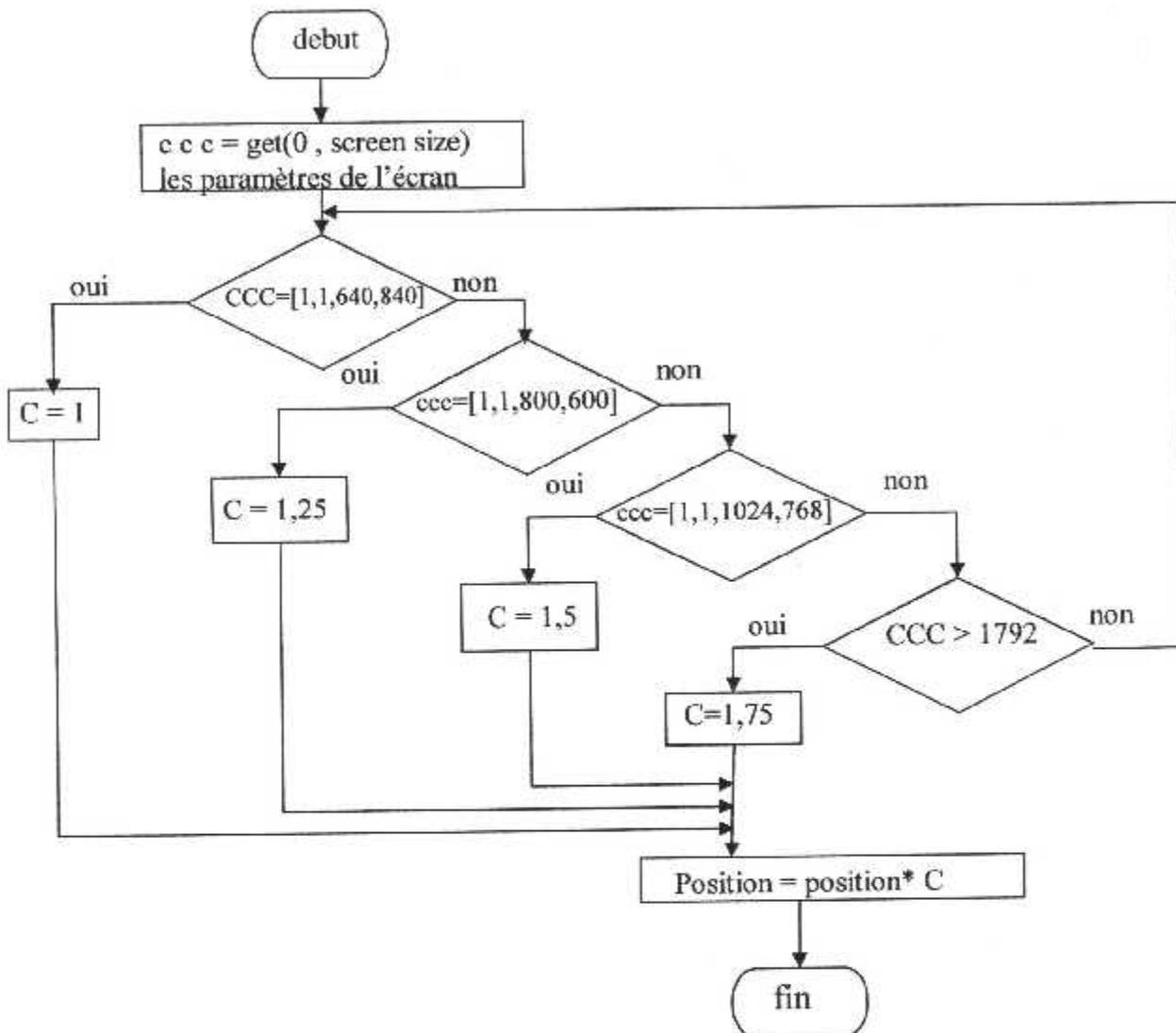


Fig III-3 Organigramme de la procédure « visualisation de l'interface »

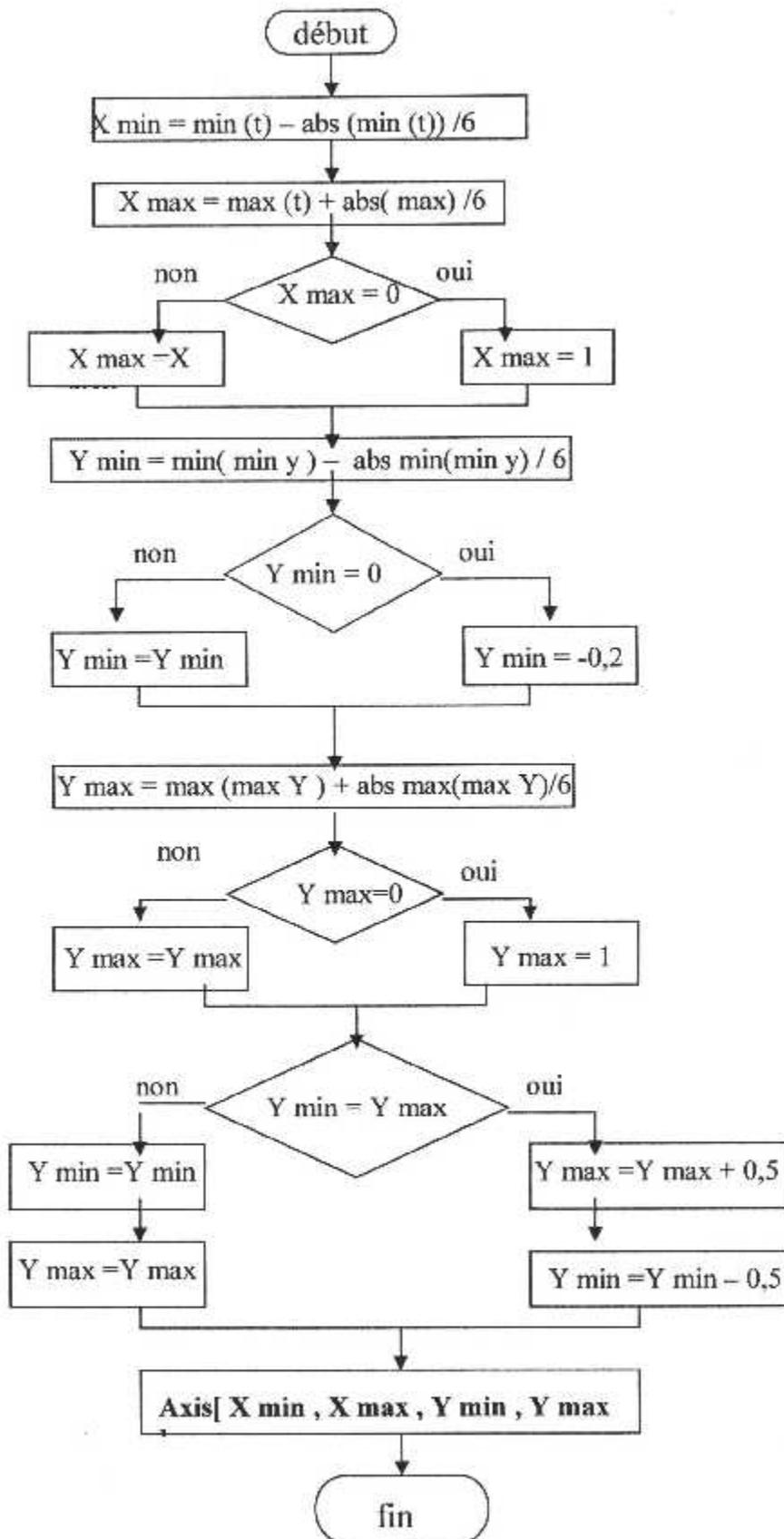


Fig III-4 a : Organigramme de la procédure « création des axes » pour les signaux simples

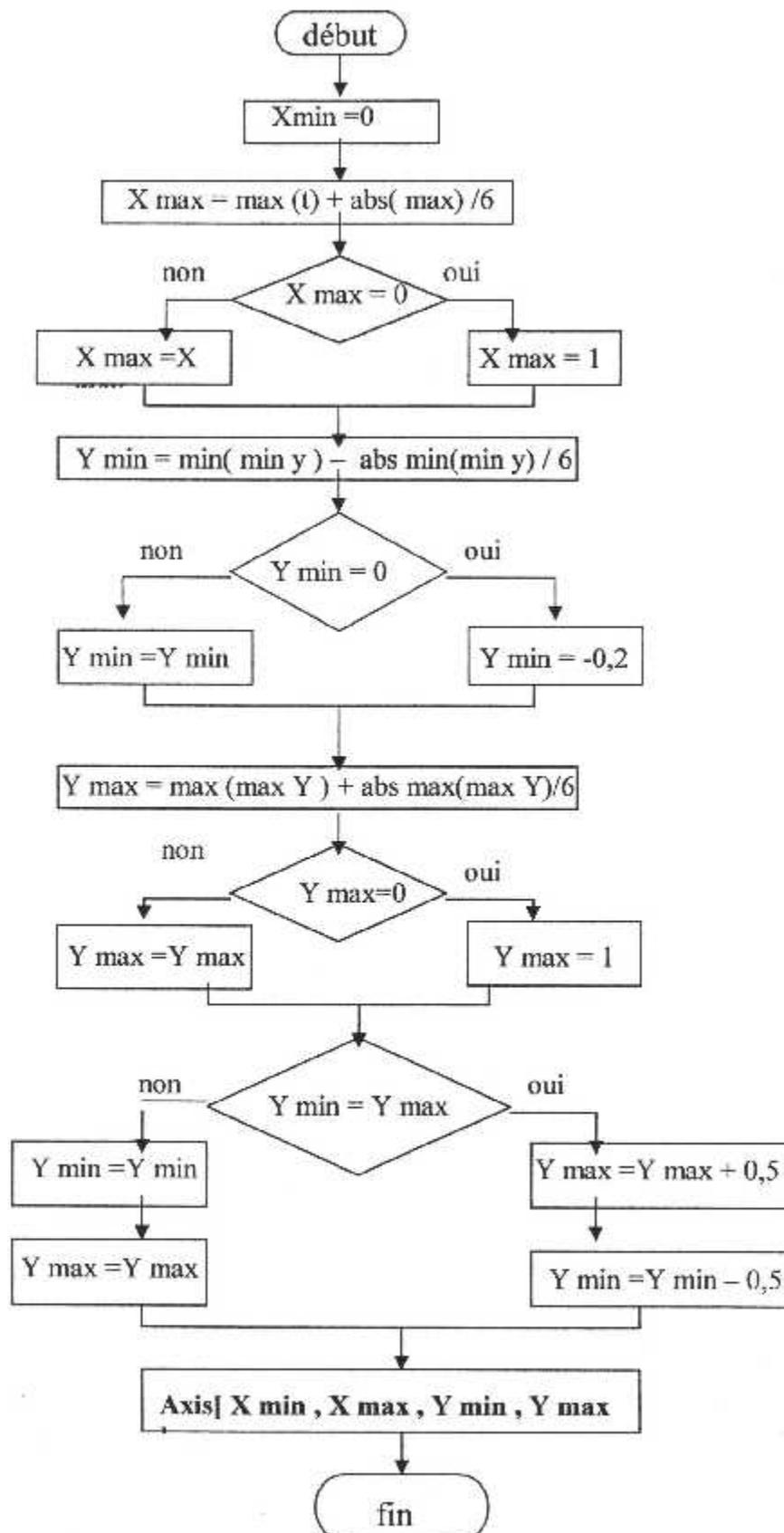


Fig III-4 b: Organigramme de la procédure « création des axes » pour les signaux *wav

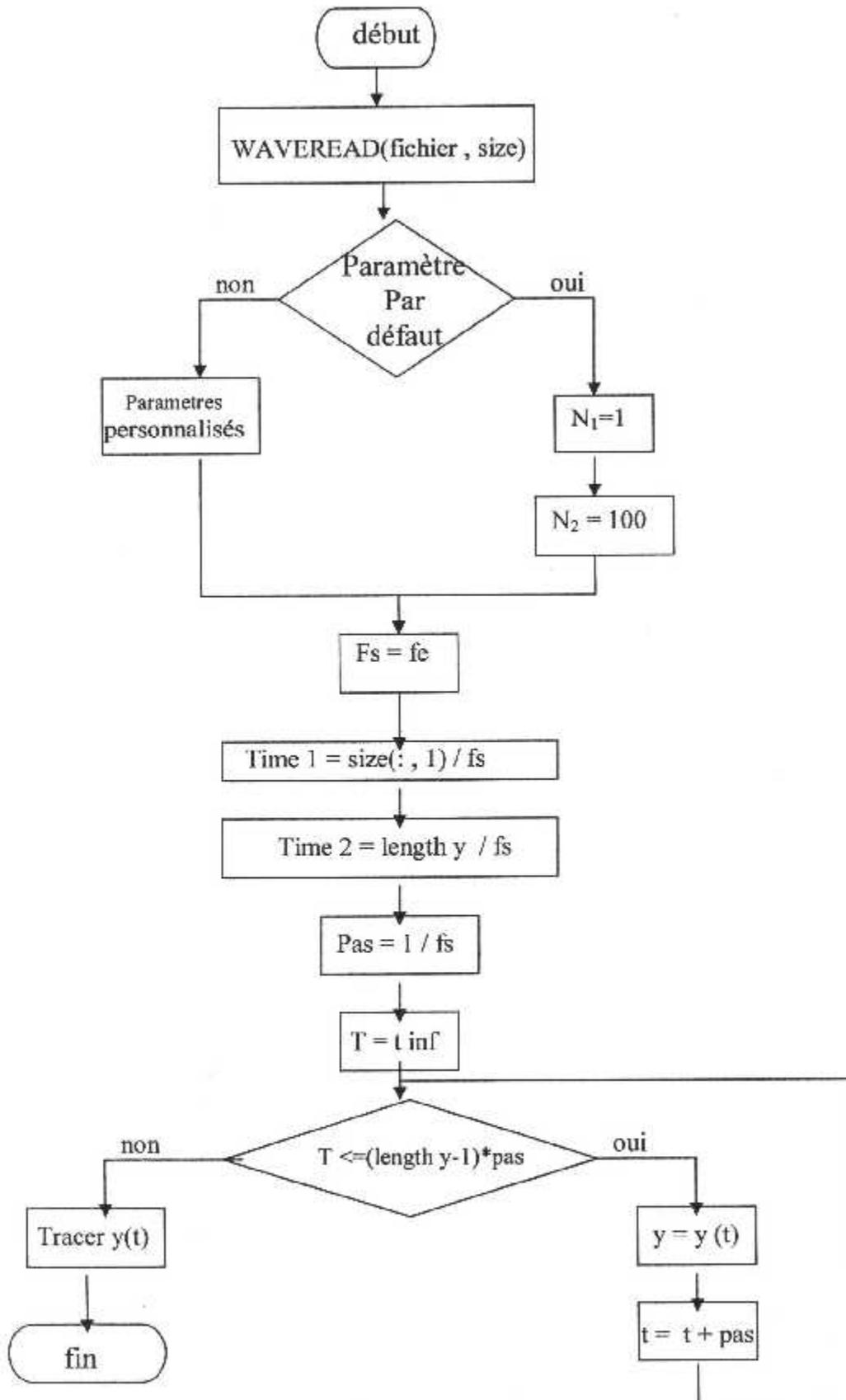


Fig III-5 Organigramme de la procédure « visualisation du signal »

III-3-2 Procédure de dilatation du signal (zoom)

La procédure « zoom » permet de visualiser à la loupe le signal. On établit l'échelle. L'axe des Y est exprimé en volt alors que l'axe des X est exprimé en bloc. Un bloc contient 256 échantillons équivalents à 256 pixels.

C= bouton à cliquer

A1= +Xmin

A2= -Xmin

A3= +Xmax

A4= -Xmax

A5= +Ymin

A6= -Ymin

A7= +Ymax

A8= -Ymax

Les figures III-6 et III-7 représentent le graphe du signal Dents De Scie zoomé et l'organigramme de la procédure zoom

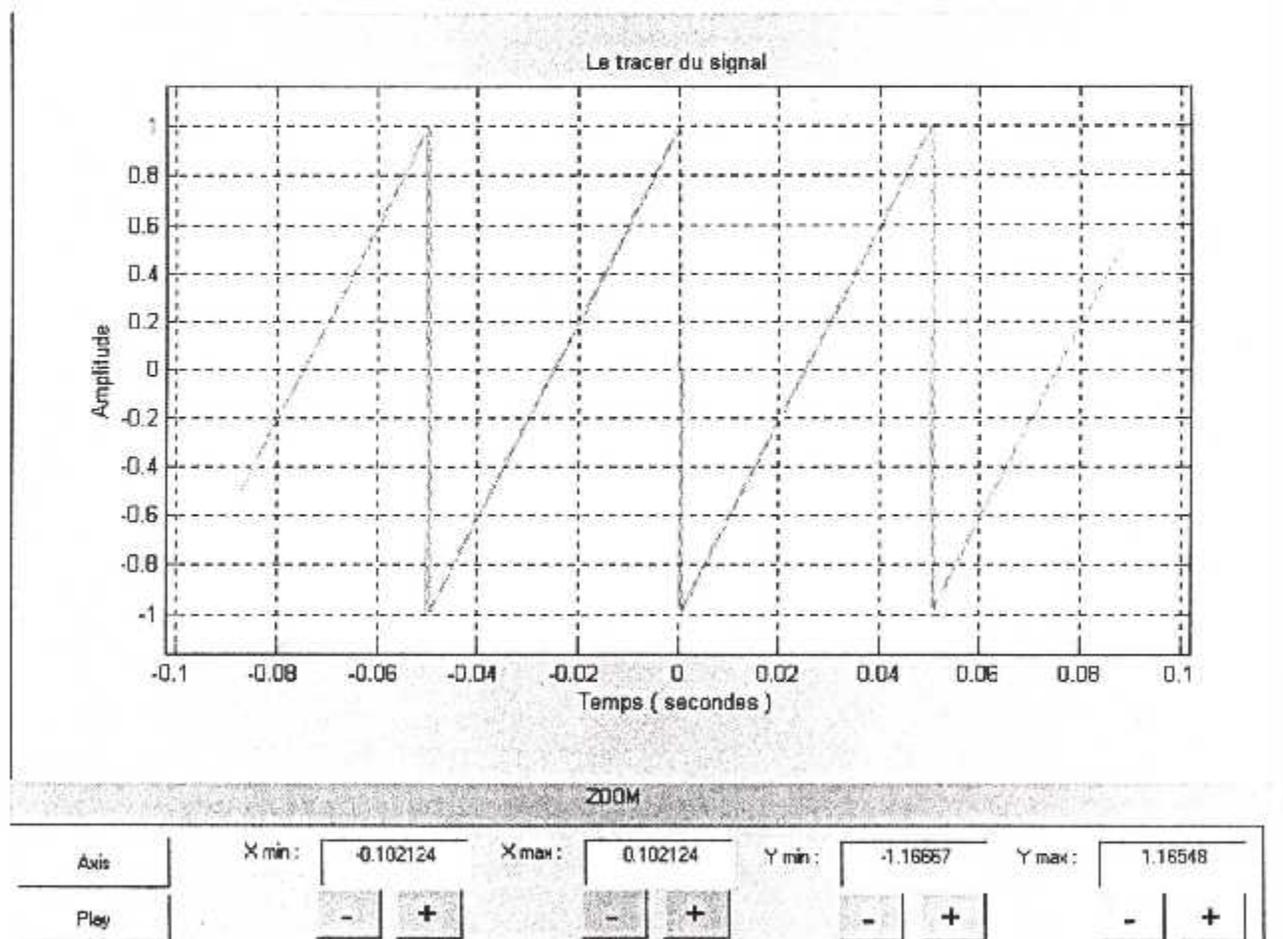
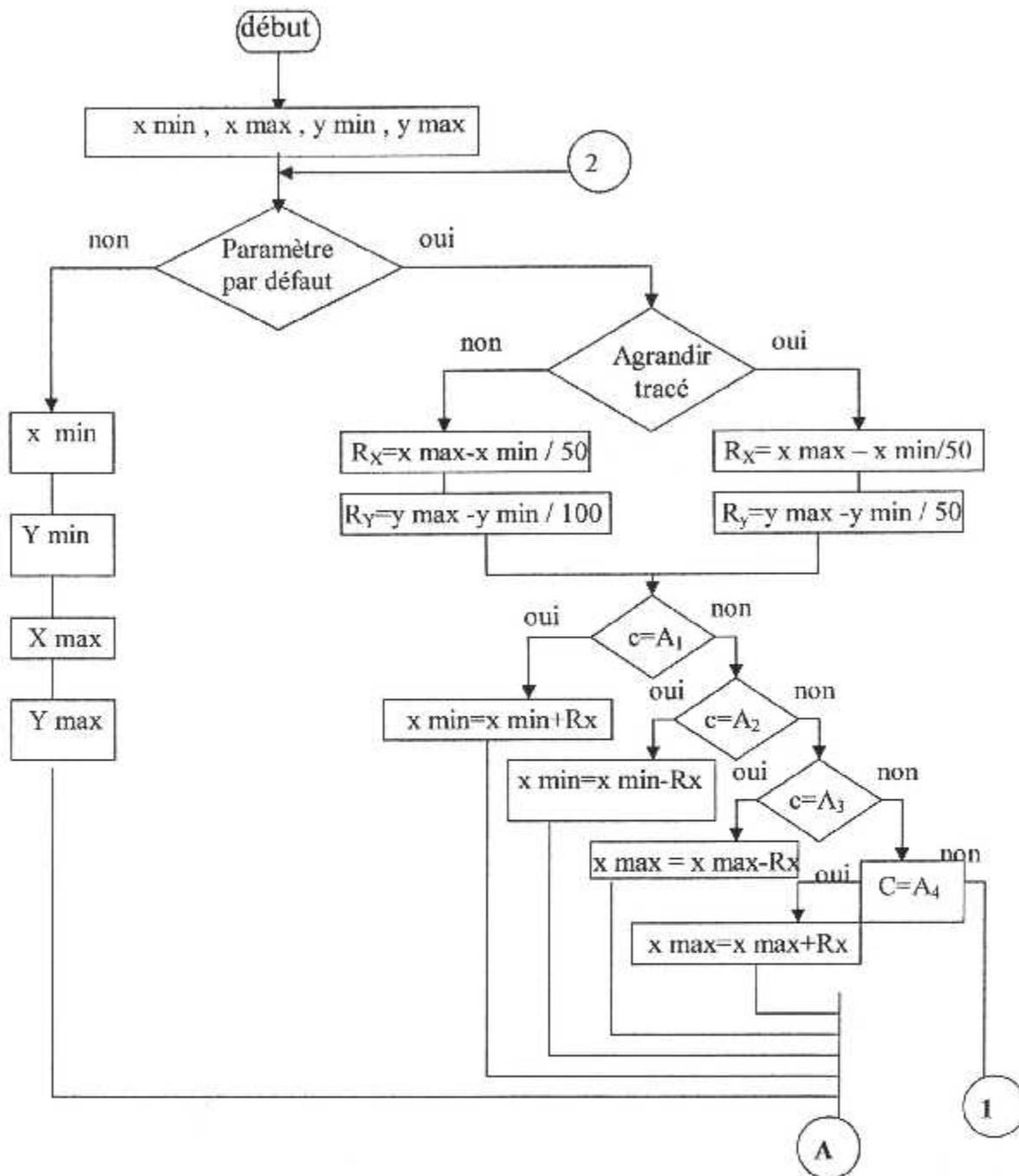


fig III-6 Signal Dents de scie zoomé



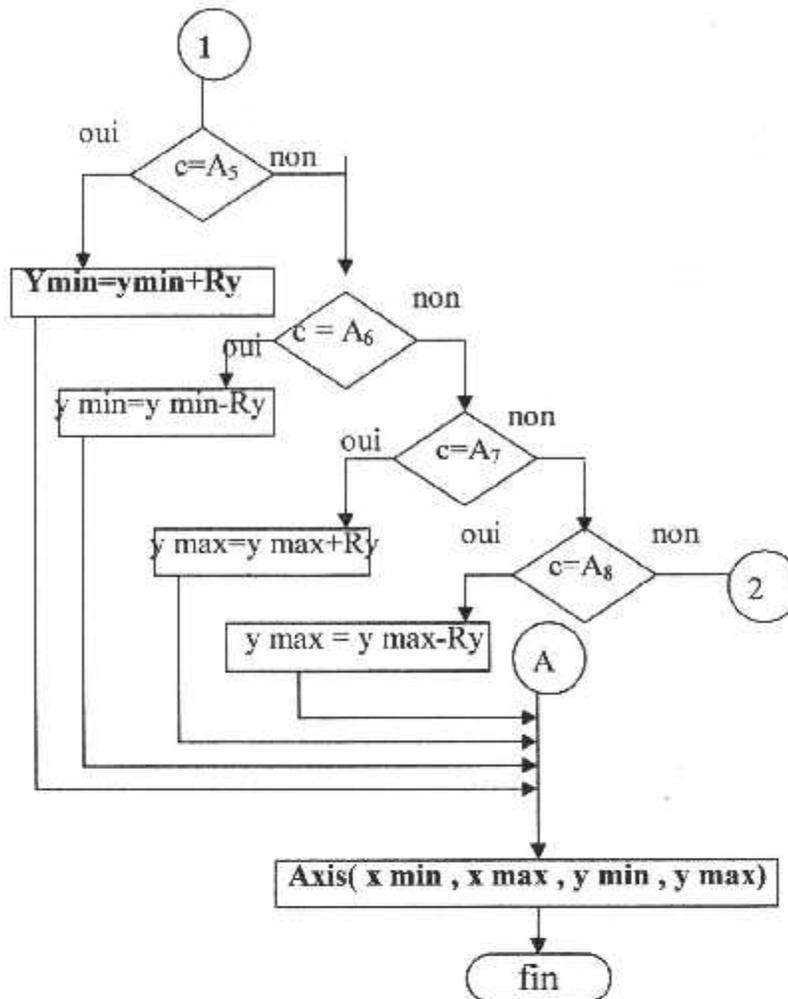


Fig III-7 Organigramme de la procédure « zoom »

III-3-3 Spectre du signal

L'analyse spectrale consiste à déterminer les caractéristiques du signal dans le domaine fréquentiel.

Le domaine fréquentiel de la parole va de quelques hertz à 5KHz. Ces analyses font essentiellement appel à deux techniques fondamentales :

- la numérisation du signal analogique par échantillonnage,
- la transformée de Fourier discrète réalisée selon l'algorithme de la FFT (Fast Fourier Transform).

III-3-3-1 Echantillonnage d'un signal analogique

Un signal échantillonné est un signal à temps discret. Il n'est défini que par des valeurs particulières du temps. Cela signifie que la variable temps est quantifiée. Dans le cas général cette discrétisation est à pas constant donc périodique de période : $f_e = 1/T_e$

La figure III-8 décrit un modèle d'échantillonnage aléatoire.

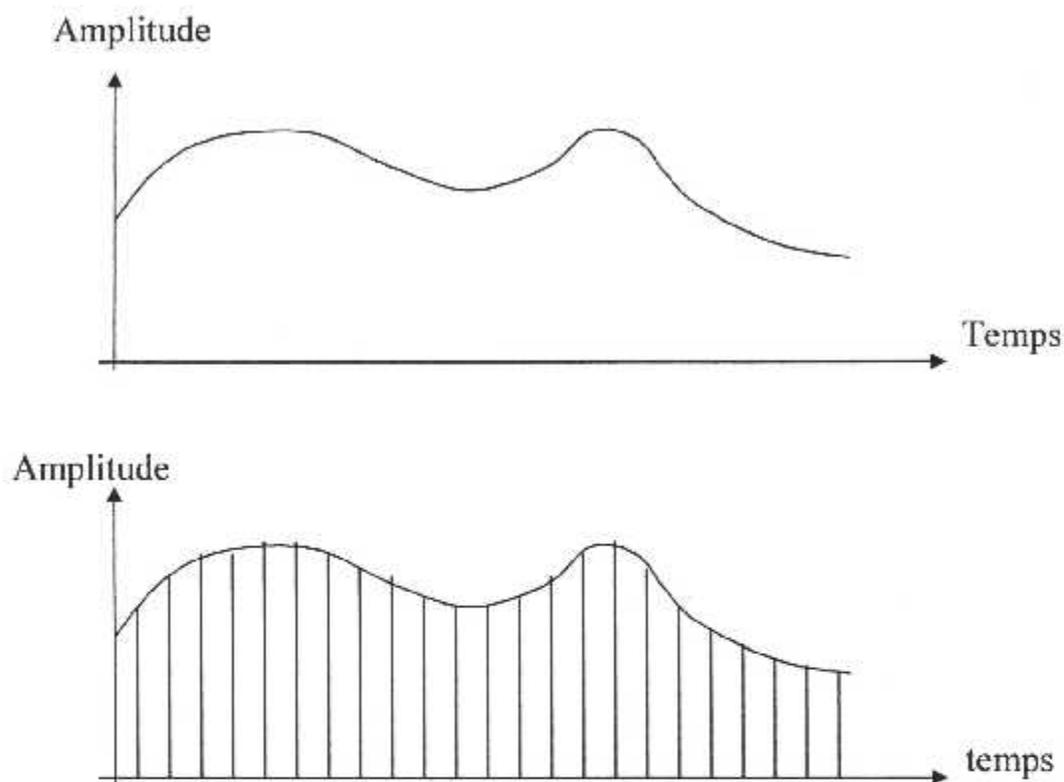


Fig III-8 Modèle d'échantillonnage aléatoire

Dans la procédure « echt », on procède au calcul d'échantillonnage tel le nombre d'échantillons par période, le nombre d'échantillons par bloc, le nombre de bloc...etc. On fixe alors la fréquence d'échantillonnage tout en respectant le théorème de « SHANNON » qui s'énonce comme suit :

"L'information véhiculée par un signal dont le spectre est à support borné n'est pas modifiée par l'opération d'échantillonnage à condition que la fréquence d'échantillonnage soit au moins deux fois plus grande que la plus grande fréquence contenue dans le signal.

La reconstitution du signal original peut être effectuée par un filtre passe-bas idéal de fréquence de coupure égale à la moitié de la fréquence d'échantillonnage."

$$F_e \geq 2v_{\max}$$

Où : F_e est la fréquence d'échantillonnage et v_{\max} la fréquence maximale du signal.

Cependant, les physiciens, par soucis de fiabilité donnent une autre formule qui est :

$$F_e \geq nv_{\max}$$

n: nombre entier variant de 5 à 50.

III-3-3-2 Fenêtres de pondération (Troncature)

La nécessité d'une fenêtre de pondération est due au fait que l'ordinateur impose une durée limitée au signal ; donc une durée limitée à sa transformée de Fourier. En d'autres termes, ceci revient à multiplier le signal par une fenêtre temporelle naturelle. Il lui correspond dans le domaine fréquentiel une fenêtre spectrale.

Pour diminuer les effets d'une telle délimitation, on superpose à la fenêtre temporelle naturelle une autre fenêtre qu'on appellera « pondération ».

Deux paramètres principaux caractérisent ces fenêtres :

- largeur de base du pic central
- le rapport de l'amplitude du premier lobe secondaire à celle du pic central qui est exprimée en « dB ».

$$\lambda_i = 20 \log W_i(f_s)/W_i(0)$$

$f(s)$: fréquence au milieu du lobe secondaire de la fenêtre particulière $W_i(f)$

Il existe différentes sortes de fenêtres de troncature, on en a utilisé huit dont les caractéristiques sont les suivantes :

Dans les tableaux ci-dessous, la colonne de gauche représente la fenêtre, la colonne centrale le spectre linéaire et la colonne de droite le spectre logarithmique (zoom sur les 400 premiers points dans ces deux derniers cas).

Les fenêtres ci-dessous ont été calculées par MATLAB à partir d'une fenêtre de 64 points et d'un calcul de FFT sur 4096 points.

a) Fenêtre rectangulaire

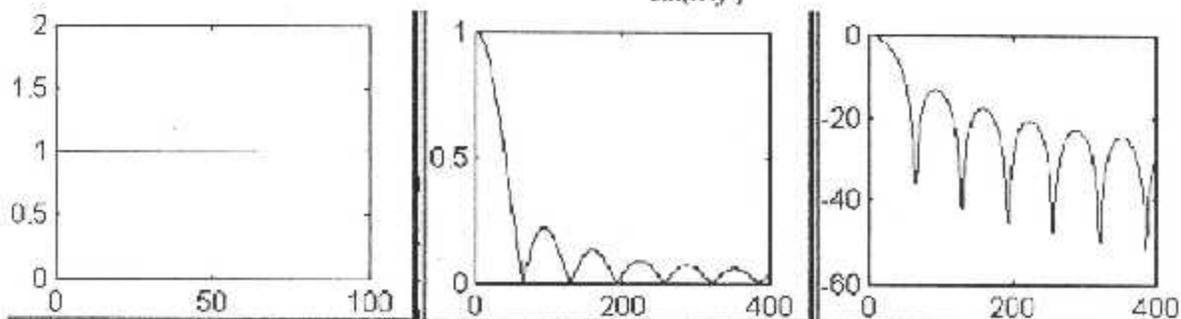
Cette fenêtre est définie par :

$$\begin{cases} f_n = 1 & \text{pour } 0 \leq n \leq N - 1 \\ f_n = 0 & \text{ailleurs} \end{cases}$$

On peut également considérer cette fenêtre comme étant le cas particulier de la fenêtre de Hamming généralisée pour $\alpha = 1$ (cf. fenêtre de Hamming).

Son expression fréquentielle est :

$$W_R(e^{j2\pi f}) = e^{-j\pi(N-1)f} \cdot \frac{\sin(\pi N f)}{\sin(\pi f)}$$



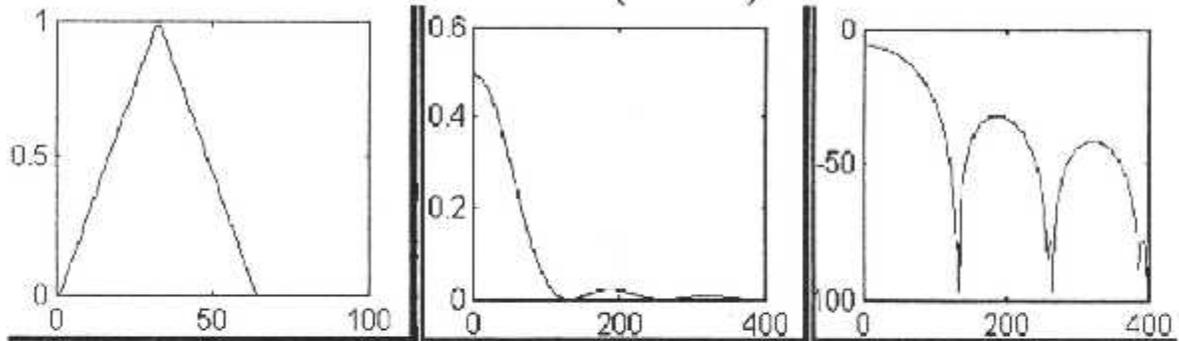
b) Fenêtre triangulaire ou de Bartlett

Cette fenêtre est définie par :

$$\begin{cases} f_n = \frac{n}{N/2} \text{ pour } 0 \leq n \leq N/2 \\ f_{N-n} \text{ pour } N/2 \leq n \leq N-1 \end{cases}$$

Son expression fréquentielle est :

$$W_T(e^{j2\pi f}) = e^{-j2\pi(N/2-1)f} \left(\frac{\sin(\pi \cdot \frac{N}{2} \cdot f)}{\sin(\pi \cdot f)} \right)^2$$



c) Fenêtre de Hamming

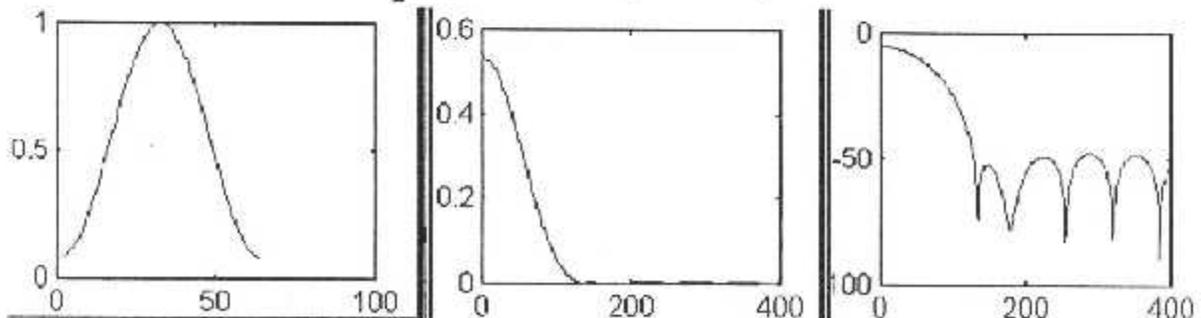
La fenêtre de Hamming généralisée a pour équation :

$$\begin{cases} f_n = \alpha + (1 - \alpha) \cdot \cos\left(\frac{2\pi \cdot n}{N}\right) \text{ pour } 0 \leq n \leq N - 1 \\ f_n = 0 \text{ ailleurs} \end{cases}$$

La fenêtre de Hamming proprement dite est le cas particulier de la fenêtre généralisée pour $\alpha = 0,54$.

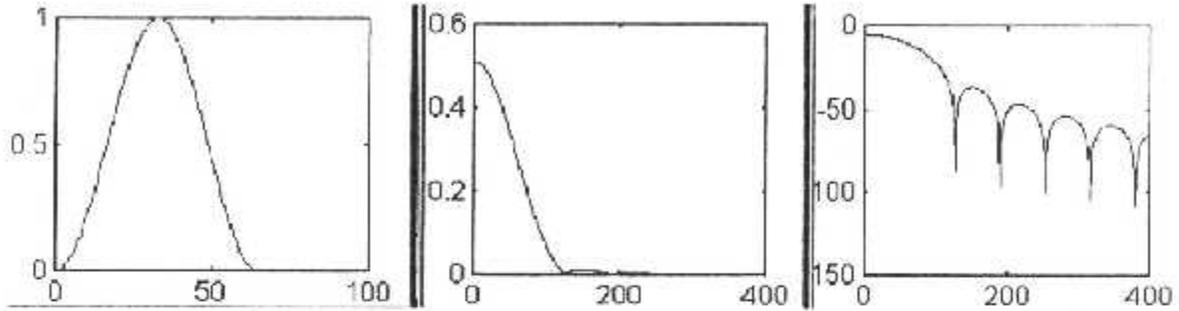
Les fenêtres de la famille Hamming se caractérisent par un pic central de largeur double de la fenêtre rectangulaire mais une atténuation des oscillations sensiblement plus importante. La représentation fréquentielle de la fenêtre de Hamming généralisée a pour équation :

$$W_{Hg}(e^{j2\pi f}) = \alpha W_R + \frac{1-\alpha}{2} [W_R(e^{j2\pi f(1-1/N)}) + W_R(e^{j2\pi f(1+1/N)})]$$



d) Fenêtre de Hanning

La fenêtre de Hann (ou Hanning) est le cas particulier de la fenêtre de Hamming généralisée pour $\alpha = 0,5$



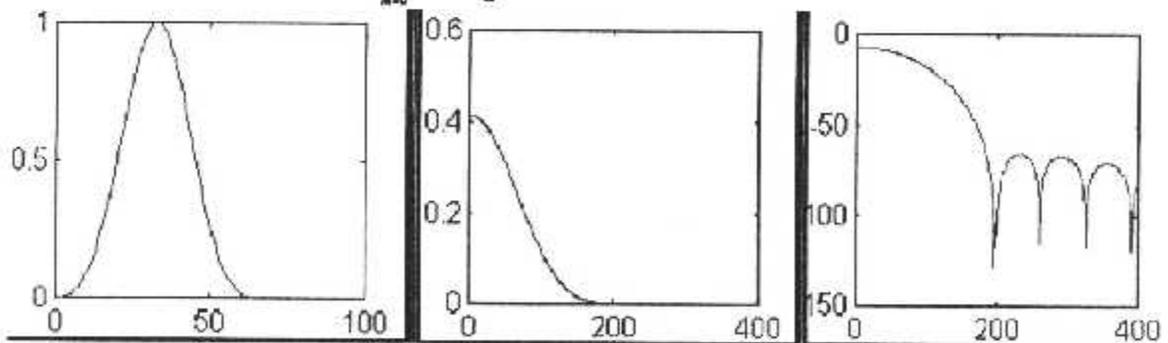
e) Fenêtre de Blackman

Cette fenêtre est une extension de l'idée de Hamming. Le premier lobe est plus large mais l'atténuation est encore plus élevée. Elle est définie par :

$$\begin{cases} f_n = \sum_{m=0}^{M-1} (-1)^m \alpha_m \cdot \cos\left(\frac{2\pi \cdot n \cdot m}{N}\right) \text{ pour } 0 \leq n \leq N-1 \\ f_n = 0 \text{ ailleurs} \end{cases}$$

Avec la contrainte : $\sum_{m=0}^{M-1} \alpha_m = 1$. Son expression fréquentielle est :

$$W_n(e^{j2\pi f}) = \sum_{m=0}^{M-1} (-1)^m \cdot \frac{\alpha_m}{2} [W_n(e^{j2\pi(f-m/N)}) + W_n(e^{j2\pi(f+m/N)})]$$



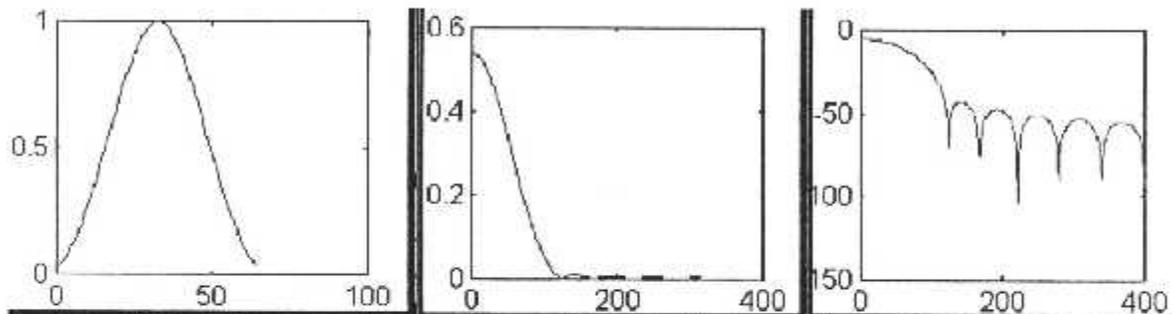
f) Fenêtre de Kaiser (avec $\beta = 5$)

La fenêtre de Kaiser permet de préciser le compromis largeur du pic central/amplitude des oscillations. Elle est définie par :

$$\begin{cases} f_n = \frac{I_0[\beta \sqrt{N^2 - 4.n^2}]}{I_0(\beta N)} \text{ pour } 0 \leq n \leq N-1 \\ f_n = 0 \text{ ailleurs} \end{cases}$$

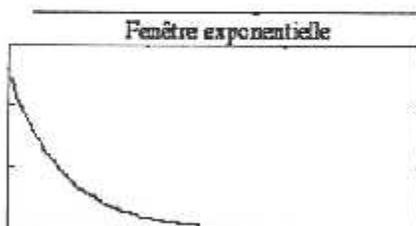
Où I_0 est la fonction de Bessel de première espèce d'ordre 0 :

$$I_0(x) = 1 + \sum_{m=1}^M \left(\frac{(x/2)^m}{m!} \right)^2 \quad (\text{En général, } 4 \leq \beta \leq 9 \text{ et } M = 14).$$



En prenant β de l'ordre de 4,5, l'atténuation est voisine de celle de Hamming mais le pic central est moins large. Pour β de l'ordre de 9, le pic est plus large mais l'atténuation plus importante.

g) Fenêtre exponentielle



Le début du signal n'est pas perturbé, mais la fin de l'enregistrement temporel est forcée à zéro. La fenêtre exponentielle ne convient que pour la mesure des signaux transitoires. Cette fenêtre est définie par :

$$\begin{cases} f_n = B.e^{-an} \text{ pour } t \geq 0 \\ f_n = 0 \text{ ailleurs} \end{cases}$$

La fenêtre à réponse exponentielle est utile pour la mesure des signaux transitoires (de type impulsion).

Son expression fréquentielle est :

$$W_E(e^{j.2\pi.f}) = \frac{1}{j.2\pi.f + a}$$

La figure III-9 montre l'organigramme de la procédure choix de la fenêtre de pondération.

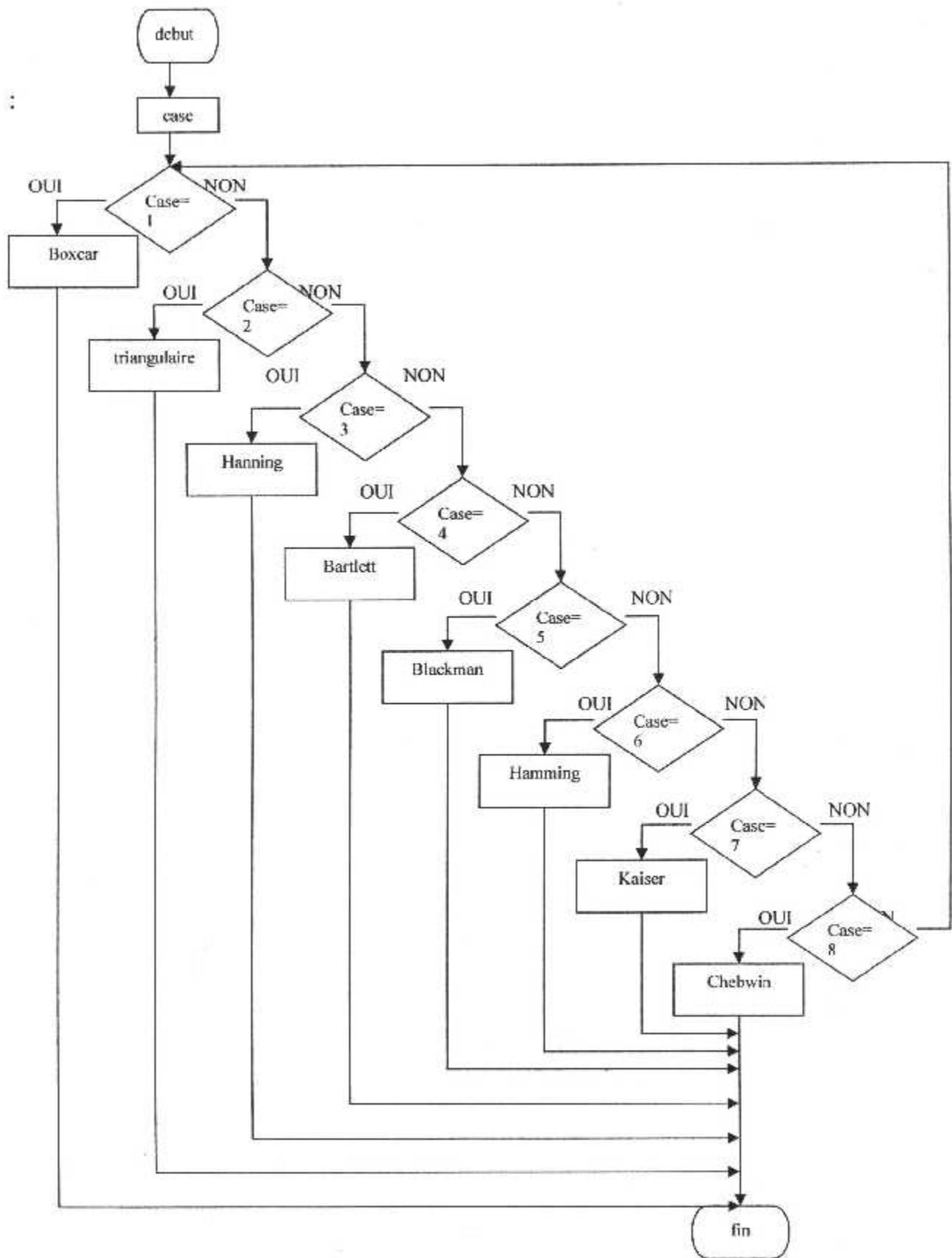


Fig III-9 Organigramme de la procédure « choix de la fenêtre de pondération »

III-3-3-3 Représentation graphique du spectre

A travers cette routine, nous offrons à l'utilisateur le choix entre trois types de tracé pour les spectres d'amplitude et de phase.

- Le tracé en « plot »
- Le tracé en « stem »
- Le tracé en « bar »

Si le signal est périodique, nous utilisons la FFT et dans le cas contraire la DTFT.

Les figures III-10 et III-11 montrent les organigrammes des procédures choix du type de tracé et du tracé des spectres d'amplitude et de phase.

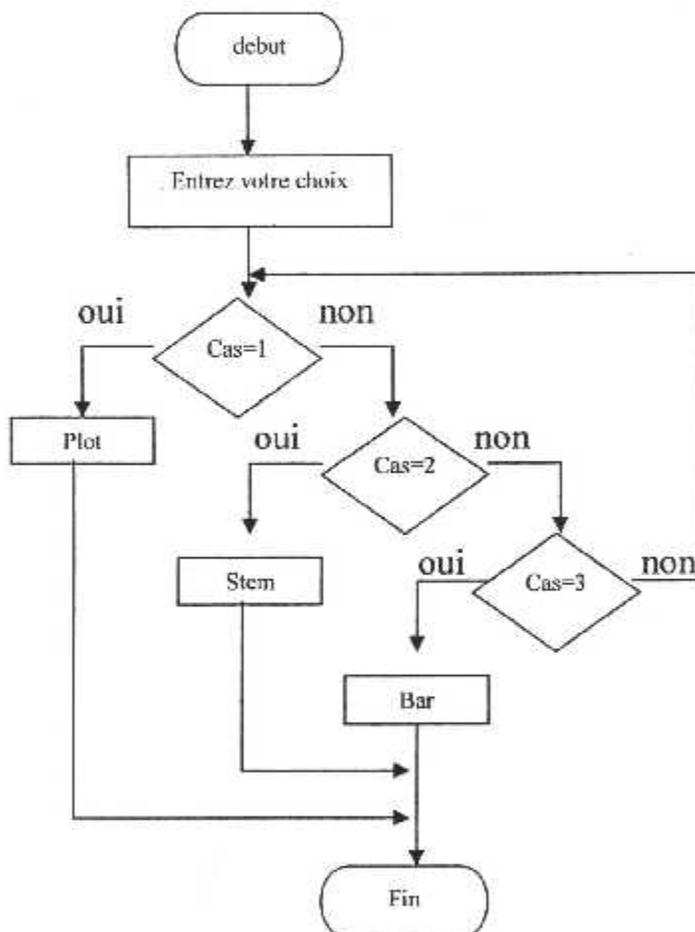


Fig III-10 Organigramme de la procédure « choix du type de tracé »

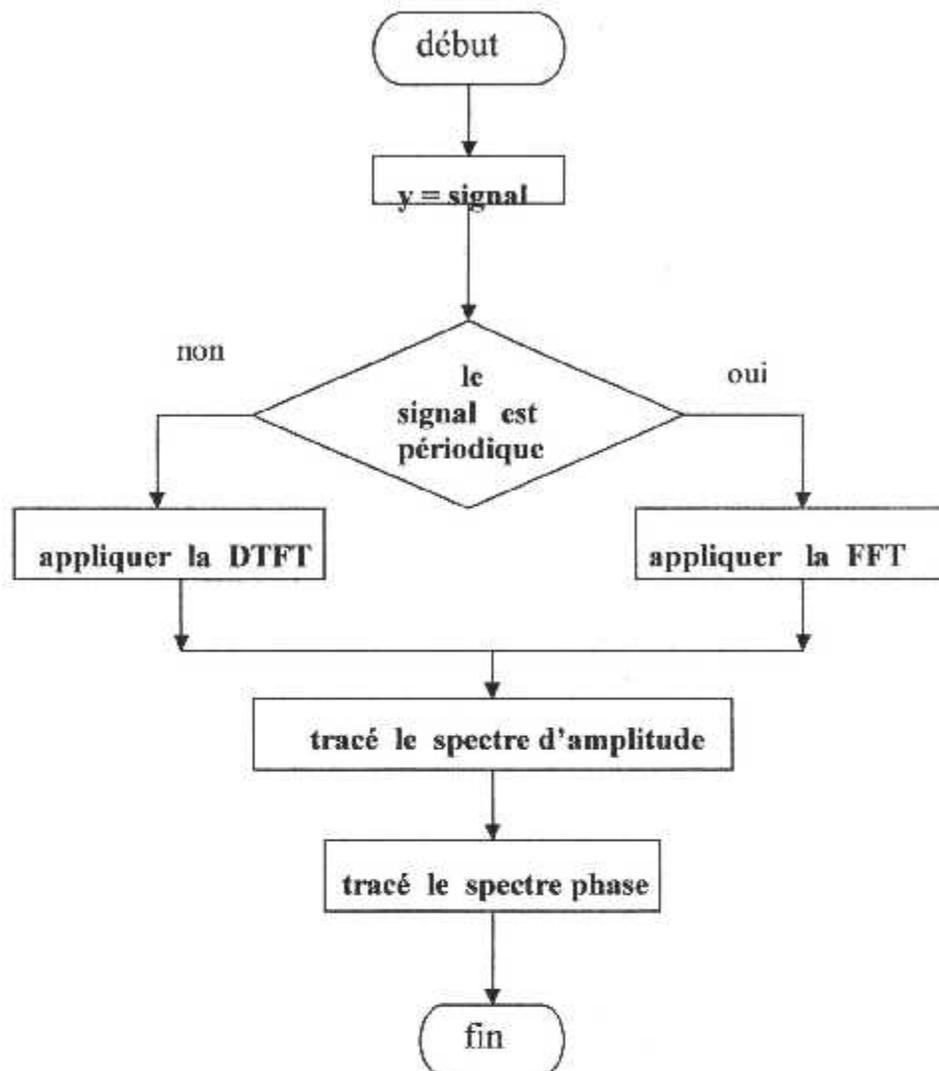
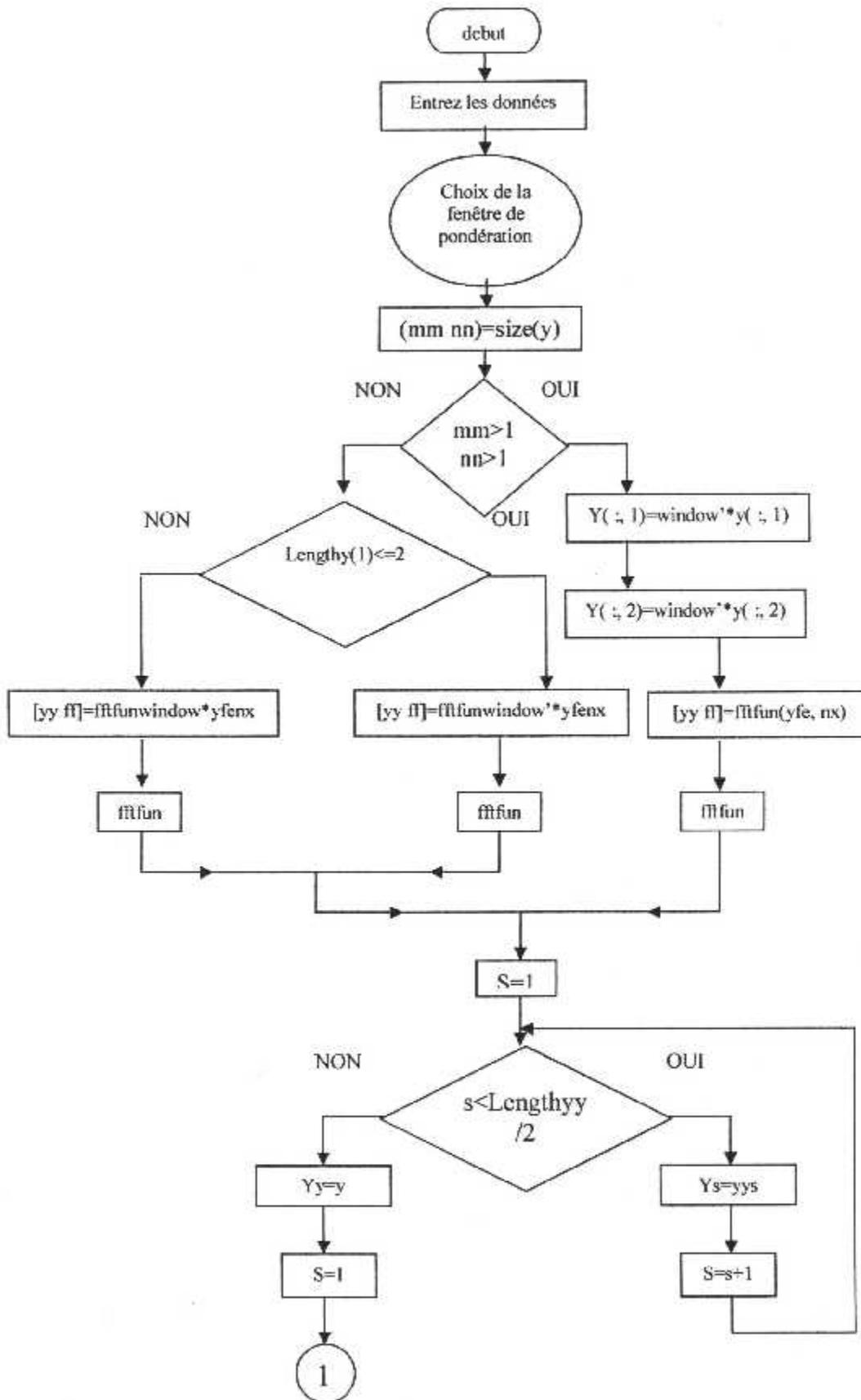


Fig III-11 Organigramme de la procédure « tracé des spectres d'amplitude et de phase »

- Tracé du spectre à partir de la FFT et de la DTFT :

Ces procédures englobent outre le choix de la fenêtre de pondération, mais aussi et surtout le choix du type de tracé et celui des axes pour le tracé.

Les figures III-12, III-13, III-14, III-15 montrent les organigrammes du tracé du spectre à partir de la FFT, à partir de la DTFT, le choix des axes pour le tracé des spectres d'amplitude et de phase.



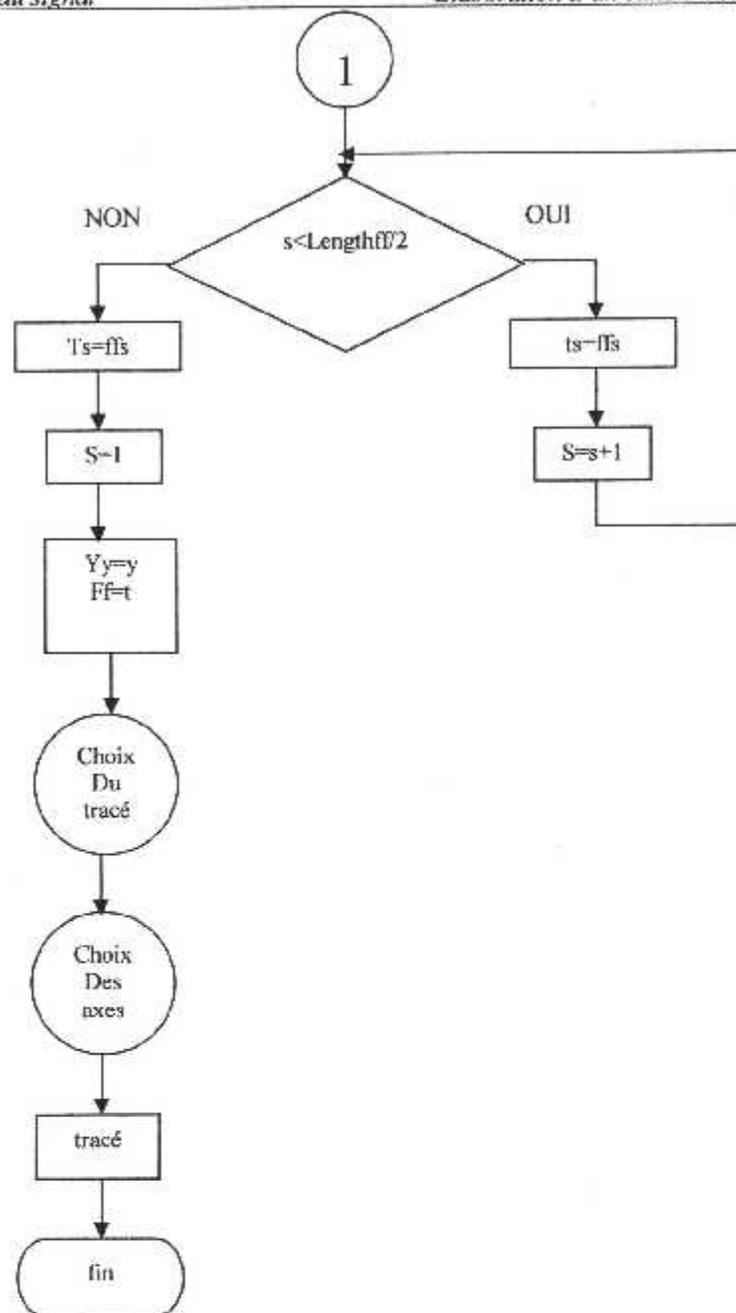


Fig III-12 Organigramme de la procédure « tracé du spectre à partir de la FFT »

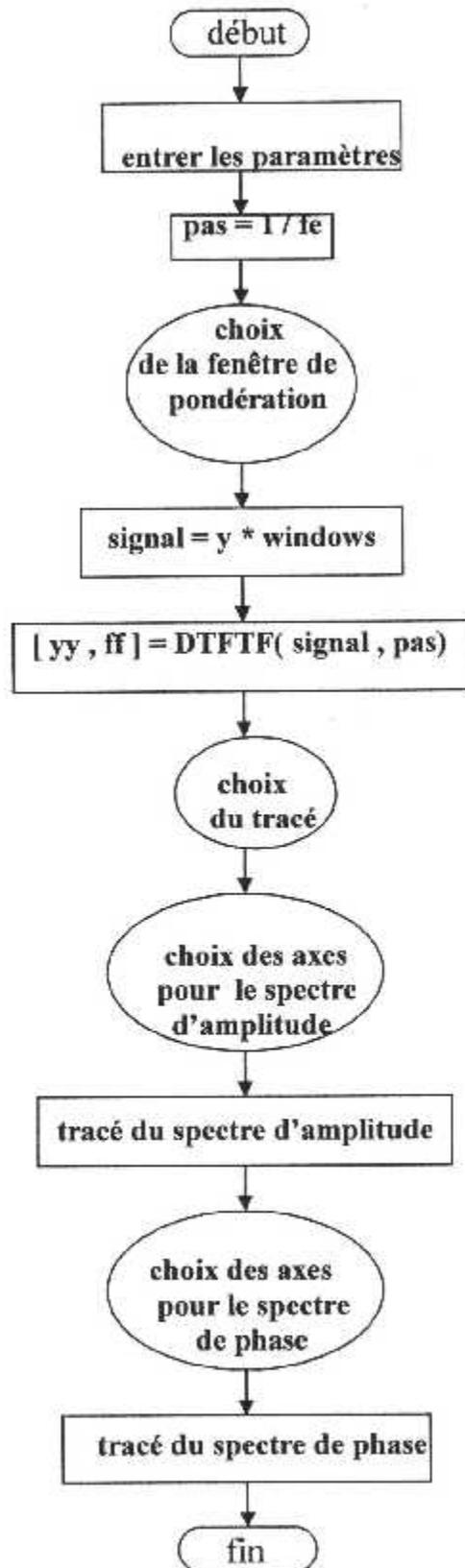


Fig III-13 Organigramme de la procédure « tracé du spectre à partir de la DTFT »

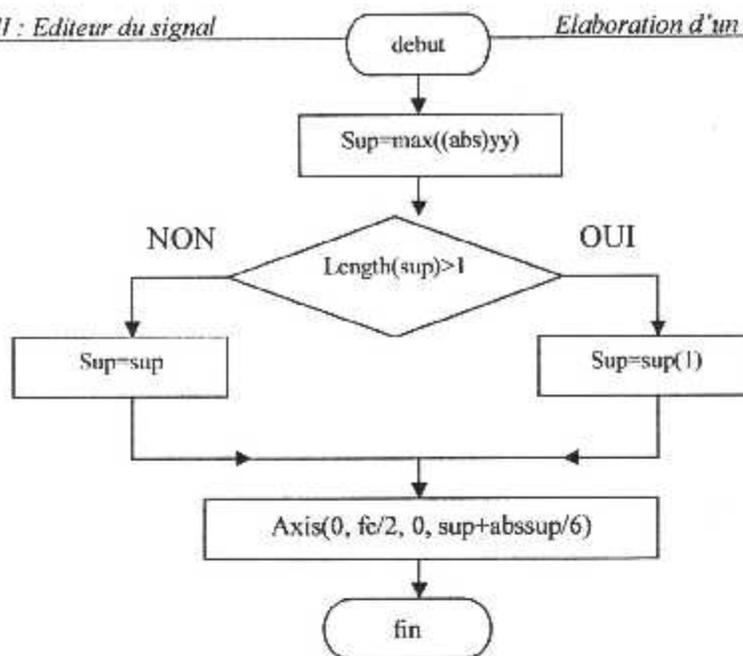


Fig III-14 Organigramme de la procédure « choix des axes pour le spectre d'amplitude »

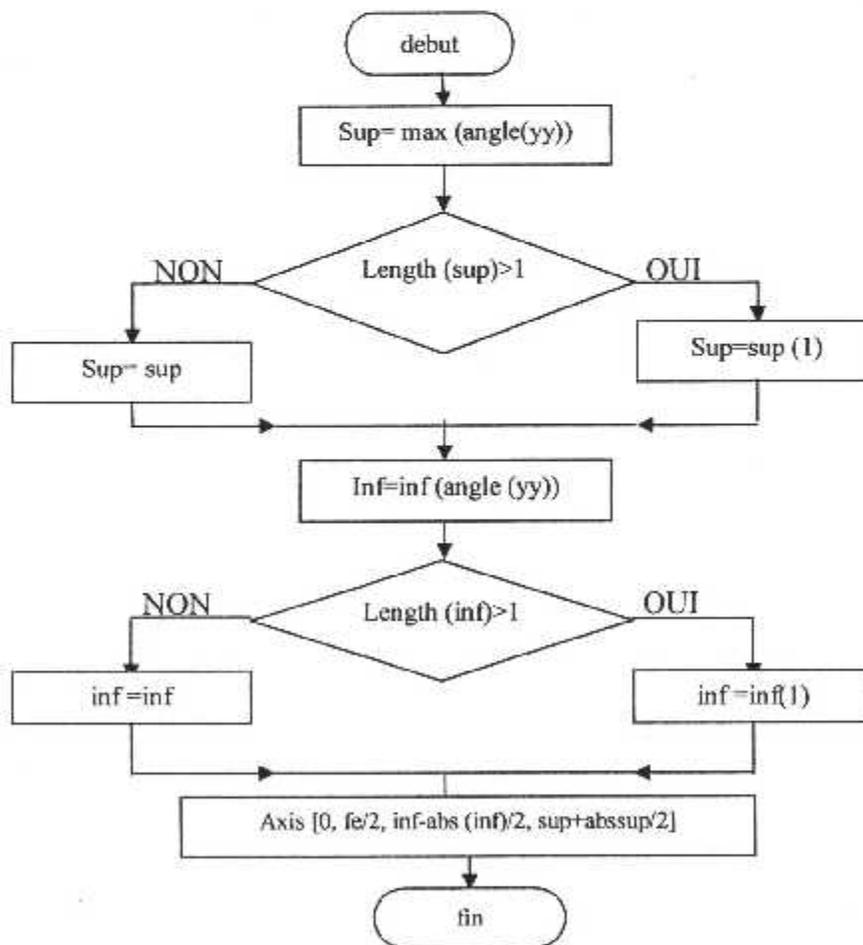


Fig III-15 Organigramme de la procédure « choix des axes pour le spectre de phase »

III-3-4 Procédure de visualisation de la primitive

Cette procédure réalise un filtre intégrateur (filtre linéaire à -6dB).
 La figure III-16 représente l'organigramme de la procédure visualisation de la primitive.

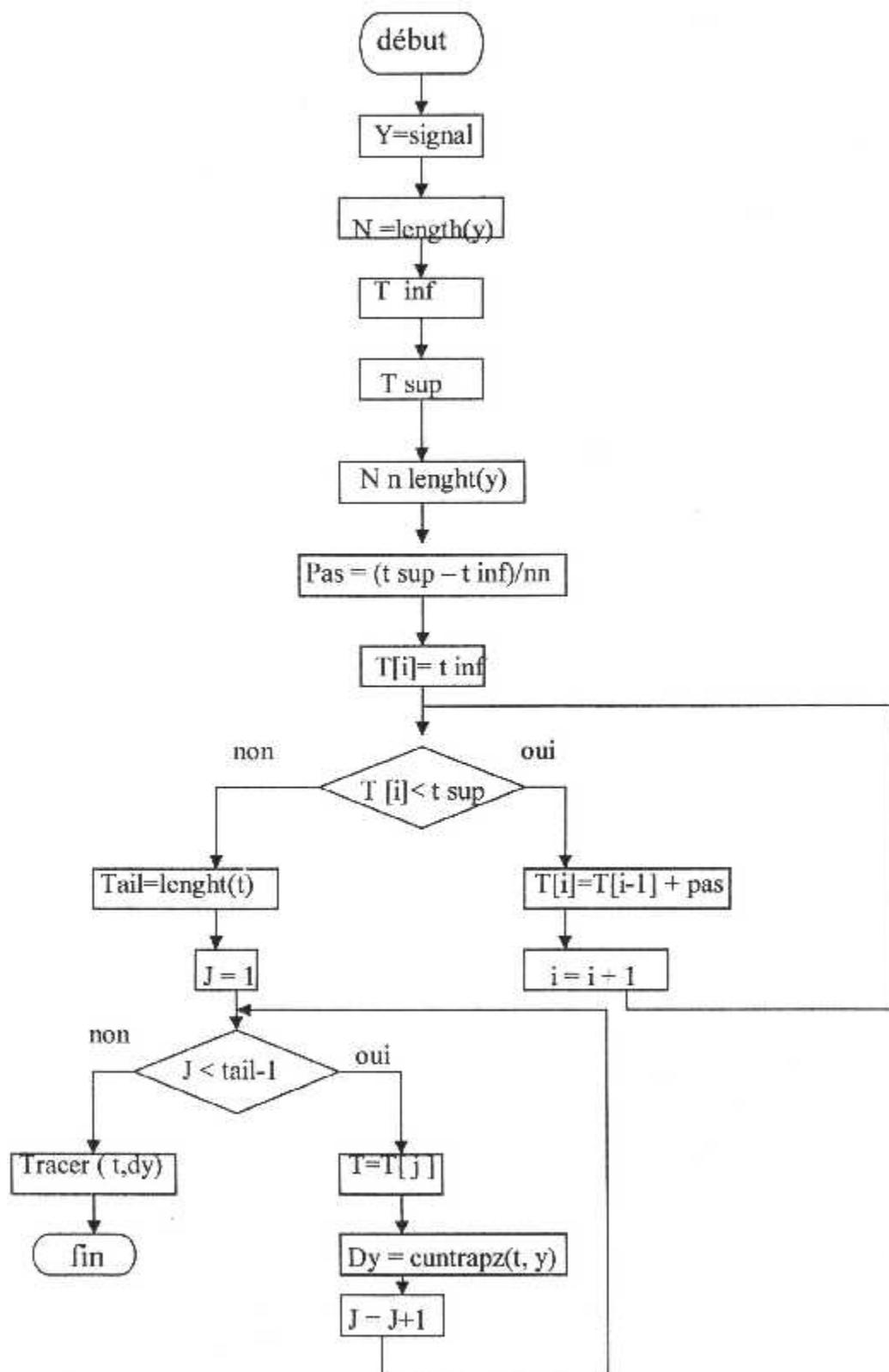


Fig III-16 Organigramme de la procédure « visualisation de la primitive »

III-3-5 Procédure de visualisation de la dérivée du signal

De la même manière que la procédure de visualisation de l'intégrale, cette procédure effectue la dérivée du signal (filtre à +6dB).

L'organigramme de cette procédure est représenté par la figure III-17

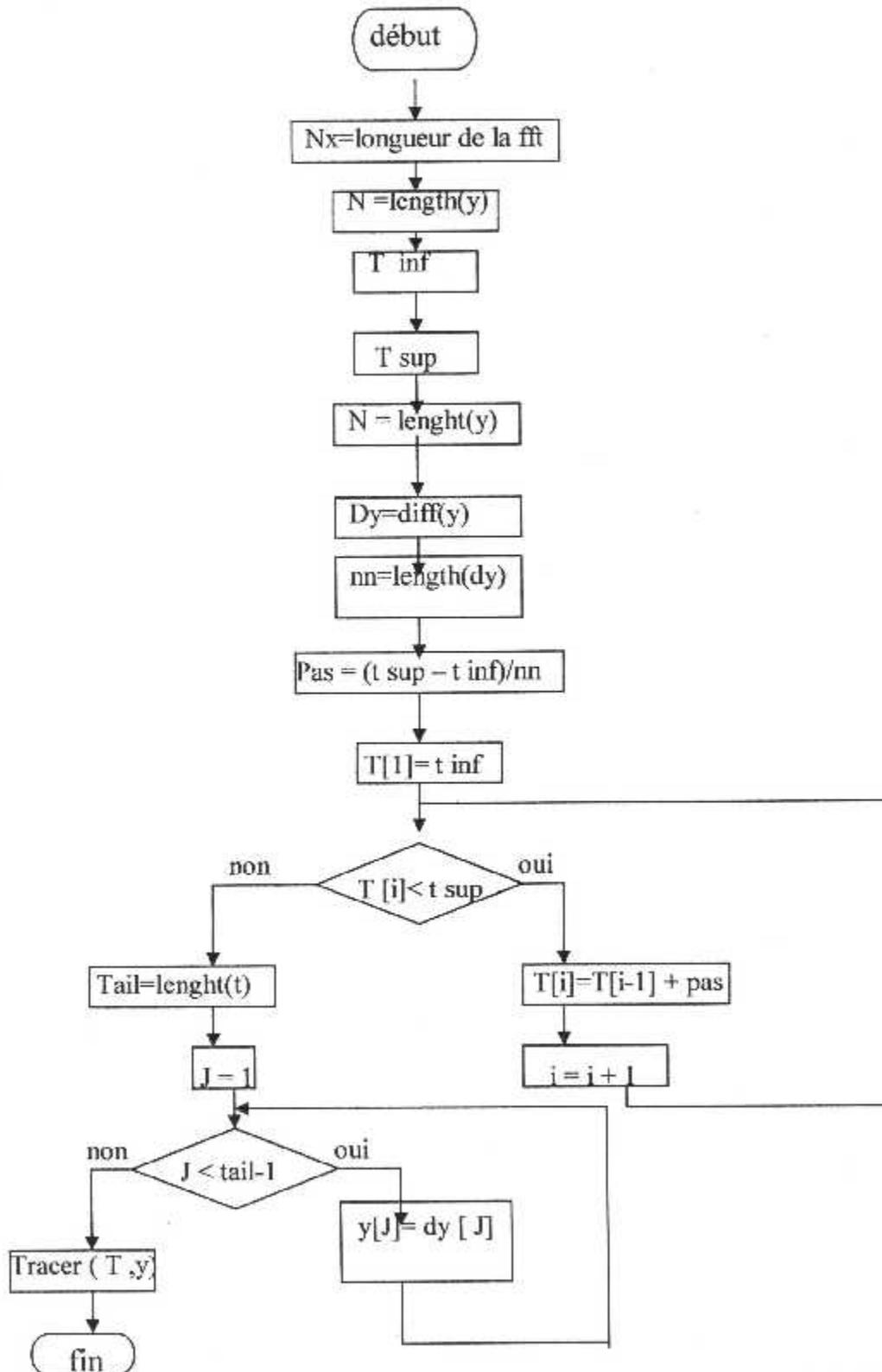


Fig III-17 Organigramme de la procédure « visualisation de la dérivée »

III-3-6 Procédure d'écoute du signal

Cette procédure permet l'écoute du signal ou une partie du signal délimitée par le zoom. L'organigramme de cette procédure est représenté par la figure III-18

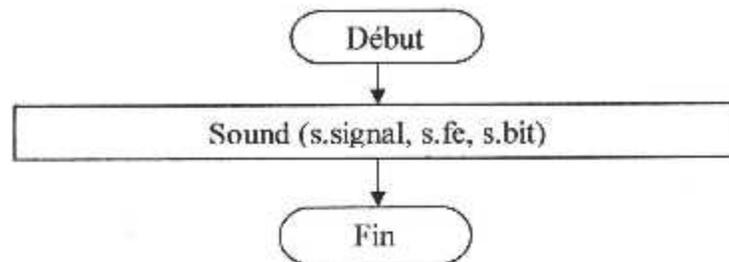


Fig III-18 Organigramme de la procédure « écoute du signal »

III-3-7 Procédure Clear

Cette procédure permet d'effacer les graphes et leurs paramètres à l'intérieur de l'interface. L'organigramme de cette procédure est représenté par la figure III-19

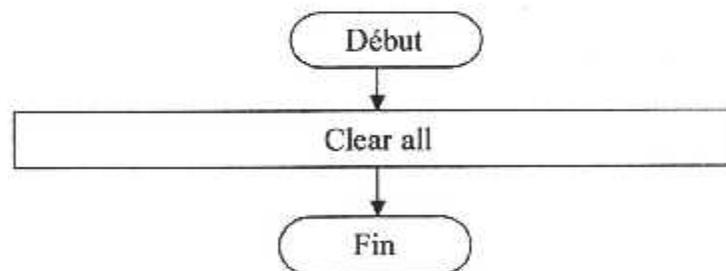


Fig III-19 Organigramme de la procédure « Clear »

III-3-8 Procédure Fin

Cette procédure permet de fermer l'interface. L'organigramme de cette procédure est représenté par la figure III-20

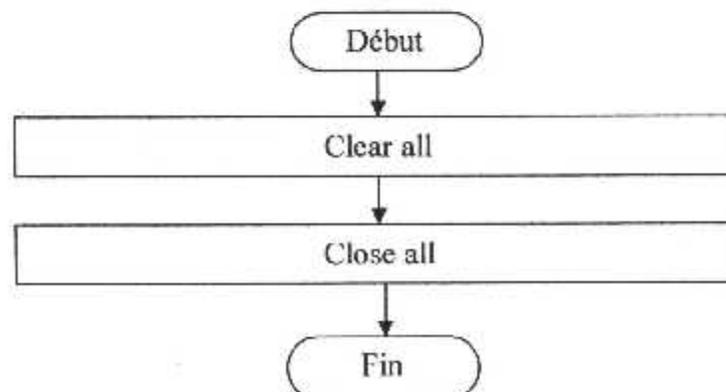
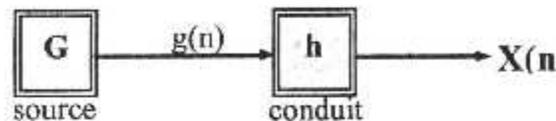


Fig III-20 Organigramme de la procédure « Fin »

III-3-9 Procédure d'analyse cepstrale

Le défaut majeur de la FFT pour le calcul du spectre vocal, réside dans l'intermodulation source/conduit qui rend difficile la mesure des formants et du fondamental. L'analyse cepstrale est une méthode qui vise à séparer leurs contributions respectives par déconvolution. Pour cela on fait l'hypothèse que le signal vocal $x(n)$ est produit par un signal excitateur $g(n)$ (source glottique) traversant un système linéaire passif de réponse impulsionnelle $h(n)$ (conduit vocal).[2]
D'après ces hypothèses, on aura le système suivant :



Donc on peut écrire pour tout $n > 0$:

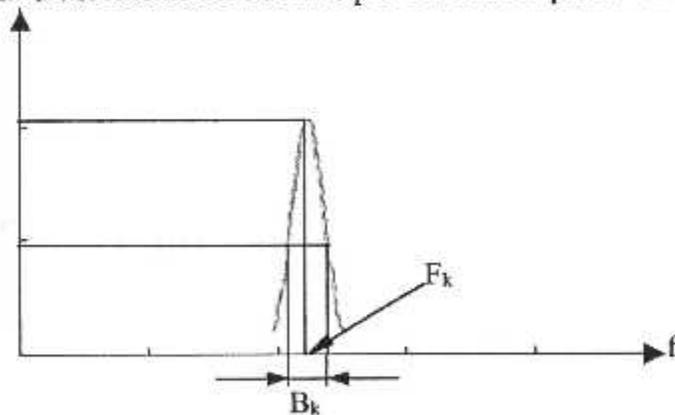
$$x(n) = g(n) * h(n)$$

$$x(n) = g(n) * h(n) \longrightarrow \hat{x}(n) = \hat{g}(n) + \hat{h}(n)$$

L'intérêt de la méthode réside dans le fait que $\hat{g}(n)$ et $\hat{h}(n)$ sont facilement séparables par un filtrage temporel et ceci grâce à l'hypothèse simplificatrice sur $g(n)$.

III-3-9-1. Estimation de la trajectoire des formants

On a vu que le conduit vocal présente un certain nombre de pulsations propres très apparentes pour un son voisé. Dans le signal sonore, ces fréquences propres (formants) sont notées F_k ($k=1,2,\dots$), et les amortissements sont définis par des bandes passantes à 3dB. (figure ci-dessous).



Représentation d'un formant

Pour un conduit vocal dont la longueur est de l'ordre de 17 cm, on peut observer 3 ou 4 formants entre 100 et 3000 Hz, et 4 ou 5 formants entre 100 et 5000 Hz. Les trois premiers formants sont essentiels pour caractériser le spectre vocal ; les deux suivants sont utiles pour une synthèse de qualité.

La détermination précise des formants F_k , exige un principe de déconvolution du signal afin d'atténuer l'effet de l'excitation, toutefois le volume de calcul impliqué par la méthode du cepstre

est important et l'on utilise beaucoup plus couramment l'analyse par LPC : Les paramètres sont dans ce cas estimés à partir du modèle AR.

La procédure générale est illustrée par le diagramme de la figure III-21

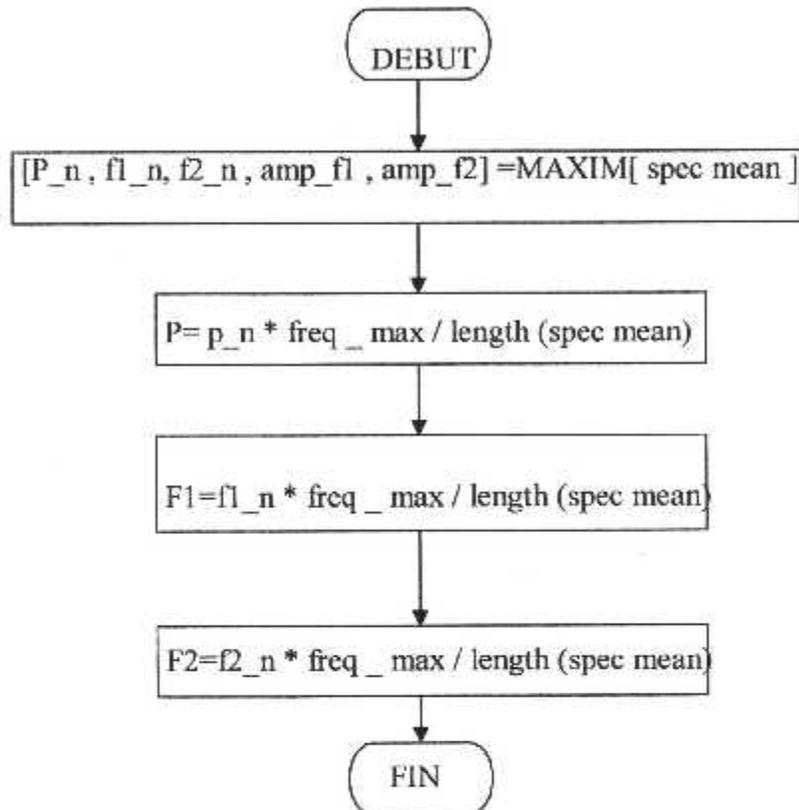


Fig III-21 Organigramme de la procédure « Reconnaissance globale » :

- La fonction extraire

Les paramètres extraits du signal. Cette approche permet une interprétation phonétique directe de la séquence à décoder. Il s'agit donc d'une reconnaissance effectuée sous les contraintes suivantes :

- modèles adaptés au locuteur (longueur du tube, symétrie),
- modèles contextuels (dépendants des deux voyelles V1 et V2),
- prise en compte d'informations dynamiques sur la transition V1-V2.

L'organigramme de cette procédure est représenté par la figure III-22

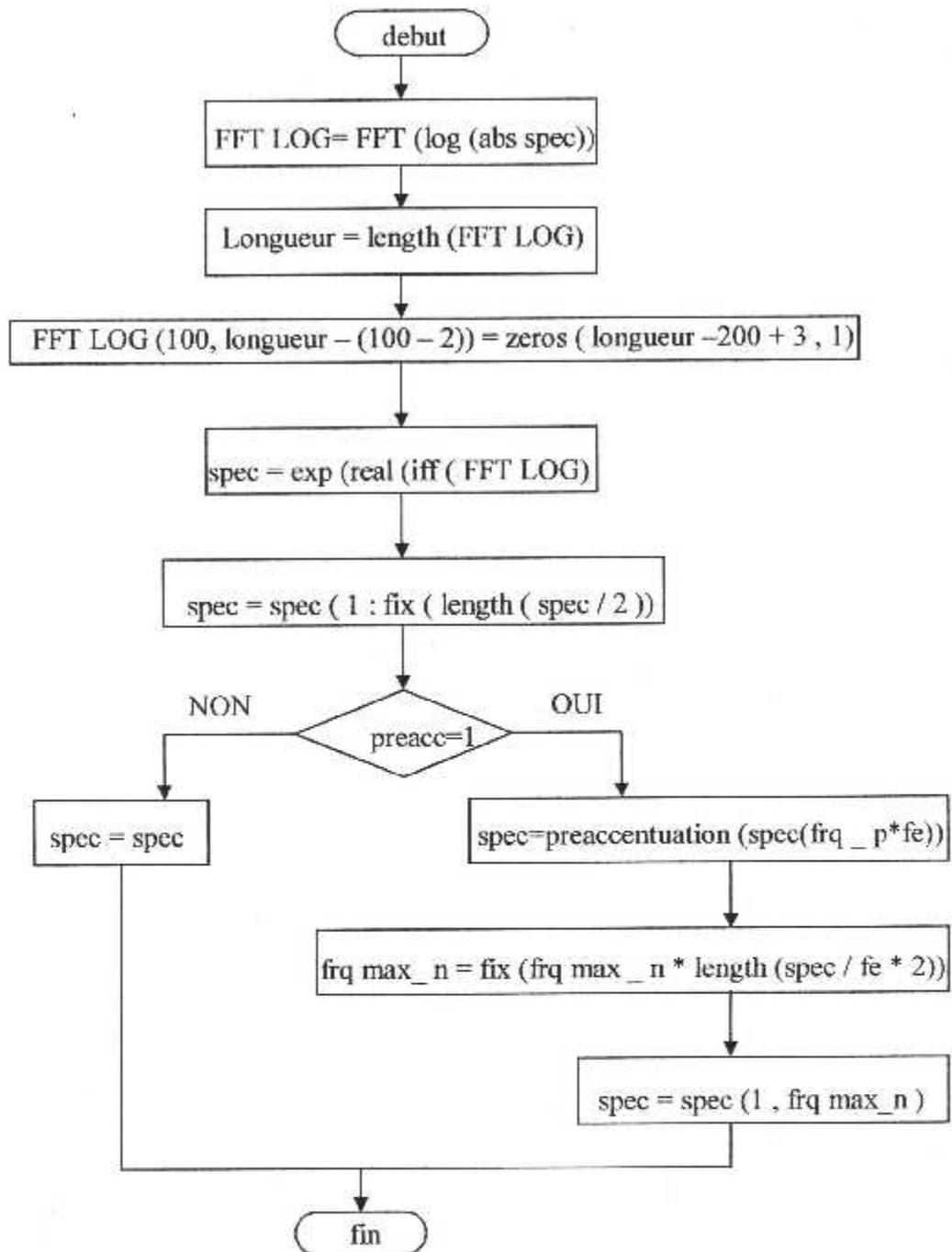


Fig III-22 Organigramme de la procédure « Extraire la fonction » :

- Méthodes d'extraction de la fréquence fondamentale :

La fréquence fondamentale correspond à la fréquence de vibration des cordes vocales. Les algorithmes d'extraction du F_0 utilisent généralement une représentation temporelle ou spectrale du signal. Les méthodes temporelles utilisent la similarité du signal d'une période à l'autre pour identifier la période fondamentale. Il est parfois possible de repérer manuellement la période fondamentale T_0 (telle que $F_0=1/T_0$) directement sur le signal. Dans le domaine fréquentiel, les algorithmes utilisent généralement les harmoniques de la fréquence fondamentale pour trouver le F_0 . Cette propriété du signal peut être visualisée sur une FFT à bande étroite.

L'organigramme de cette procédure est représenté par la figure III-23

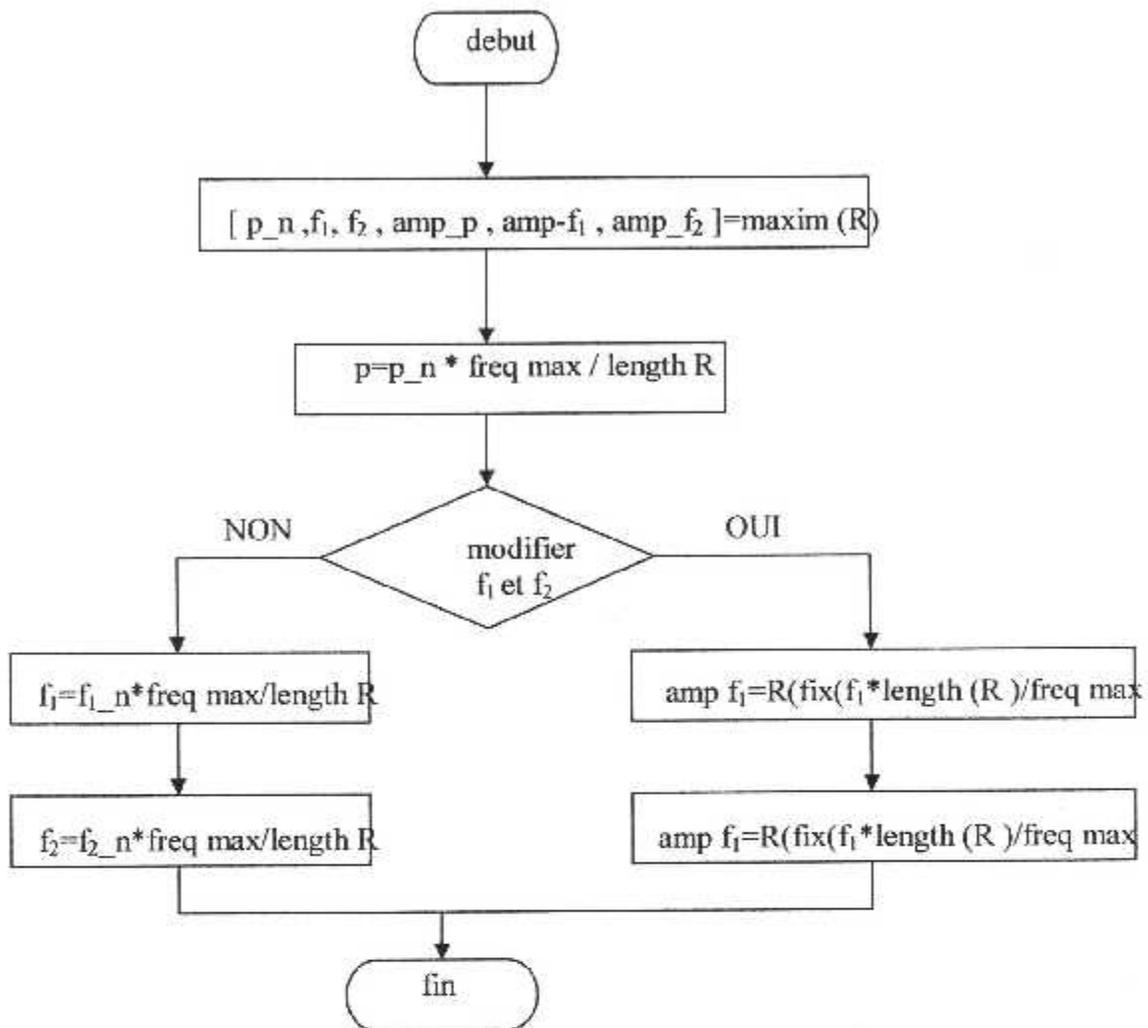


Fig III-23 Organigramme de la procédure « Détermination du formant et du pitch » :

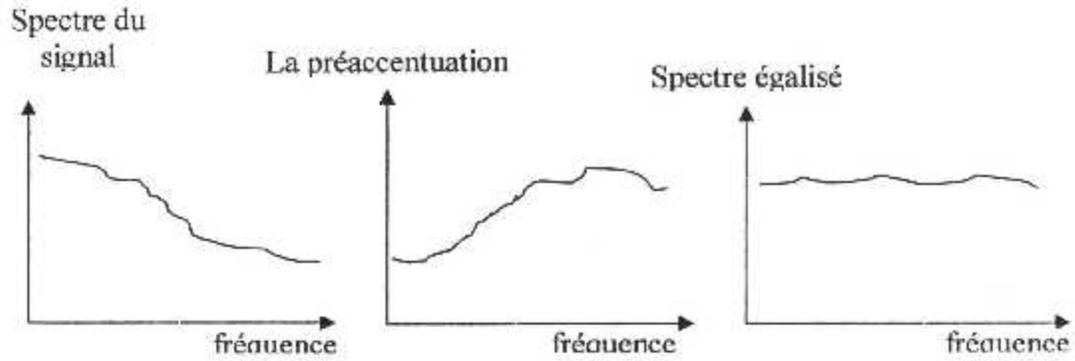
III-3-9-2 Procédure de préaccentuation du signal

En général, le signal vocal se caractérise par une perte de 6 dB/Octave, due à l'influence de la source d'excitation et au rayonnement des lèvres. Une perte de 6 dB/Octave veut dire que les hautes fréquences ont une énergie plus faible que celle des basses fréquences. Pour pallier à cet inconvénient la préaccentuation permet d'égaliser les sons aigus avec les sons graves (figure ci-dessous).

L'opération consiste à faire passer le signal à travers un filtre de transmittance :

$$H(Z) = 1 - \mu Z^{-1}$$

Le facteur de préaccentuation est pris entre 0.9 et 1 (souvent 0.95). Comme conséquence, la préaccentuation introduit une légère distorsion spectrale.



$$X(n) \rightarrow \boxed{H(Z)=1-\mu \times Z^{-1}} \rightarrow Y(n)=X(n)-X(n-1)$$

L'organigramme de cette procédure est représenté par la figure III-24

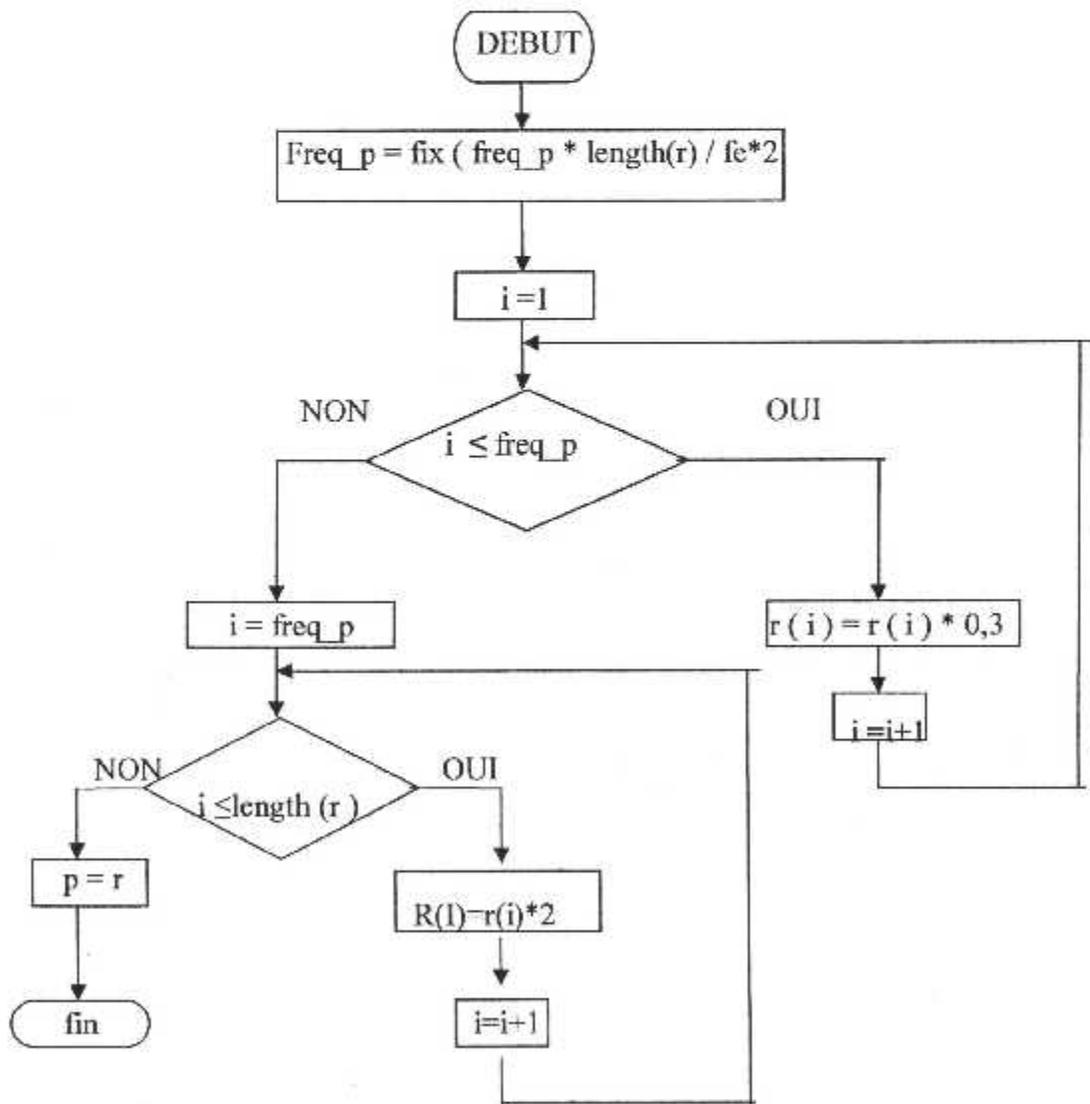


Fig III-24 Organigramme de la procédure « préaccentuation » :

III-3-10 Procédure spectrogramme du signal

Un sonagramme est une représentation de l'évolution du spectre fréquentiel d'un son en fonction du temps. Le sonagramme permet de voir plusieurs aspects du son, c'est-à-dire son *spectre*:

- Il permet de distinguer chaque partiel (ou harmonique): lignes horizontales de bas en haut
- Il différencie leur intensité par des couleurs
- Il suit leur évolution dans le temps de gauche à droite

Le sonagramme et le spectrogramme jouent le même rôle. La plupart des spectrogrammes sont maintenant produits à l'aide d'un ordinateur équipé d'une carte de numérisation et d'un programme d'analyse et de représentation graphique du signal sonore. Un spectrogramme est une image graphique représentant les propriétés acoustiques (durée, fréquence, amplitude, timbre) des unités phoniques sous-jacentes aux langues. Un phonéticien averti peut y reconnaître les traits acoustiques (bruit ou son, périodique ou apériodique, impulsionnel ou continu) associés à chaque son et, dans le cas des voyelles, les formants (e.g. longues bandes noires horizontales sur le spectre) qui assurent leur discrimination auditive.

Les figures III-25 et III-26 représentent respectivement les organigrammes des procédures « spectrogramme » et tracé du sonagramme.

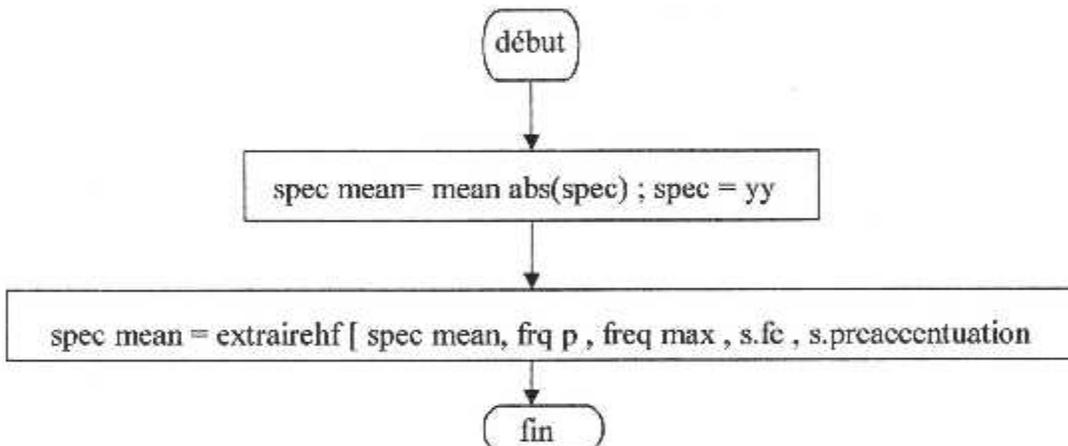


Fig III.25 Organigramme de la procédure « spectrogramme » :

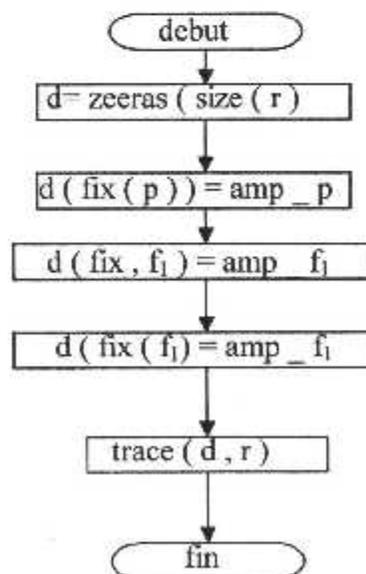


Fig III.26 Organigramme de la procédure « tracé du sonagramme »

Chapitre IV

Mode d'emploi de l'éditeur de signal

Références [2], [9]

IV- Mode D'Emploi de l'Editeur du Signal

IV.I - Introduction

Compte tenu de la multiplicité des menus figurant dans ce logiciel, il s'est avéré nécessaire d'introduire un chapitre traitant du mode d'emploi de celui-ci.

Le présent logiciel comporte deux modules :

- Le synthétiseur des signaux :

A travers ce programme, on génère des signaux dont le type, la fréquence, la durée, l'amplitude, la phase, le pas, le retard, la limite, la longueur sont déterminés par l'utilisateur.

- L'éditeur de signal :

Ce programme permet la visualisation et le traitement des signaux générés par le synthétiseur.

IV.2 - Mode d'emploi du synthétiseur des signaux

Démarrez votre ordinateur de la manière habituelle ensuite lancez le langage de programmation MATLAB. A l'aide de la souris, cliquez sur « Path Browser » dans le menu ;

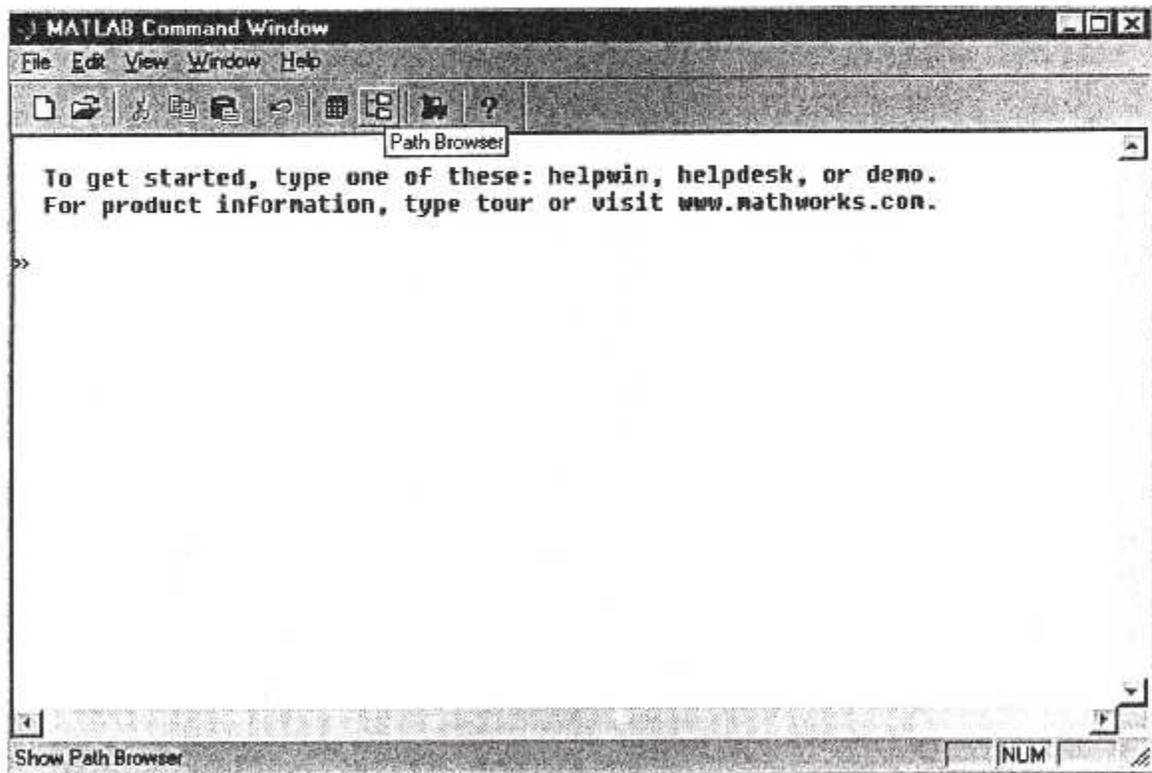


fig IV-1 Environnement matlab

Une fenêtre apparaîtra. Dans cette fenêtre, cliquez sur « Browse ».

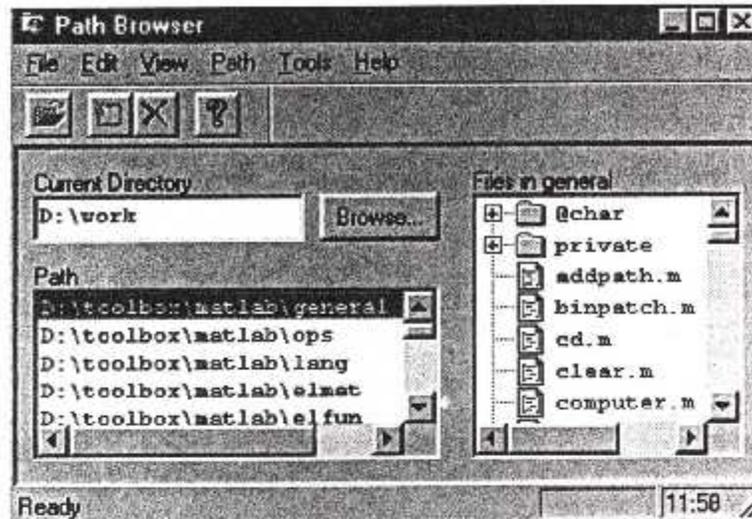


fig IV-2 Espace d'ouverture du contenu de l'ordinateur

Ce dernier fait apparaître le contenu de votre ordinateur. Toujours à l'aide de la souris, ouvrez le répertoire qui contient le programme de nom « salah » sélectionnez-le et cliquez sur « OK » ; ensuite fermez « Browse ».

Dans notre cas, le répertoire choisi est le « work ».



fig IV-3 Contenu de l'ordinateur

Revenez maintenant sur la première fenêtre et tapez « signal » et faire « OK ». Vous voyez apparaître une interface qui montre en haut les différents boutons des signaux à analyser.

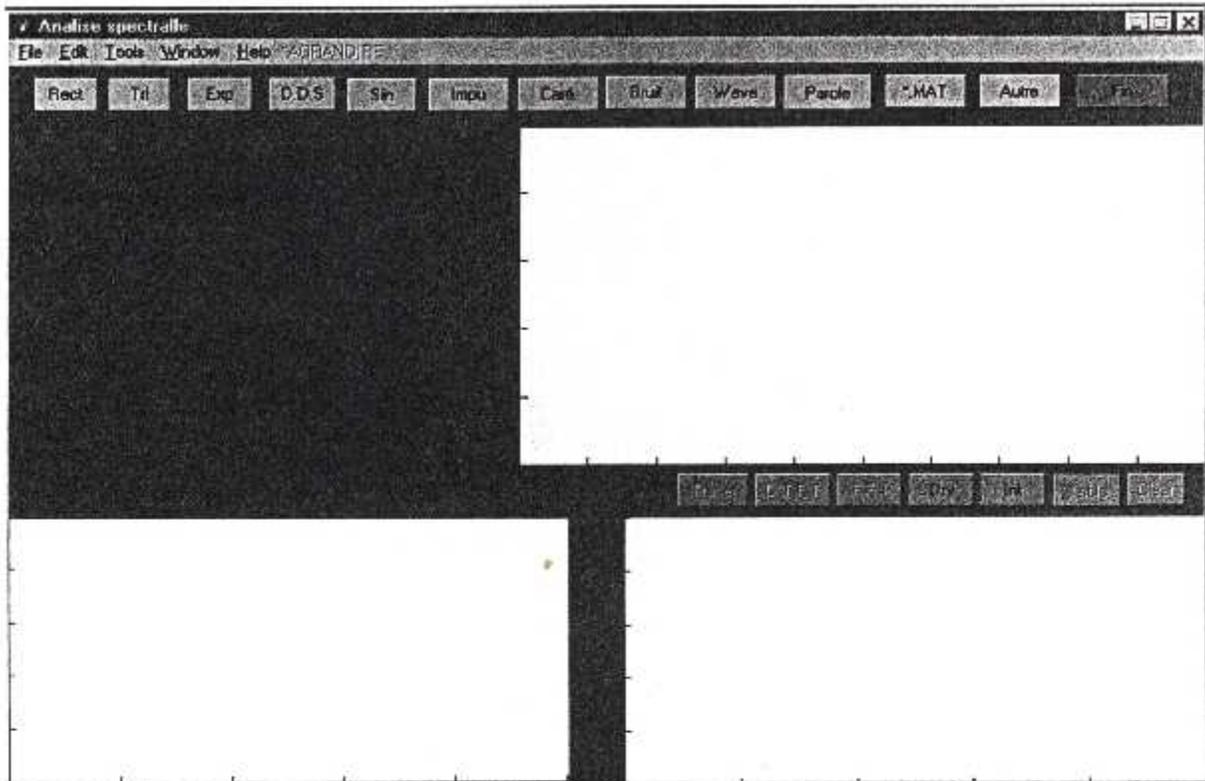


fig IV-4 Interface du synthétiseur de signaux

Il suffit de cliquer sur un bouton pour voir apparaître à gauche les paramètres par défaut du signal généré, paramètres que vous pouvez modifier et tracer le signal.

Par exemple si vous cliquez sur le bouton Sin, le signal généré est celui d'une somme de sinusoides et vous voyez apparaître à gauche les paramètres par défaut de ce signal.

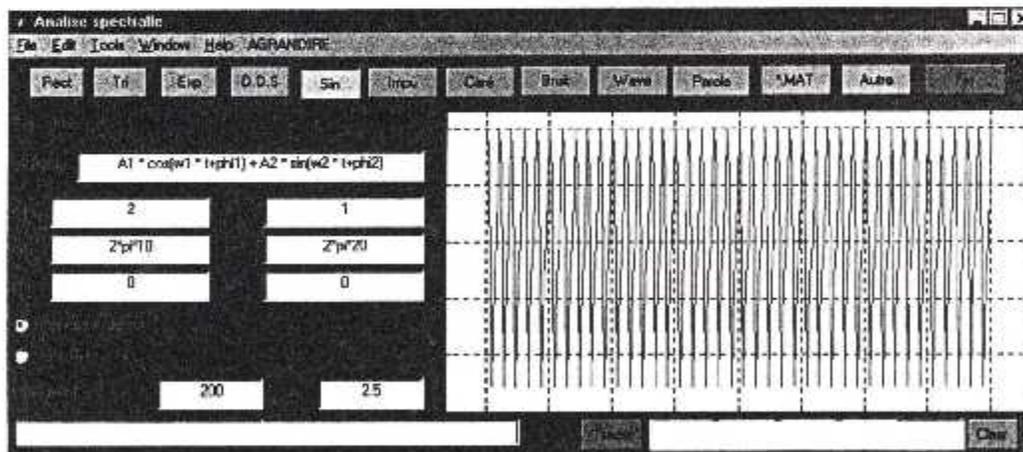


fig IV-5 Graphe d'une somme de sinusoides

Pour modifier, il suffit de cliquer dans le cadran du paramètre que vous voulez changer pour voir apparaître le curseur. A l'aide du clavier, vous pouvez effacer le paramètre par défaut et mettre un paramètre personnalisé.

Pour choisir un autre signal, vous pouvez cliquer directement sur son bouton. Pour tout effacer, cliquez sur « Clear » et pour quitter le synthétiseur, cliquez sur « Fin ».

IV.3 – Mode d'emploi de l'éditeur de signal

Avant tout, se munir d'un fichier de parole de nom [*.Wave]. Pour cela, cliquez sur le bouton « Wave » ensuite sur « Browse ».

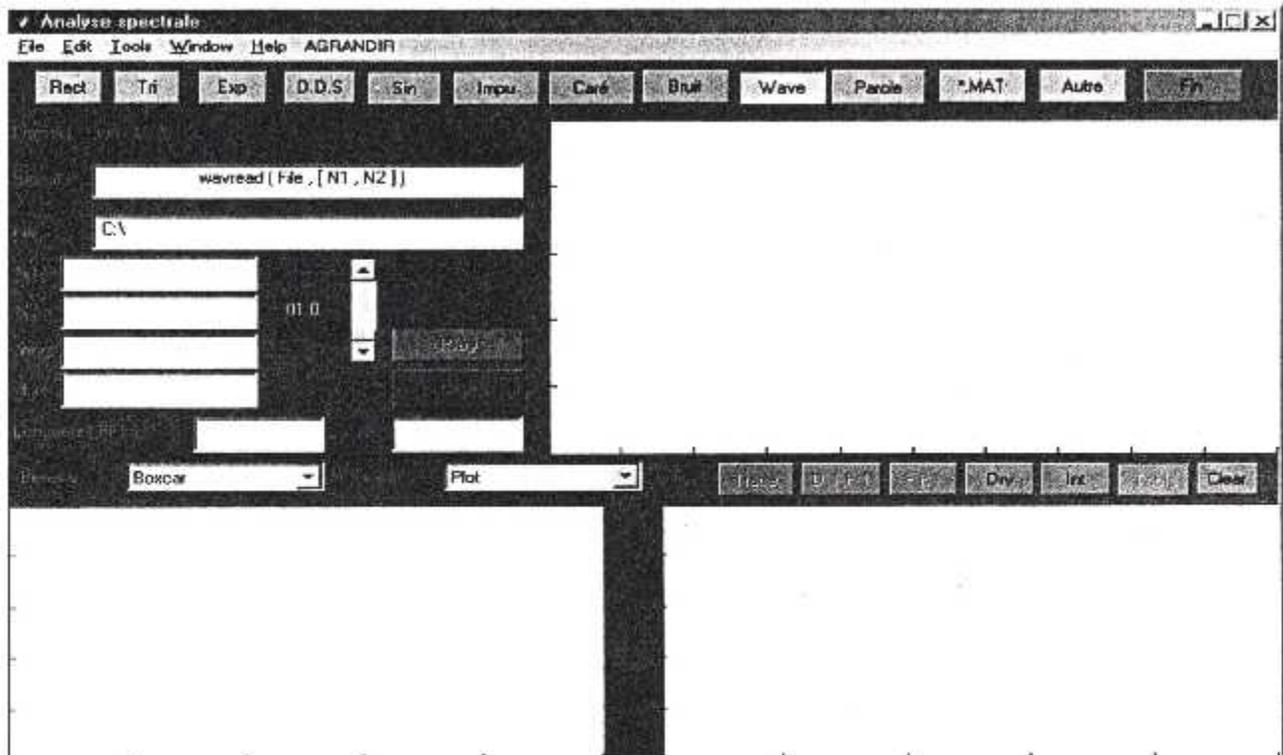


fig IV-6 Interface de l'éditeur de signal

Choisissez un fichier de parole dans votre ordinateur ou bien choisissez un dans une liste que nous avons proposée et qui se trouve dans le répertoire « Salah » et cliquez sur « Ouvrir ». Cette voyelle orale est ainsi sélectionnée pour être analysée.

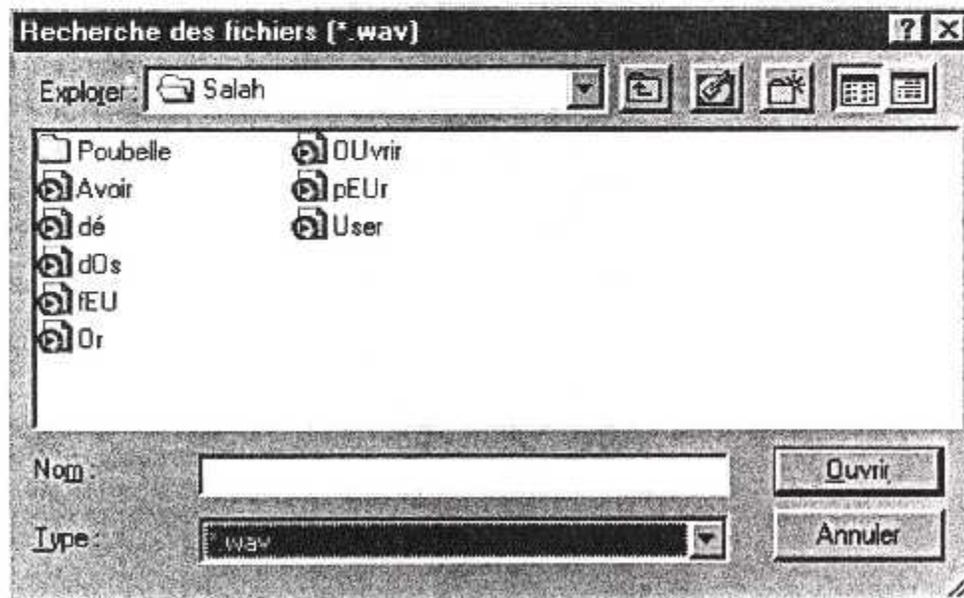


fig IV-7 Liste des signaux de parole

- **Visualisation du signal**

Par exemple choisissons la voyelle orale «Or » et cliquons sur « tracer ». nous visualisons un signal de fréquence d'échantillonnage, de durée, de longueur inscrites à gauche de l'interface. Mais il faut que N2 soit inférieur ou égal à la longueur du fichier pour avoir un bon tracé.

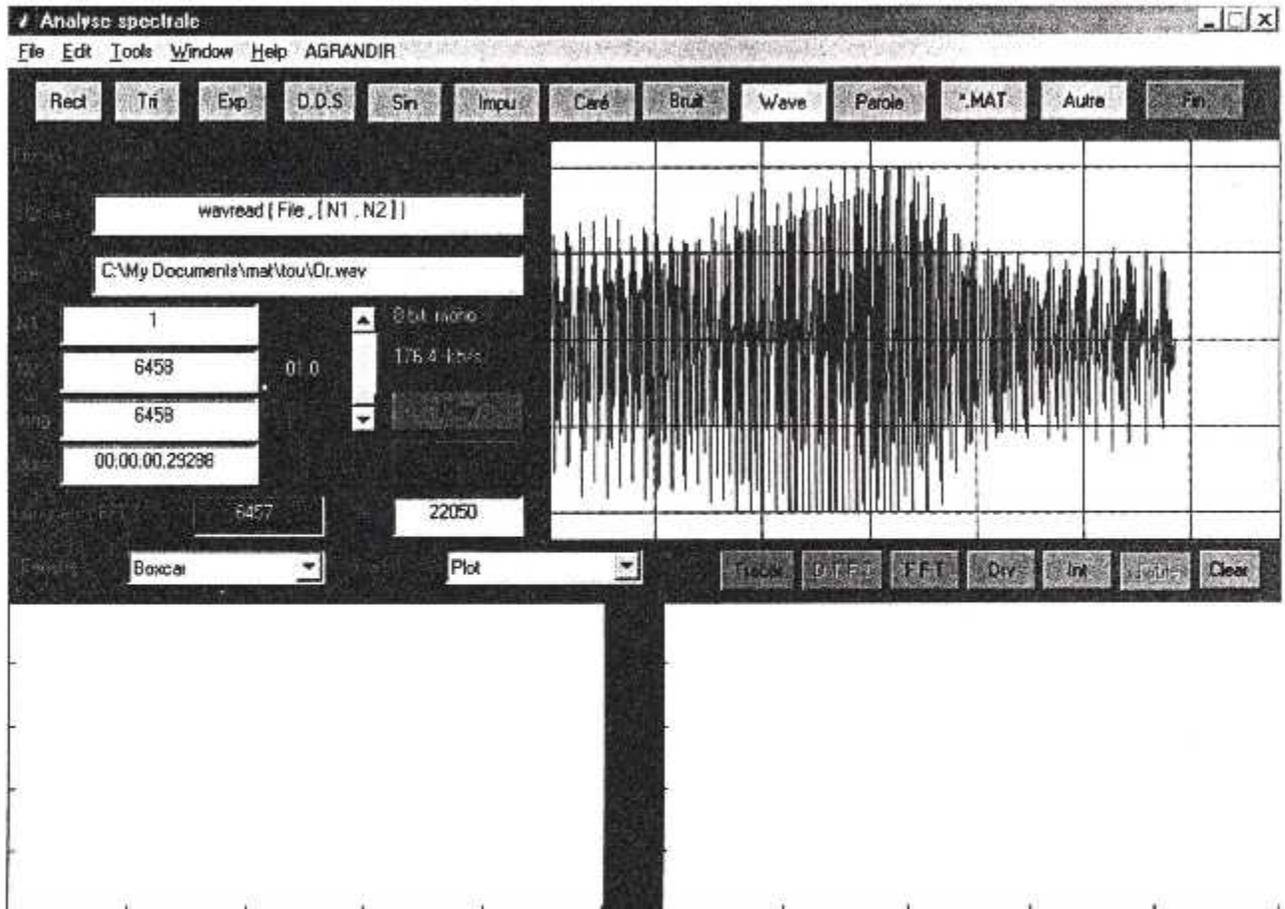


fig IV-8 Graphe d'un signal de parole de nom « or »

- **Visualisation des spectres d'amplitude et de phase**

Nous pouvons tracer les spectres d'amplitude et de phase (par la FFT) en cliquant sur le bouton « FFT ». Pour cela, nous pouvons choisir la fenêtre et le type de tracer (plot, stem, barre).

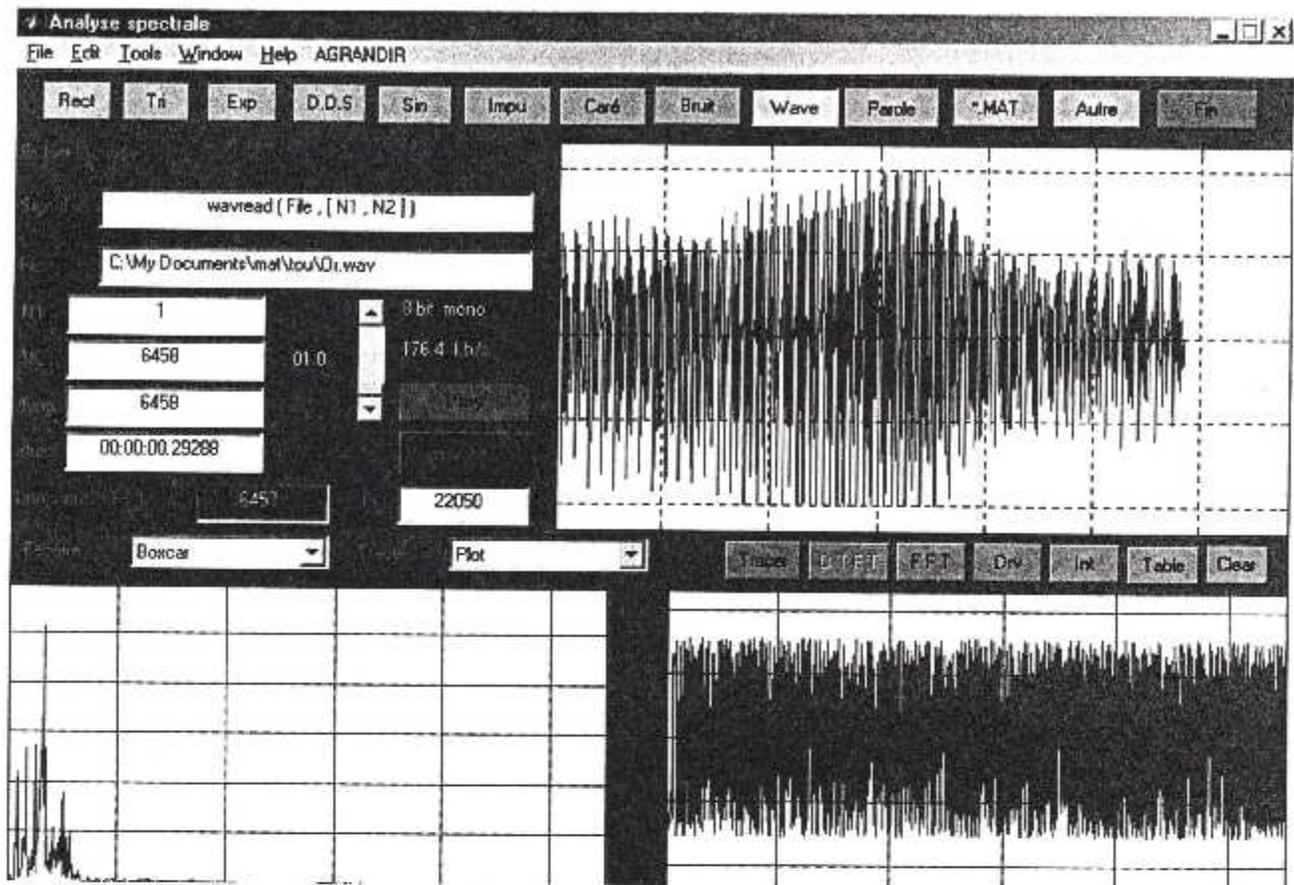


fig IV-9 Signal et spectres d'amplitude et de phase de la voyelle orale O comme « or »

Nous pouvons agrandir les différents tracés en cliquant sur « agrandir » dans le menu.

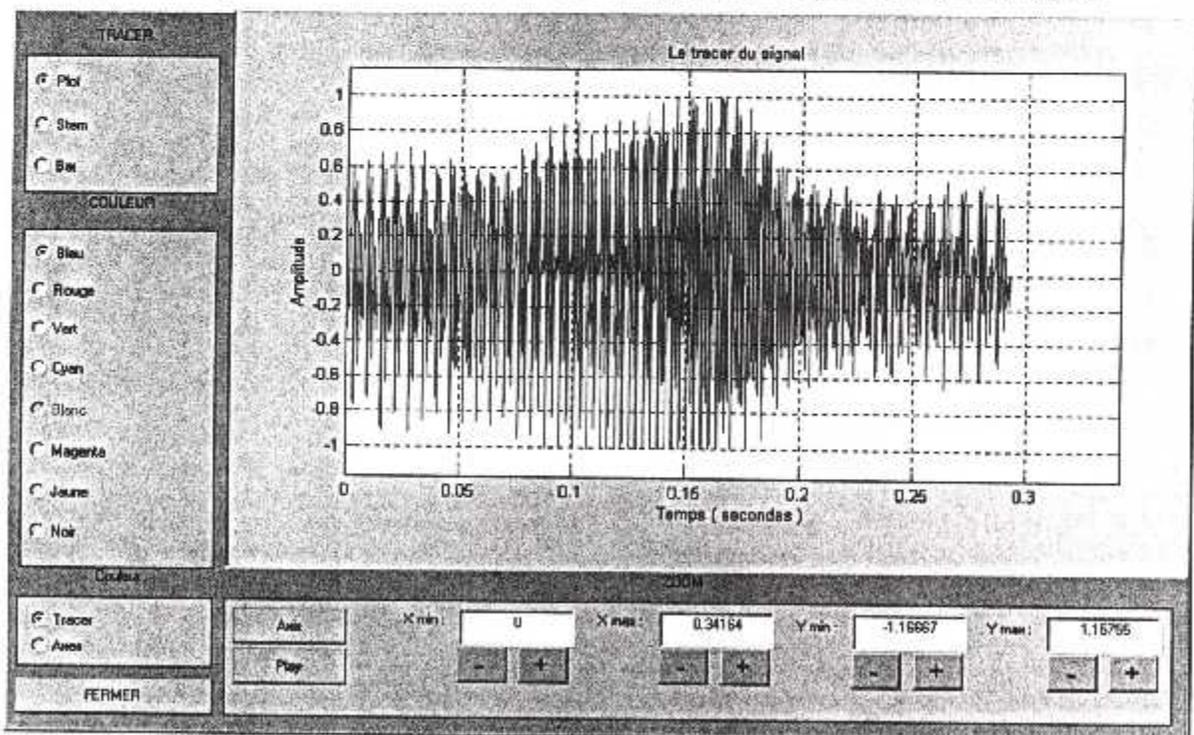


fig IV-10 Agrandissement du signal O comme « or »

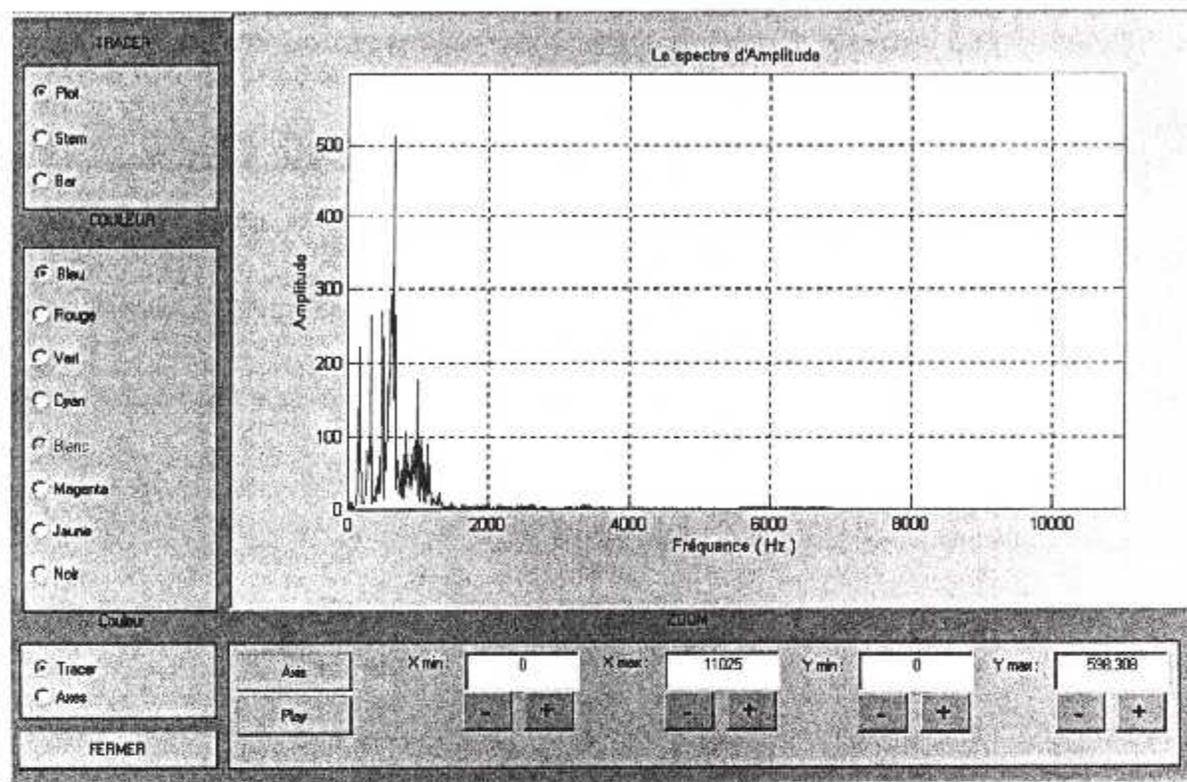


fig IV-11 Agrandissement du spectre d'amplitude du signal « or »

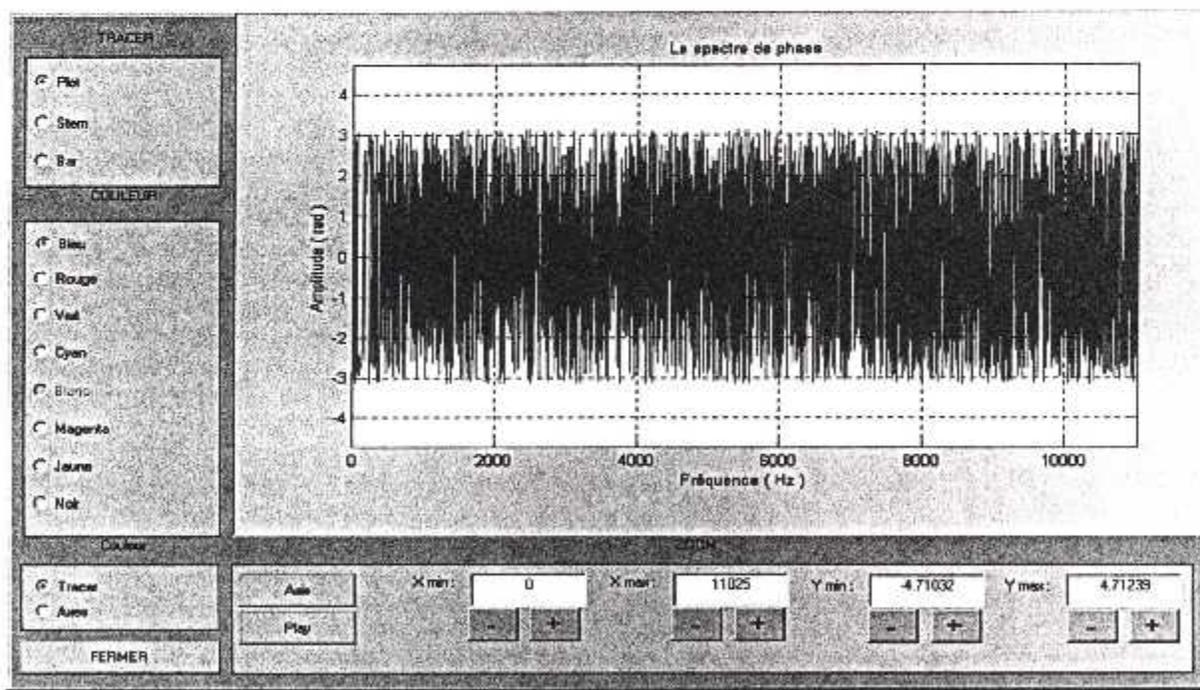


fig IV-12 Agrandissement du spectre de phase du signal « or »

• Zoom sur le signal

Il permet de visualiser le signal « à la loupe ». le signal peut s'étirer ou se rétrécir tant horizontalement que verticalement par simple clic sur un « + » ou un « - » .

A l'aide du zoom, vous pouvez délimiter la portion du signal que vous désirez traiter. Il suffit de définir votre intervalle tant pour les X que pour les Y et de cliquer sur « axis » .

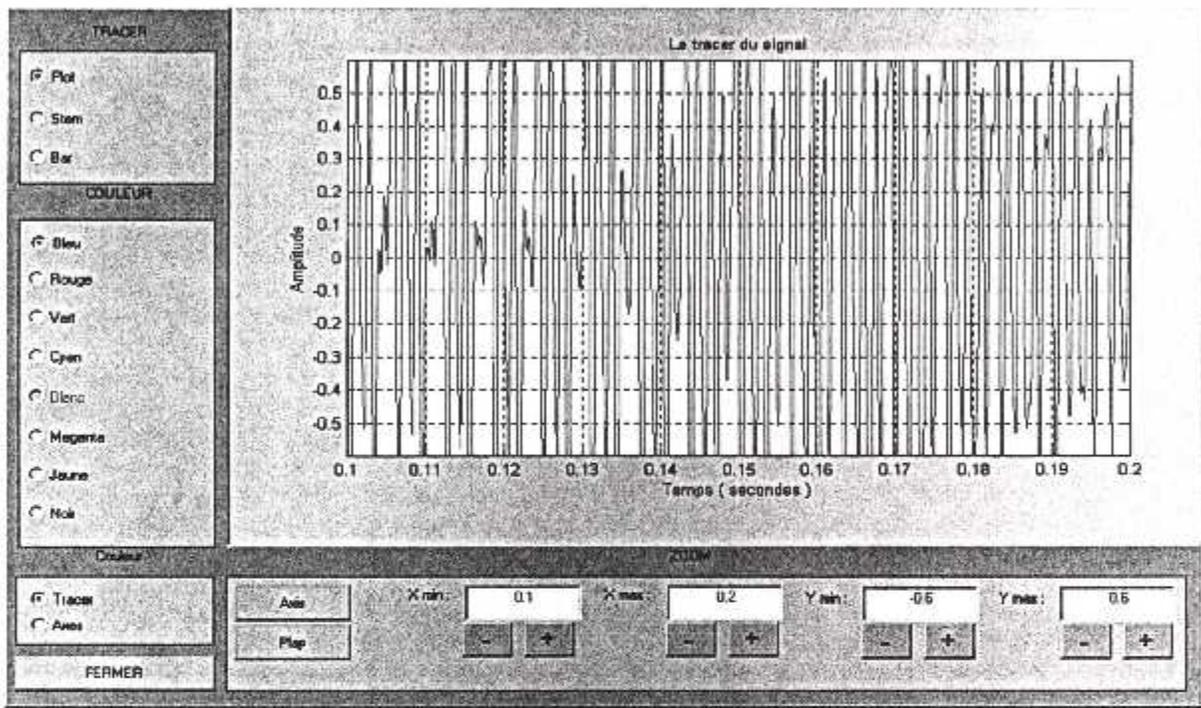


fig IV-13 Zoom du signal « or »

Dans notre exemple, nous avons zoomé le signal entre 0.1 et 0.2 pour les X et entre -0.6 et 0.6 pour les Y.

• Ecoute du signal

Cliquez sur « Play » pour écouter le signal. Vous pouvez sélectionner une portion du signal en le zoomant et écouter uniquement cette portion.

Pour quitter l'agrandissement, cliquez sur « fermer ».

Dans l'agrandissement, vous pouvez changer les couleurs des axes et du tracé en choisissant l'axe ou le tracé et la couleur de votre choix.

• Analyse du signal

Pour cela, cliquez sur le bouton « Parole » dans l'interface, une autre interface apparaît. Dans cette dernière, choisissez le signal que vous voulez analyser dans la liste des signaux et cliquez sur « charger ». vous voyez les différents composants de ce signal.

Prenons toujours l'exemple sur la voyelle orale « Or ».

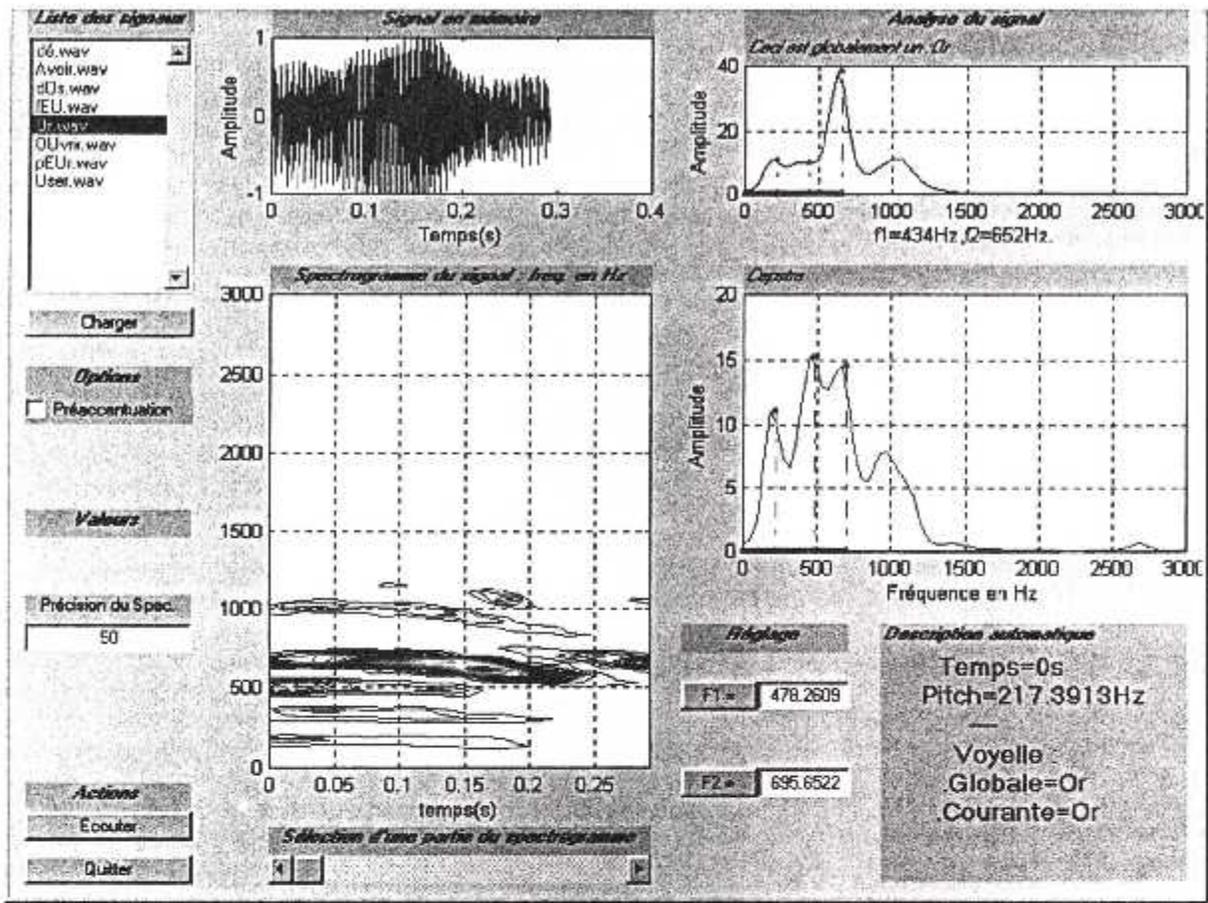


fig IV-14 Interface de l'analyse du spectre de la parole

Dans cet exemple, nous voyons les fréquences des formants ($F1 = 434\text{Hz}$ et $F2=652\text{Hz}$).

- **Le cepstre**

Avec la courbe du cepstre, vous pouvez détecter le pitch c'est-à-dire la fréquence du fondamental $F=217.391\text{Hz}$.

- **Le sonagramme**

Il représente la durée, la fréquence et le spectrogramme (c'est-à-dire l'intensité) de la voix. Dans notre exemple, la durée est de 0.28s, les fréquences $F1=478.2609\text{Hz}$, $F2=695.6522\text{Hz}$. L'intensité est représentée par le spectrogramme dont on peut accentuer la précision en augmentant la valeur dans « Précision du Spec ». Pour quitter, cliquez sur « quitter ».

Chapitre V

Exemples d'application

Références [2], [5], [6]

V. Exemples D'application

Nous avons utilisé des fenêtres de 64 points et un calcul de FFT sur 4096 points. Les signaux utilisés sont de fréquences $F=20\text{Hz}$, d'amplitudes 50V pour la fenêtre de Boxcar, 25V pour la fenêtre de Bartlett et 23V pour celles de Blackman et de Hamming. Ces signaux sont échantillonnés à $F_e=2.5\text{Hz}$

V.1.Comparaison des différentes fenêtres de pondération

Nous avons utilisé huit fenêtres de pondération dans ce logiciel. Nous allons prendre l'exemple sur certaines d'entre elles pour en faire des comparaisons.

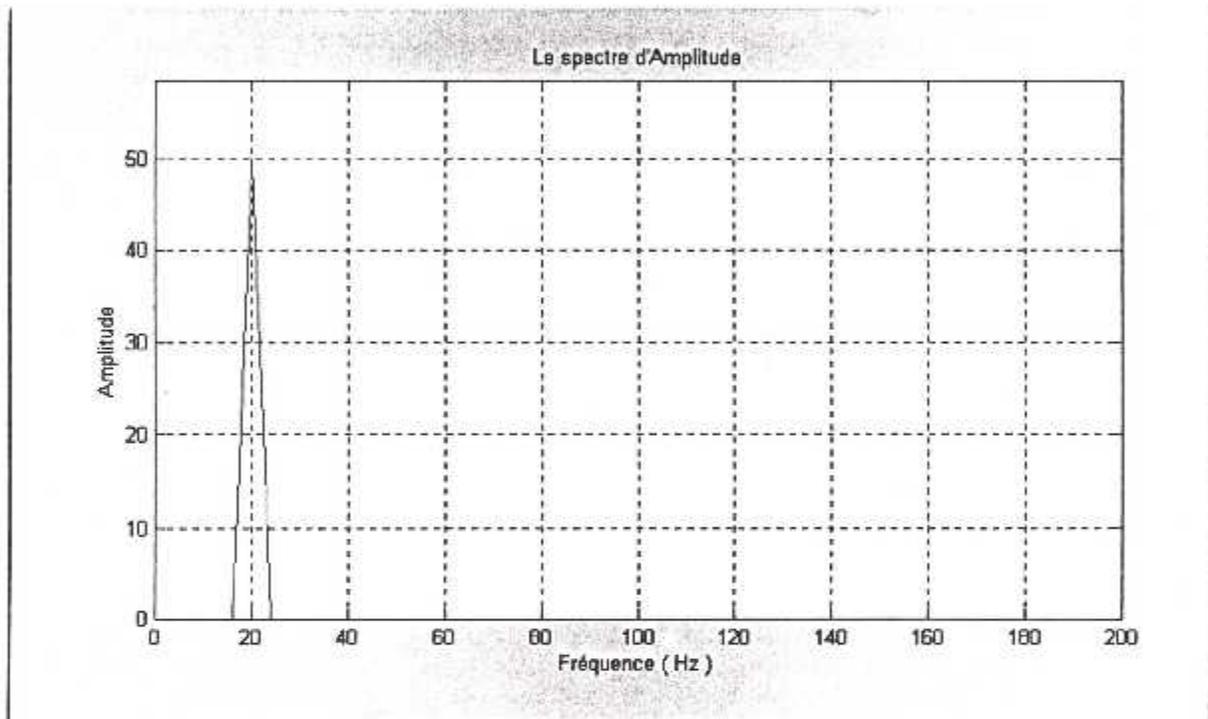


Fig V-1 Visualisation du spectre d'amplitude du signal sinusoïdal
Fenêtre de Boxcar, type : plot

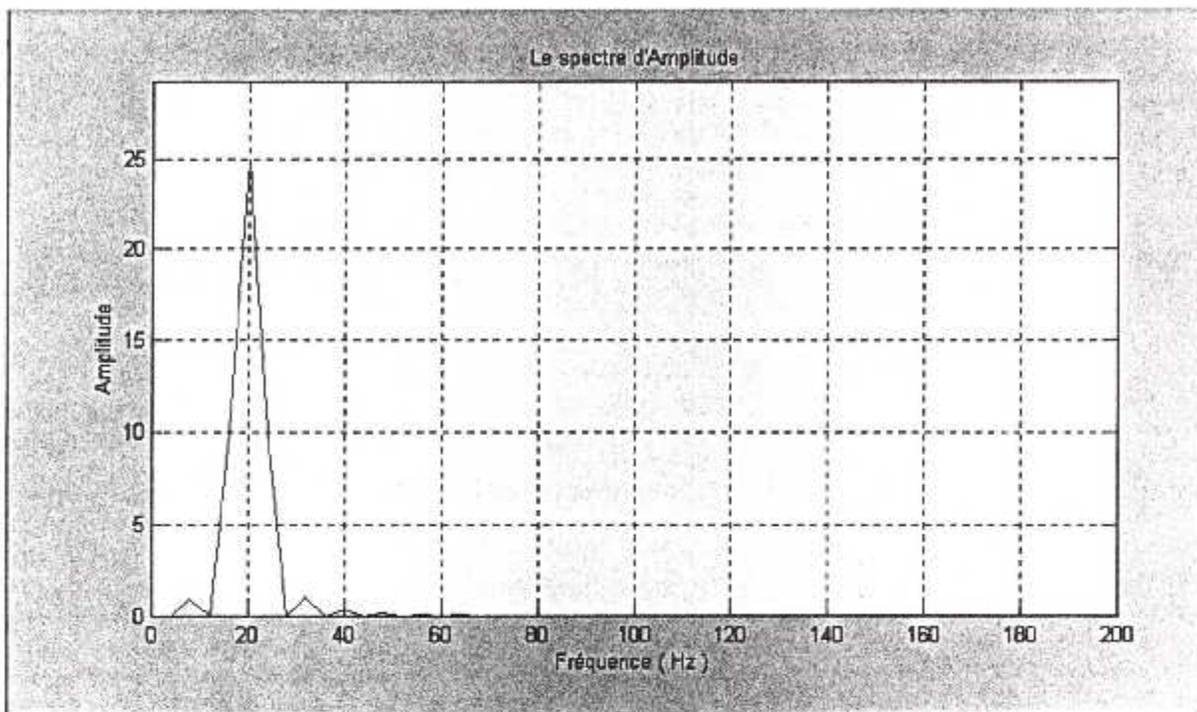


Fig V-2 Visualisation du spectre d'amplitude du signal sinusoïdal Fenêtre de Bartlett, type : plot

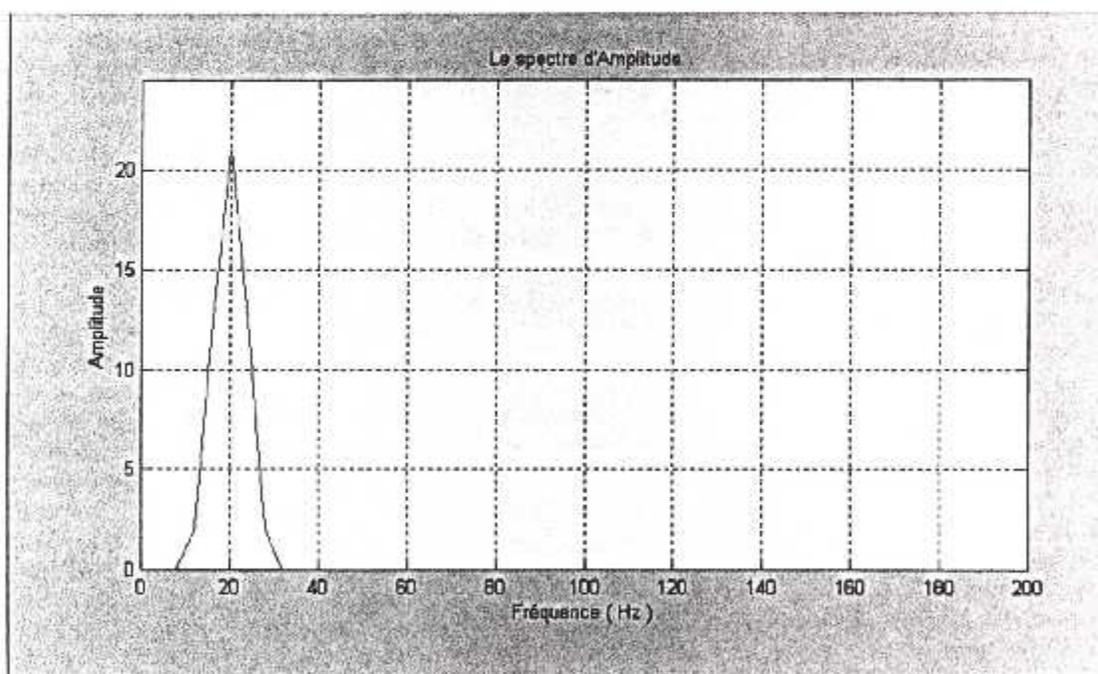


Fig V-3 Visualisation du spectre d'amplitude du signal sinusoïdal Fenêtre de Blackman, type : plot

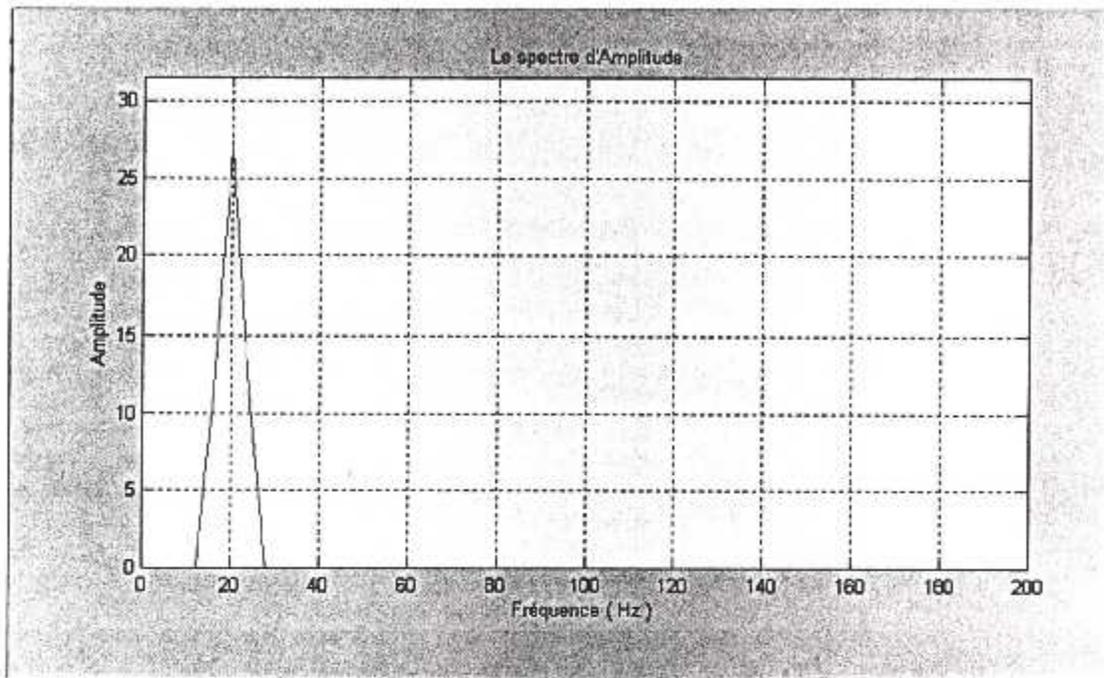


Fig V-4 Visualisation du spectre d'amplitude du signal sinusoïdal
Fenêtre de Hamming, type : plot

En essayant de comparer ces différentes fenêtres, nous constatons que :

- Le lobe principal de la fenêtre de Boxcar est rétréci par rapport aux autres ;
- Les lobes secondaires sont beaucoup plus accentués au niveau de la fenêtre de Bartlett ce qui ne donne pas une grande précision ;
- Au niveau de la fenêtre de Blackman le premier lobe est plus large mais l'atténuation est encore plus élevée. Cette fenêtre est une extension de l'idée de Hamming ;
- La fenêtre de Hamming a un pic central plus large mais une atténuation des oscillations sensiblement plus importante.

Parmi ces fenêtres, celle de Boxcar nous donne une grande précision.

V. 2 Variation de la fréquence des signaux

Dans ce logiciel, l'utilisateur peut augmenter la fréquence comme il le souhaite jusqu'à une certaine limite au-delà de laquelle il ne constatera pas de changement significatif au niveau de la courbe.

Par exemple pour le signal carré, à partir de 1500Hz, la courbe ne varie presque plus. En revanche, au fur et à mesure que nous augmentons la fréquence, le son devient de plus en plus aigu jusqu'à s'atténuer vers les 20000Hz.

Quand nous augmentons la fréquence d'échantillonnage, le nombre d'échantillon aussi augmente.

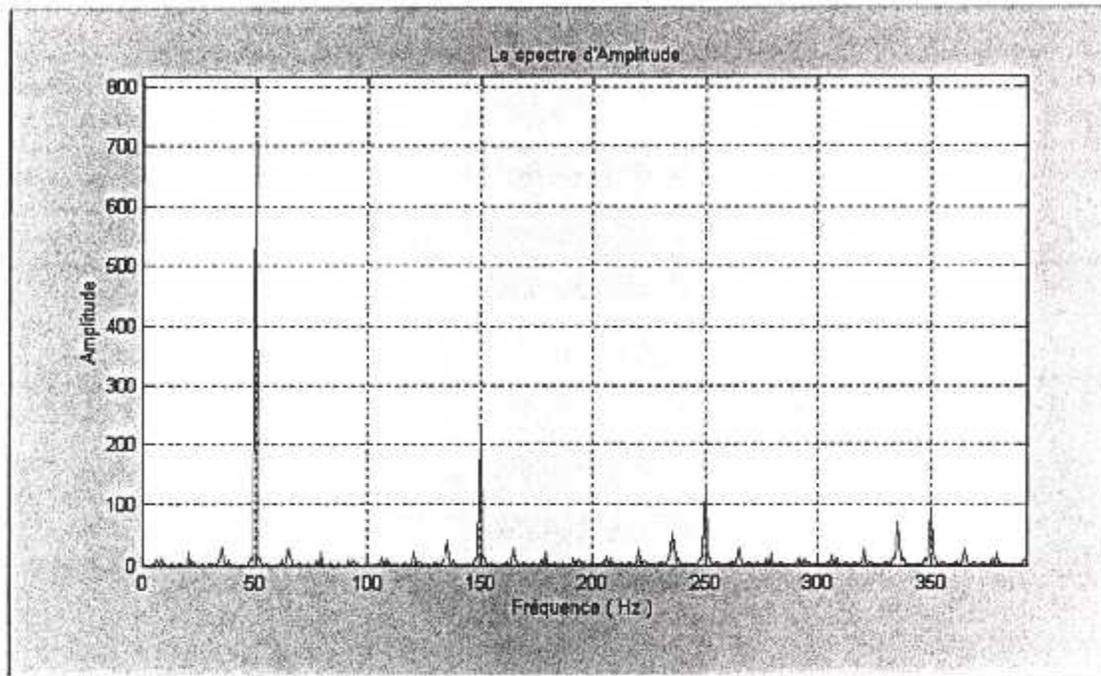


Fig V.5 spectre d'amplitude d'un signal carré de $F_0=50\text{Hz}$, d'amplitude 700V échantillonné à $F_e=2.5\text{Hz}$ fenêtre de Boxcar

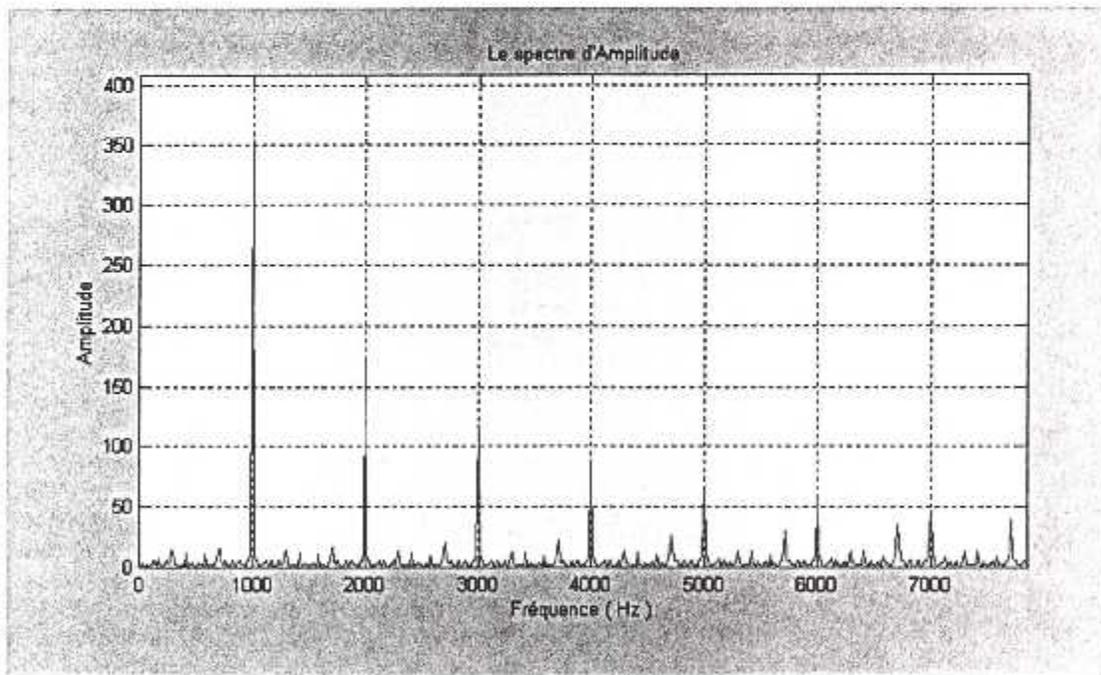


Fig V.6 spectre d'amplitude d'un signal dents de scie de $F_0=1000\text{Hz}$ d'amplitude 350V échantillonné à $F_e=2.5\text{Hz}$ fenêtre de Boxcar

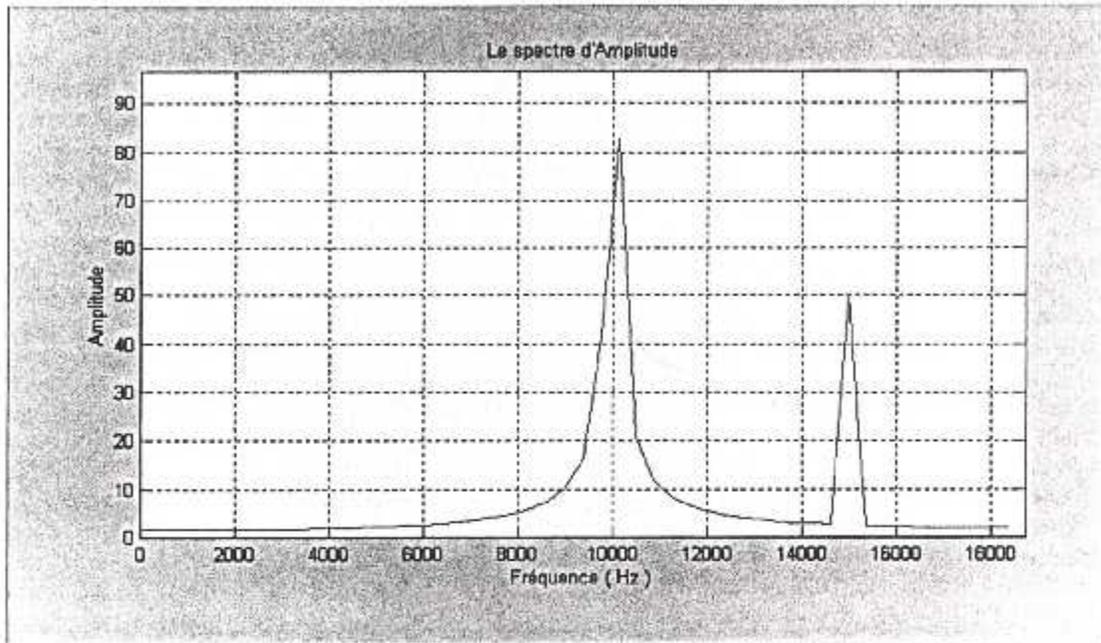


Fig V.7 spectre d'amplitude d'une somme de sinusoïdes de $F_{01}=10$ KHz, $F_{02}=15$ KHz d'amplitudes 80V et 50V échantillonné à $F_e=2.5$ Hz fenêtre de Boxcar

Nous avons choisi différents F_0 (fréquence fondamentale) pour les courbes V.5, V.6 et V.7. Une grande fréquence ne nous donne pas assez d'information sur la courbe. En revanche, elle nous donne beaucoup d'informations concernant la voix. Plus la fréquence est grande, plus le son de la voix est aigu.

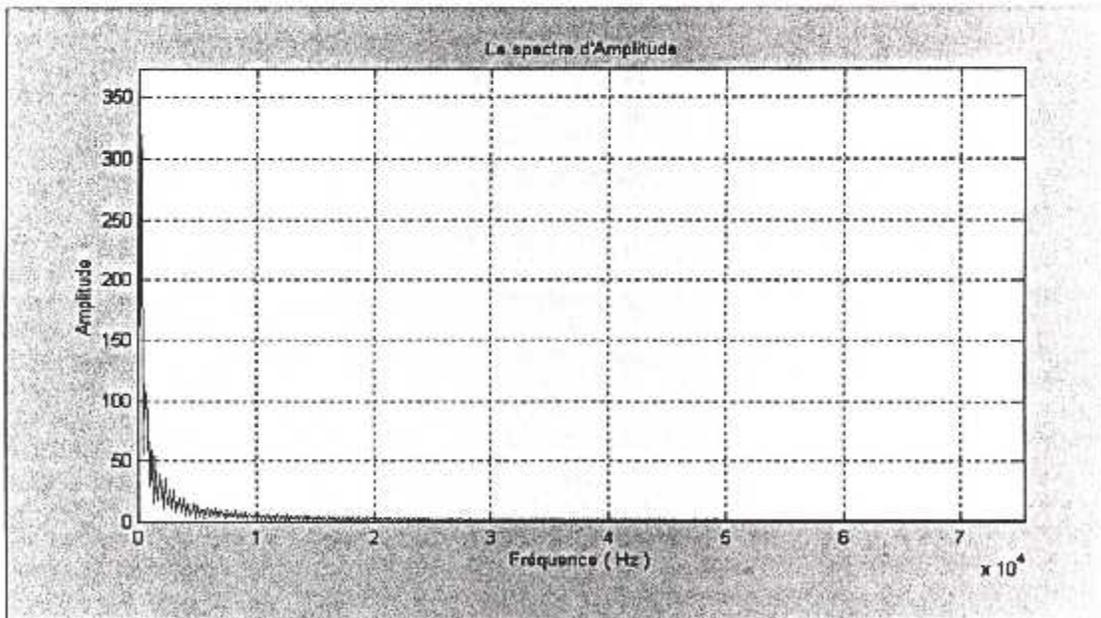


Fig V.8 spectre d'amplitude d'un signal carré de fréquence 70KHz de fréquence d'échantillonnage 120Hz et d'amplitude 325V(fenêtre de Boxcar)

V.3 Zoom de quelques signaux

Le zoom consiste à dilater le signal entre deux valeurs précises. Nous pouvons tracer le spectre du signal zoomé comme le montre la figure V.9. Nous pouvons également écouter cette portion du spectre.

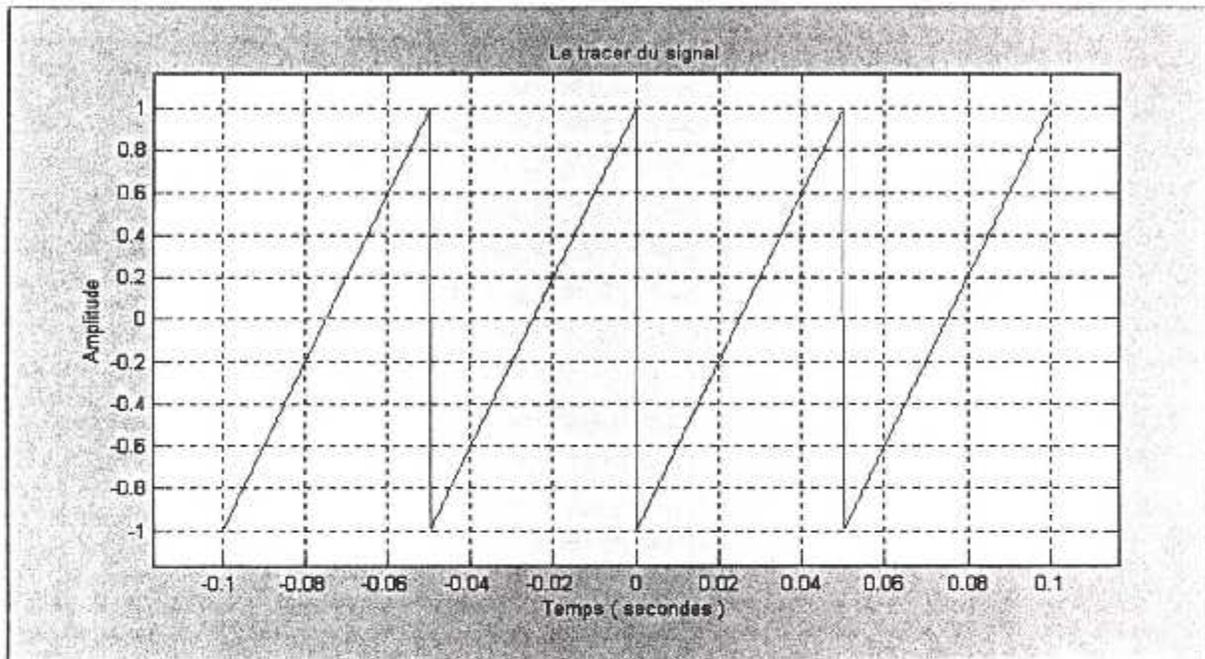


Fig V-9 Visualisation du zoom entre -0.1 et 0.1 du signal Dents de scie

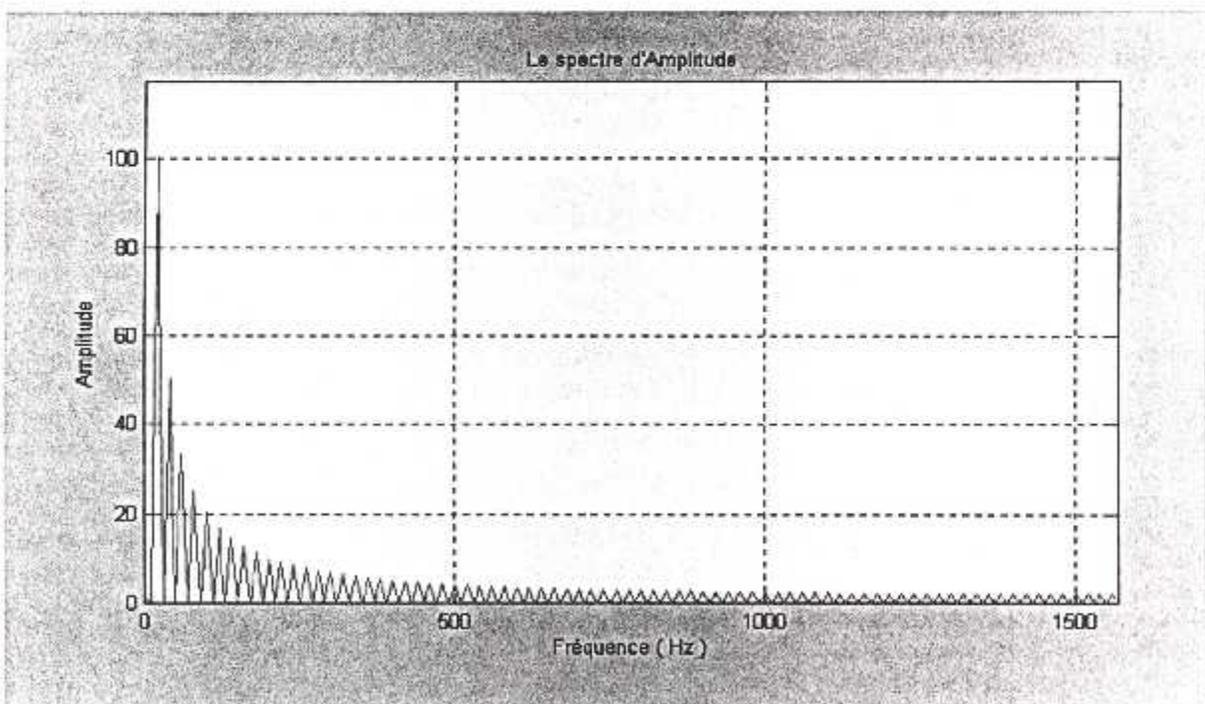


Fig V-10 Visualisation du spectre d'amplitude du signal DDS zoomé
Fenêtre de Hanning, type : plot

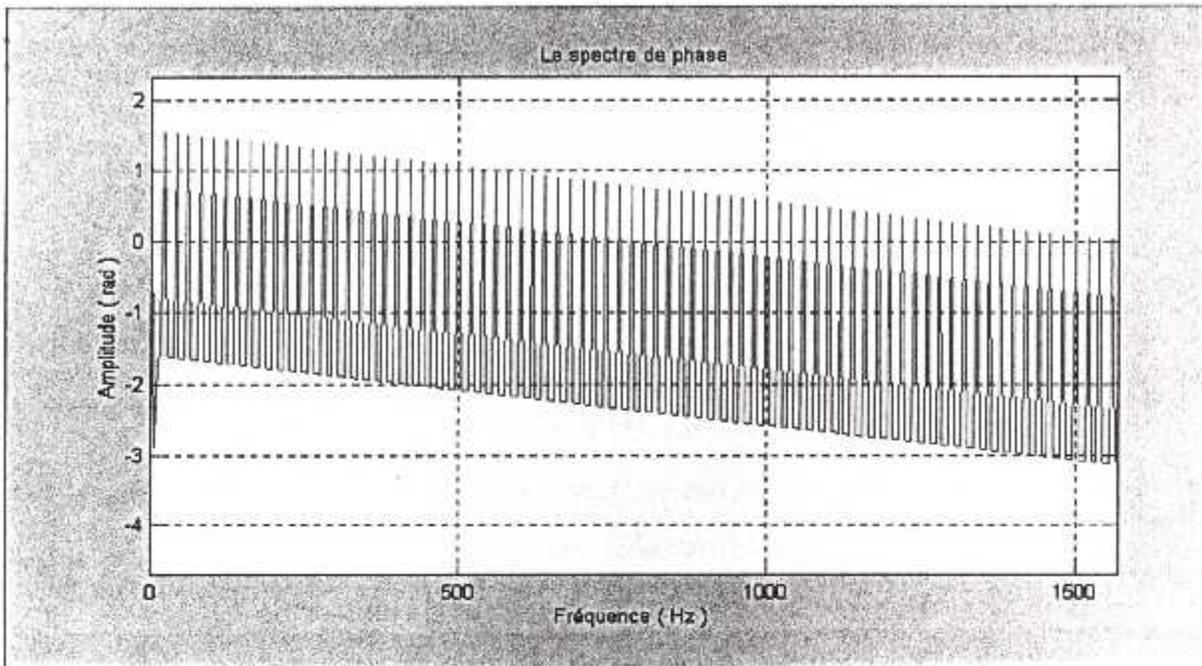


Fig V-11 Visualisation du spectre de phase du signal DDS zoomé
Fenêtre de Hanning, type : plot

V.4 Exemple de visualisation d'un signal de parole

Pour les signaux de parole, nous avons limité la fréquence à 3000Hz pour pouvoir bien visualiser les détails importants à savoir le pitch et la fréquence des formants. Dans l'exemple de la voyelle « ou » comme « ouvrir », la durée est de 0.57s, la fréquence du fondamental est de 217.3917Hz, les fréquences des formants sont : F1=391Hz et F2=652Hz, la précision du spectre est de 100.

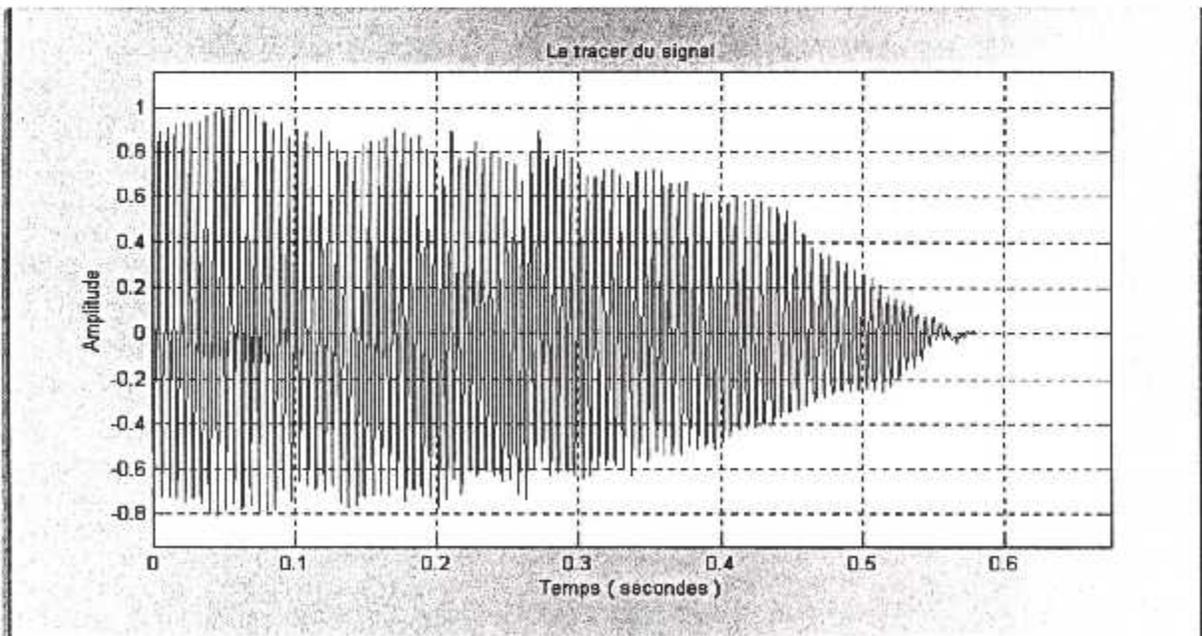


Fig V-12 Visualisation du signal de la voyelle orale « OU »

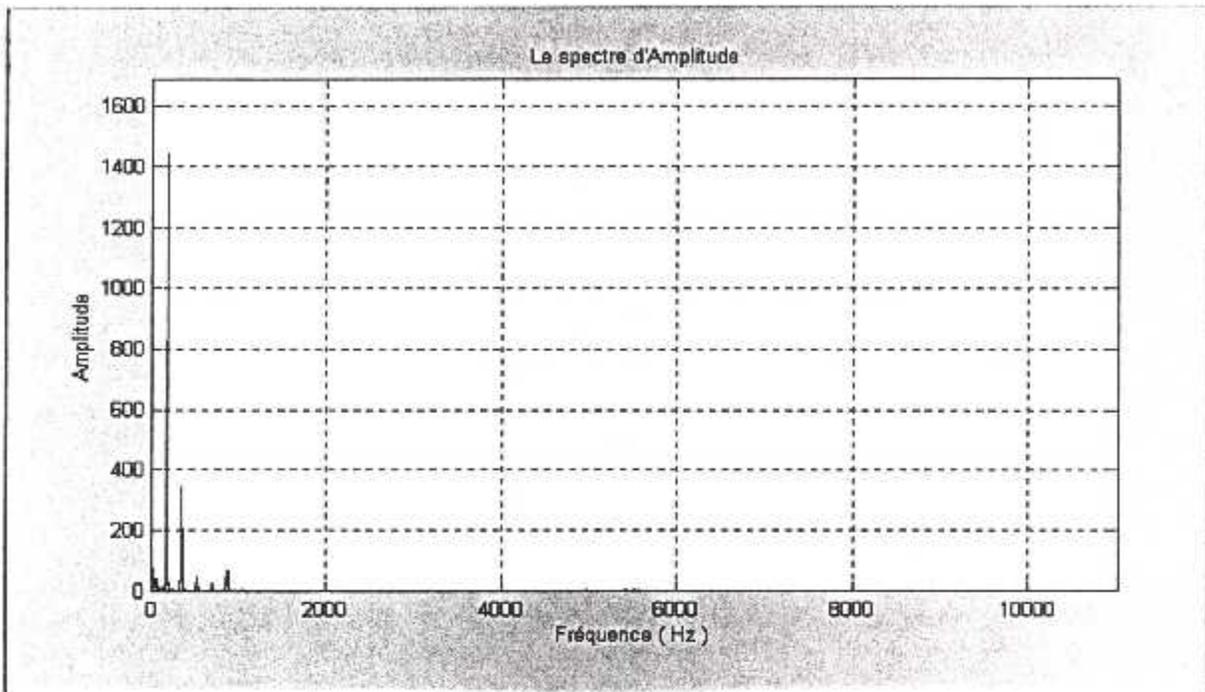


Fig V-13 Visualisation du spectre d'amplitude de la voyelle orale « Ou »

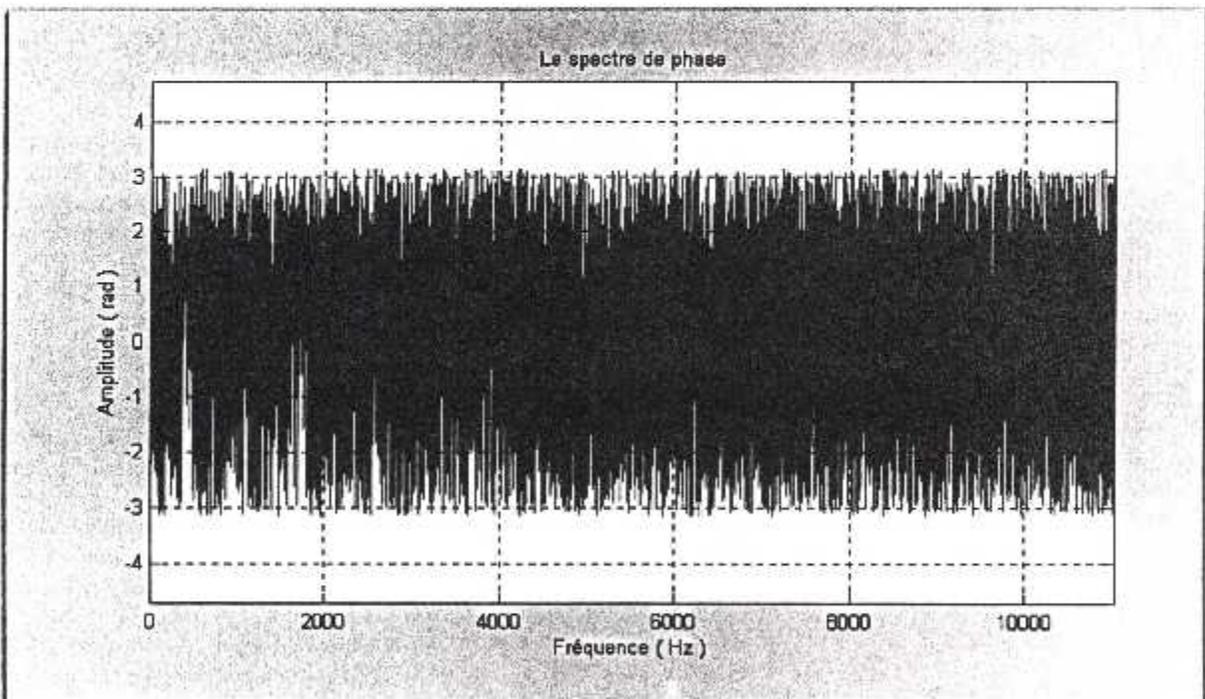


Fig V-14 Visualisation du spectre de phase de la voyelle orale « Ou »

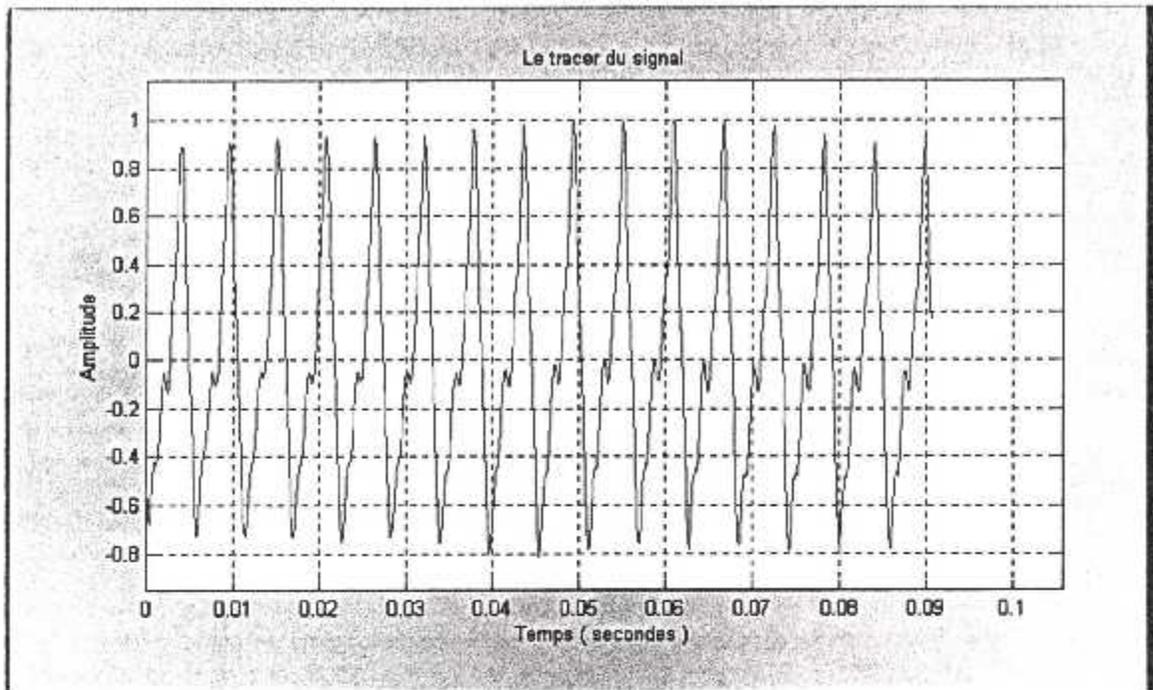


Fig V-15 Visualisation du zoom du signal de la voyelle orale « OU »

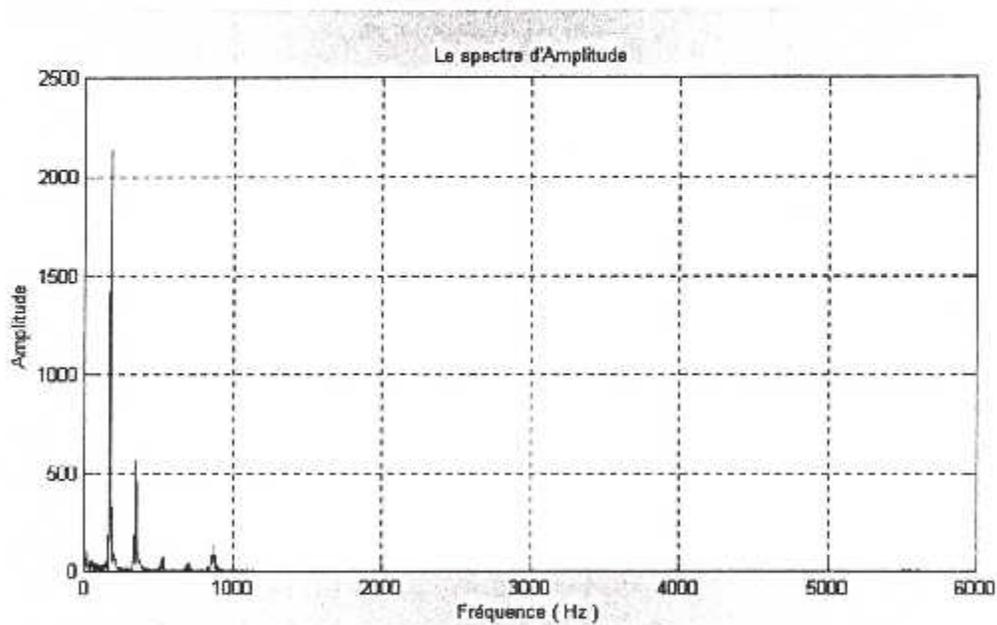


Fig V-16 Visualisation du spectre d'amplitude du signal de la voyelle orale « OU » zoomé

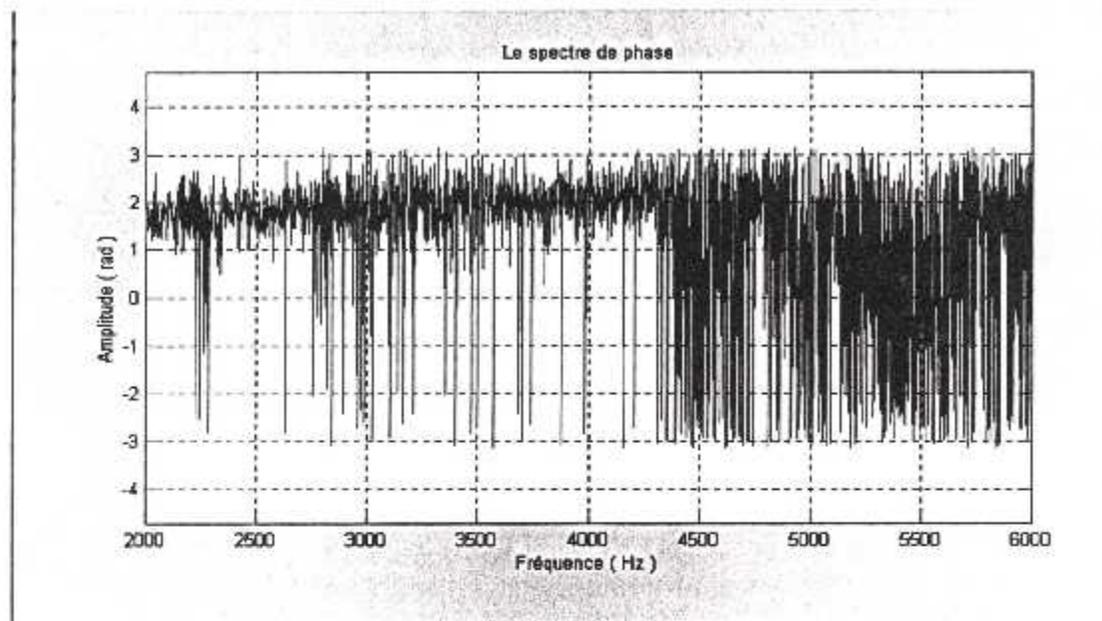


Fig V-17 Visualisation du spectre de phase du signal de la voyelle orale « OU » zoomé

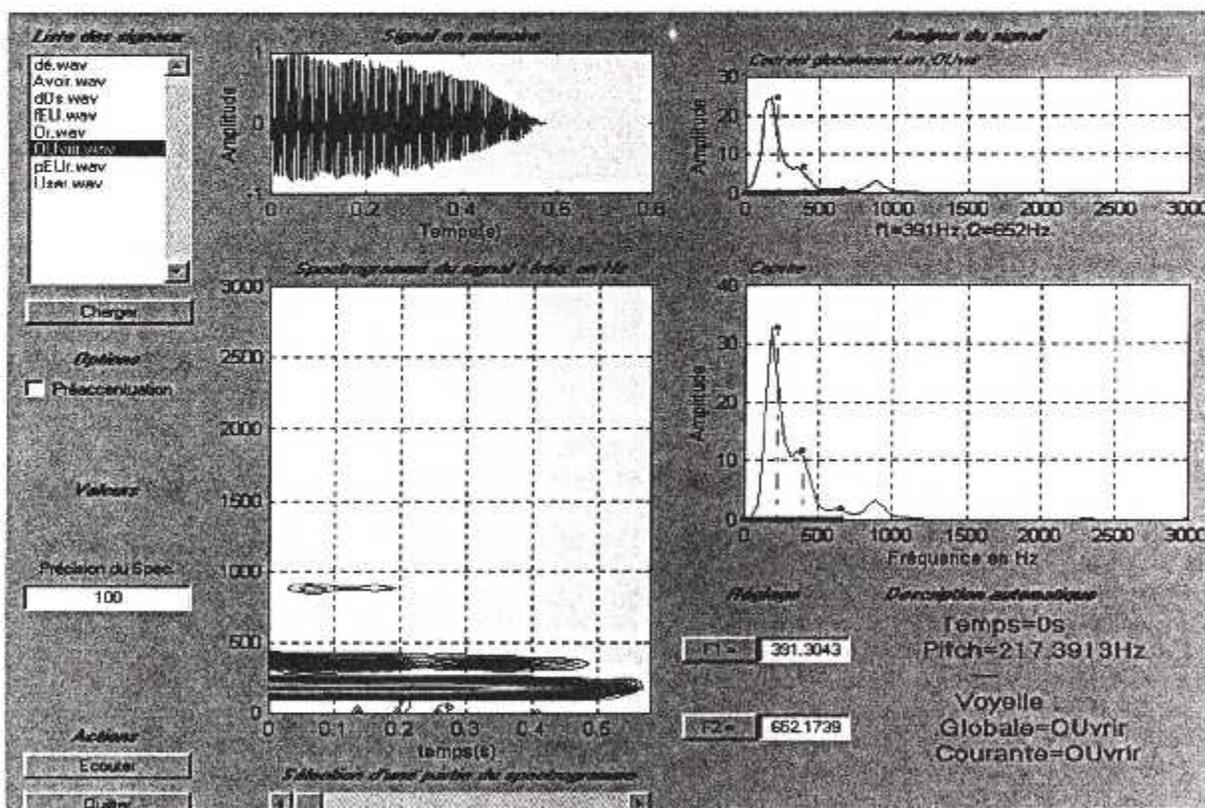


Fig V-18 Visualisation analytique du signal de la voyelle orale « OU »

Dans la figure V.18, le spectrogramme est tel que l'importance d'une composante spectrale de la voix est indiquée par la couleur. Le jaune-rouge indique, généralement (c'est une convention parmi d'autres), une très forte présence de cette composante contrairement aux bleus-noirs. La « précision du spec » placé à gauche permet de faire varier la palette de couleur et donc d'ajuster qualitativement le graphique à la nature du son analysé.

Conclusion générale

Conclusion Générale

Notre travail avait pour but le traitement numérique des signaux de parole. Pour ce faire, nous avons élaboré deux interfaces.

- Une interface consacrée au synthétiseur des signaux. Elle nous a permis de générer quelques signaux simples. Le choix de cette forme de synthèse a été fait afin de pouvoir tester l'éditeur de signal en vue d'une aide à l'analyse du signal de la parole.
- L'autre pour l'éditeur de signal, ce dernier nous a permis d'éditer et de traiter des signaux de fréquence bien déterminée.

Nous pouvons alors dire que notre éditeur de signal nous a donné des résultats que nous escomptions à savoir :

1. la détection du fondamental et de la fréquence des formants ;
2. le zoom de n'importe quelle portion du signal ;
3. l'écoute du signal ;
4. la visualisation de l'intensité de la voix.

Ce logiciel trouve également son application dans d'autres domaines tels que l'étude des signaux acoustiques, les vibrations des machines tournantes, l'étude de la stabilité des ailes. Les analyses spectrales peuvent contribuer à la détection des défauts là où ils apparaissent, permettant ainsi d'effectuer des réparations avant qu'une panne catastrophique ne se produise.

La simulation décrite ici peut être considérée comme un système de base pouvant servir à des études fondamentales et à des travaux pratiques pour de nombreuses manipulations.

Annexes

FFT : Fast Fourier Transform (transformée de Fourier rapide)

RNIS : ligne de transmission numérique qui a tendance à remplacer le modem. Il est très important dans les réseaux locaux en raison de sa rapidité (64000 à 1000000 de bps). Il n'a pas besoin de faire moduler ses « messages » binaires en sons pour transmettre les informations par le biais de la ligne téléphonique.

CD : compact disc (disque compact)

DVD : digital versatil disc

MP3 : format de compression audio

FM : frequency modulation (modulation de fréquence)

2D : deux dimensions

3D : trois dimensions

Rect : abréviation de rectangulaire

Tri : abréviation de triangulaire

Exp : abréviation de exponentielle

DDS : abréviation de dents de scie

Sin : abréviation de sinusoïdale

Imp : abréviation de impulsionnel

***.mat** : lieu de stockage des fichiers mathématiques

A : représente l'amplitude du signal

T : représente le temps

T_{inf} : représente la valeur inférieur du temps

T_{sup} : représente la valeur supérieur du temps

Long : représente la longueur du vecteur FFT

Rectpuls : fonction matlab permettant le tracé d'une impulsion rectangulaire

Tripuls : fonction matlab permettant le tracé d'une impulsion triangulaire

Lim : représente la limite quand T tend vers + ou - l'infini de l'impulsion exponentielle

Sawtooth : fonction matlab permettant le tracé d'un signal DDS

Cos : abréviation de cosinus

A₁ : représente l'amplitude de la première sinusoïde

A₂ : représente l'amplitude de la deuxième sinusoïde

W₁ : représente la fréquence de la première sinusoïde

W₂ : représente la fréquence de la deuxième sinusoïde

Phi₁ : représente la phase de la première sinusoïde

Phi₂ : représente la phase de la deuxième sinusoïde

Echelon : fonction matlab permettant le tracé d'une impulsion unité

Square : fonction matlab permettant le tracé du signal carré

Randn : fonction matlab permettant le tracé du bruit

Size : représente la longueur du fichier

X_{min}=min(T)-abs(min(T))/6 : formule matlab permettant d'ajuster la valeur minimale de l'axe des abscisses dans l'interface

X_{max}=max(T)+abs(max(T))/6 : formule matlab permettant d'ajuster la valeur maximale de l'axe des abscisses dans l'interface

Y_{min}=min(miny)-abs(min(miny))/6 : formule matlab permettant d'ajuster la valeur minimale de l'axe des ordonnées dans l'interface

Y_{max}=max(max y)-abs(max(max y))/6 : formule matlab permettant d'ajuster la valeur maximale de l'axe des ordonnées dans l'interface

WAVEREAD : c'est la lecture du fichier de WAV

N₁ : nombre minimal d'échantillons du signal WAV

N₂ : nombre maximal d'échantillons du signal WAV

BIBLIOGRAPHIE



- [1] M.Mokhtari et A.Mesbah 1995)
Apprendre et maîtriser matlab



- [2] Bennani Hicham (1996)
Elaboration d'un éditeur de signal sur PC
Thèse d'ingénieur, avionique
Institut d'aéronautique de Blida



- [3] Marut Kunt
Traitement numérique des signaux
Edition Dunod



- [4] René Boite et Murat Kunt
Traitement de la parole
Edition Presse
Polytechniques Romande



- [5] Calliope
La parole et son traitement automatique
Edition Masson



- [6] Cours destiné à l'ENST-Br (1984)
Analyse et synthèse de la parole



- [7] Mouloud Smassel juin 1992
Analyse et synthèse de la parole par LPC
Thèse d'ingénieur, télécommunication
Institut de télécommunication d'Oran



- [8] Bouanani Abdel Kader juin 1993
Le suivi des formants
Thèse d'ingénieur, télécommunication
Institut de télécommunication d'Oran



- [9] Meslem.M et Belaissa.B (1998)
Modélisation du conduit vocal par une méthode de simulation
Thèse d'ingénieur, avionique
Institut d'aéronautique de Blida