

Tesc

وزارة التعليم العالي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

ECOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التخصصات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

DEPARTEMENT : d'electronique

PROJET DE FIN D'ETUDES

SUJET

RECONNAISSANCE DE LA PAROLE
EN MODE MULTILOCUTEUR PAR
DES METHODES GLOBALES

Proposé par :

Etudié par :

Dirigé par :

M^r BENIDDIR . M^r BENACHOUR.A . M^r BENIDDIR

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT : d'électronique

المدرسة الوطنية المتعددة التقنيات
المكتبة —
BIBLIOTHEQUE —
Ecole Nationale Polytechnique

PROJET DE FIN D'ETUDES

SUJET

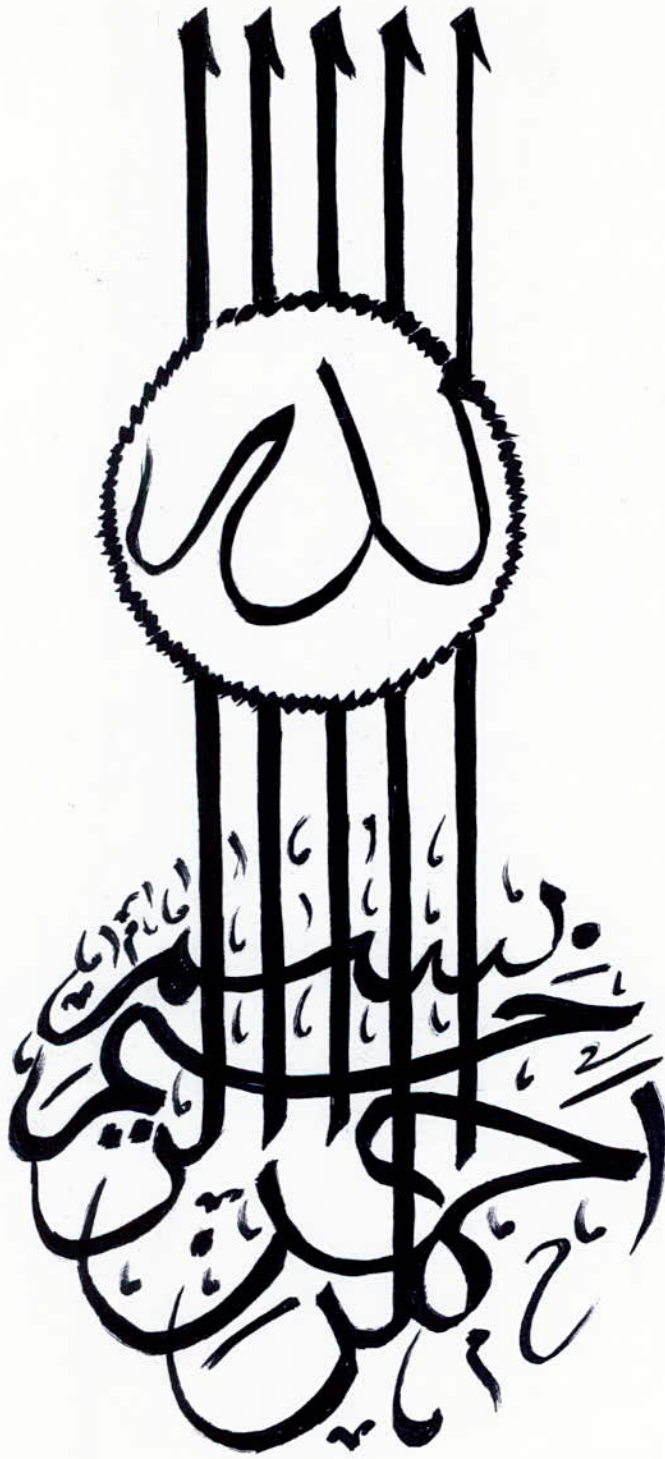
RECONNAISSANCE DE LA PAROLE
EN MODE MULTILOCUTEUR PAR
DES METHODES GLOBALES

Proposé par :
M^r BENIDDIR

Etudié par :
M^r BENACHOUR.A

Dirigé par :
M^r BENIDDIR

PROMOTION : JUIN 90



SOMMAIRE

I- INTRODUCTION

I-1 Généralités	1
I-2 Présentation du travail	4
I-3 Organisation générale du système étudié	6

II- ANALYSE ACOUSTIQUE

II-1 Introduction	7
II-2 Quelques méthodes d'analyse	8
II-2-1 Prédiction linéaire	8
II-2-2 Analyse par T.F.C.T	9
II-3 Analyse cepstrale	9
II-3-1 Systèmes homomorphiques	10
II-3-2 Systèmes convolutifs	11
II-3-3 Définition de l'opérateur de déconvolution ..	13
II-3-4 Définition du cepstre	13
II-3-5 Propriétés des coefficients cepstraux	14
II-4 Mise en oeuvre de l'analyse cepstrale	17
II-5 Organigramme	19

III- COMPARAISON DYNAMIQUE DES MOTS

III-1 Introduction	24
III-2 Position du problème	26
III-3 Définitions	27
III-3-1 Distance locale	27
III-3-2 Chemin d'alignement	27

IIID-3-3 <i>Distance globale</i>	28
IIID-4 Programmation dynamique	29
IIID-5 Détermination du chemin de déformation et de la distance globale associée	29
IIID-6 Organigramme	33
IV)- DÉCISION	
IVD-1 Introduction	39
IVD-2 Technique des K.N.N	39
IVD-3 Rejet	40
IVD-4 Organigramme	41
V)- APPRENTISSAGE	
VD-1 Introduction	44
VD-2 Situation du problème	45
VD-3 Algorithme de classification	46
VD-3-1 <i>Définitions</i>	47
VD-3-3-1 <i>Définition de la métrique</i>	47
VD-3-3-2 <i>Fonction d'homogénéité</i>	48
VD-3-3-3 <i>Fonction critère</i>	48
VD-3-2 <i>Algorithme</i>	49
VD-4 Choix des représentants	49
VD-5 Organigramme	50
VI)- TESTS ET RÉSULTATS	
VID-1 Introduction	54
VID-2 Tests en mode monolocuteur	55

VID-2-1 Influence de l'énergie	56
VID-2-2 Décalage des formants	57
VID-2-3 Influence de la longueur des mots	59
VID-2-4 Influence des paramètres décalage et longueur combinés	61
VID-2-5 Essai de quelques mots étrangers	63
VID-3 Tests en mode multilocuteur	64
VID-3 Conclusions	69
VID- CONCLUSIONS GÉNÉRALES	71
VII- ANNEXE	

CHAPITRE I INTRODUCTION

I-1 GENERALITES

La parole est le moyen le plus naturel et le plus utilisé par l'Homme pour communiquer avec ses semblables de l'espèce humaine. Cependant ces dernières années on assiste à une véritable profusion de machines de toutes sortes de telle façon qu'il est pratiquement impossible pour l'Homme moderne d'entreprendre un travail quelconque dans un environnement où la machine est absente.

Jusqu'à ces dernières années, on s'est toujours heurté au problème crucial de la communication Homme-Machine, et l'un des aspects les plus importants de ce problème est le langage qui véhicule cette communication.

Cette dernière s'est toujours faite à travers des boîtes à boutons, des claviers, des écrans de visualisation etc.....

Cette façon de procéder a deux inconvénients majeurs:

1- La mobilisation de certaines ressources de l'Homme qui auraient pu être utilisées par ailleurs pour d'autres tâches et la non utilisation d'autres telle que la parole qui est le moyen de communication par excellence de l'Homme.

2- Le deuxième inconvénient est lié à la vitesse de transmission de l'information qui est beaucoup plus conséquente lors de la communication parlée que celle mise en oeuvre à l'aide d'un clavier par exemple.

La reconnaissance automatique de la parole est un sujet de

recherche qui est apparue il n'y a pas longtemps (les premières tentatives remontent à 1939 avec le vocodeur de M. Dudley aux U.S.A).

Son but ultime est en quelque sorte de donner des oreilles aux machines et de rendre ainsi la communication Homme-machine plus naturelle et plus accessible .

Et outre l'intérêt purement scientifique de telles investigations l'arrivée à maturité de ces recherches devraient avoir des retombées extrêmement bénéfiques .

Dans l'état actuel de la connaissance en la matière , la réalisation d'un système de reconnaissance universel , capable de comprendre ou plus justement de reconnaître la parole dans toutes les situations est encore un objectif futuriste .

Les difficultés à résoudre avant d'atteindre éventuellement ce but ultime sont considérables

En effet , le signal acoustique de la parole est un signal extrêmement complexe et extrêmement variable et c'est là où réside la principale difficulté . Théoriquement toute langue est constituée d'un ensemble fini et relativement restreint d'éléments phonétiques qu'on appelle phonèmes , se sont les éléments de base qui entrent dans la constitution de n'importe quel son d'une langue [11]. Toutefois, le contenu physique de ces phonèmes est sujet à une grande variabilité suivant le contexte où ils apparaissent . C'est ce que les spécialistes du domaine appellent le phénomène de coarticulation . En d'autres mots , c'est l'influence qu'exercent les phonèmes voisins sur l'articulation d'un phonème donné.

On n'a jusqu'à présent pas réussi à trouver des modèles généraux pour représenter ces phonèmes .Les recherches actuelles abordent le problème de la reconnaissance de la parole par les machines suivant deux optiques :

1- les méthodes globales :

Dans ces méthodes les mots sont considérés comme des entités entières et sont codés comme telles et un silence suffisant doit obligatoirement séparer 2 mots consécutifs .Le système compare le mot à un ensemble d'empreintes vocales préalablement enregistrées et dégage celle qui est la plus proche s'il ya lieu .Beaucoup de réalisations sont aujourd'hui disponibles sur le marché avec des performances tout à fait acceptables .

Le principal désavantage est lié au fait que dans les meilleurs des cas , le nombre de mots que le système est en mesure de reconnaître est encore très limité .

2- Les méthodes analytiques :

Ce sont les approches les plus intéressantes mais aussi les plus difficiles .Elles sont intéressantes car elles nous donnent la possibilité de reconnaître la parole continue , elles sont difficiles au vu des nombreux problèmes qui restent encore sans réponse générale satisfaisante .Le processus de reconnaissance passe par plusieurs étapes .La première est celle d'analyse : la parole est segmentée c'est à dire divisée en segments , chaque segment représente l'emplacement probable d'un phonème ou autre unité phonétique choisie .Les paramètres pertinents de ce segment sont extraits par une méthode

d'analyse adéquate et comparés aux phonèmes du dictionnaire , il en ressort de cette comparaison des probabilités de correspondance aux différents phonèmes . Ces probabilités seront ensuite affinées en utilisant des informations syntaxiques, sémantiques et contextuelles . C'est une méthode très prometteuse notamment avec l'arrivée en puissance des systèmes experts . Il faut également signaler que les systèmes disponibles sont soit monolocuteur soit multilocuteur .

Les premiers sont des systèmes qui ne reconnaissent que les mots prononcés par le locuteur dont les mots en référence dans le dictionnaire sont les siens . Un essai de reconnaissance par un autre locuteur s'accompagne en règle générale par une dégradation du taux de reconnaissance . La raison en est que la structure d'un même mot change d'un locuteur à un autre , cette variabilité est présente même pour un locuteur donné . Les seconds sont une tentative de généralisation des premiers , leurs objectifs est d'arriver à reconnaître une catégorie plus importante de locuteurs . On atteint cet objectif en utilisant certaines astuces comme par exemple l'utilisation de plusieurs références par mot . Ces références seraient représentatives d'un groupe important de locuteurs , ou encore l'adaptation automatique aux nouveaux locuteurs .

1D-2 PRESENTATION DU TRAVAIL

L'objectif de notre travail est l'étude d'un système de reconnaissance de la parole par la méthode globale en mode multilocuteurs pour une éventuelle implantation sur un

microprocesseur tel que le TMS 320 10 .Nous avons dans ce cadre réalisé :

- ** Un programme complet de reconnaissance en mode monolocuteur
- ** Un programme de classification automatique en vue de tirer les représentants des mots à reconnaître et les intégrer dans le dictionnaire du système .
- ** Un programme complet de reconnaissance en mode multilocuteurs

Nous avons également établi un ensemble de tests destinés à voir le comportement de nos différents programmes .Tous nos programmes ont été écrit en langage évolué en l'occurrence le TURBO-PASCAL et les tests ont été réalisés en simulant les voyelles A O I E U Y .

Nous avons donc étudié successivement

- L'analyse acoustique
- La normalisation temporelle
- La classification
- La décision
- Tests et resultats

ID-3 ORGANISATION GENERALE DU SYSTEME ETUDIE

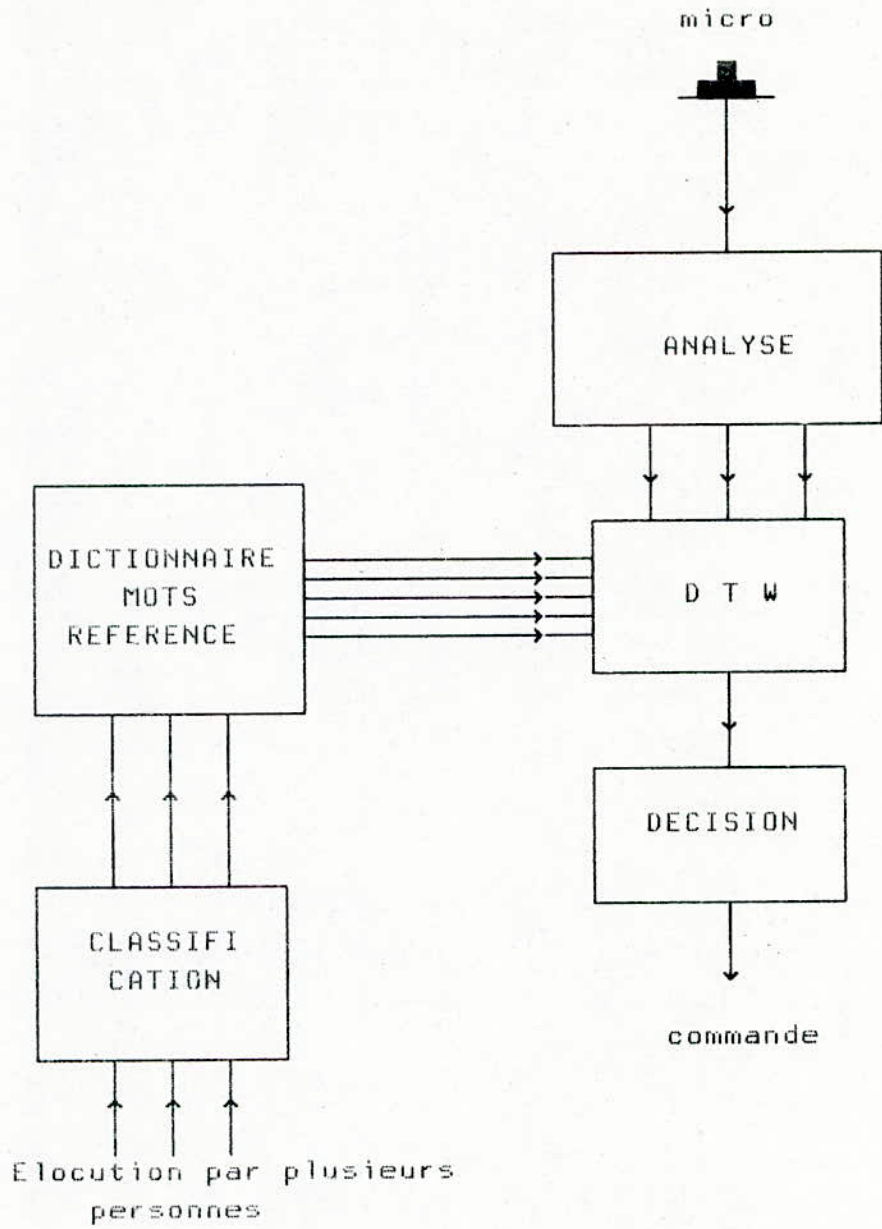


Fig (1-1)

CHAPITRE II ANALYSE ACOUSTIQUE

II-1 INTRODUCTION

L'analyse acoustique est la première étape que subit le signal de la parole pendant le processus de reconnaissance . Elle a deux rôles :

-- Réduction du volume des informations contenues dans le signal original .

-- Extraction des paramètres pertinents représentatifs qui soient des caractéristiques propres au mot énoncé.

L'analyse acoustique est un des problèmes fondamentaux de notre travail .

Le signal de la parole est un signal qui véhicule des informations de nature très diverses c'est aussi un signal extrêmement redondant .

Ces informations ont trait à l'état physique et psychique de la personne qui parle et également des informations concernant le mot lui-même . La nature de notre travail , qui rappelle le s'agit d'étudier un système multilocuteurs , donc l'analyse doit s'attacher à filtrer toutes les informations qui concernent le locuteur et ne garder en définitive que ce qui est propre au mot prononcé .

Le signal de la parole est produit par le système phonatoire constitué par une source d'énergie représentée par les poumons et d'un ensemble de cavités (cavité buccale , cavité nasale

et cavité pharyngiale qui forment des résonateurs acoustiques) .Ceux-ci modulent le signal issu des poumons et transitant ou non à travers les cordes vocales Cette brève description du système phonatoire est utile pour connaître la manière avec laquelle sont formés les sons et les mots et orientera notre choix pour la méthode d'analyse du signal acoustique .

II)- 2 QUELQUES METHODES D'ANALYSE

II)-2-1 Prediction linéaire

Cette méthode se base sur l'idée qu'un échantillon (dans le domaine temporel) du signal acoustique peut être prédit par la somme pondérée des p échantillons passés du même signal .

$$S_p(n) = \sum_{k=1}^p C(k) \cdot S(n-k)$$

p : ordre de prediction .

S_p : signal prédit .

S : signal reel .

Le problème étant de trouver les coefficients de pondération C(k) pour minimiser la différence $S(n) - S_p(n)$.

Le problème est assez similaire à celui qui consiste à calculer un filtre .La prédiction linéaire vise en fait à modéliser le conduit vocal par un filtre approprié .Chaque mot correspond à un filtre . Cette méthode est essentiellement utilisée en synthèse vocale .

II)-2-2 Analyse par T.F.C.T :

C'est une analyse du type fréquentielle . Elle est réalisée à travers un banc de filtres (logiciels ou matériels) et les énergies issues de chaque filtre sont prises comme paramètres pertinents.

IID-3 ANALYSE CEPSTALE :

L'idée directrice de cette méthode d'analyse est que le signal de la parole $S(n)$ est considéré comme résultant d'une combinaison par convolution de deux signaux :

- L'excitation représentée par le signal $e(t)$ d'une part et issu de la source du système phonatoire .
- La réponse impulsionnelle du filtre représenté par le conduit vocal $h(n)$ d'autre part .

La modélisation du conduit vocal par un filtre n'est pas arbitraire , celui ci se conduit réellement comme tel en ce sens que comme on l'a vu , étant constitué de cavités raisonnantes , il amplifie certaines fréquences et en atténue d'autres .

Nous écrivons donc :

$$S(n) = e(n) * h(n) = \sum_{k=-\infty}^{+\infty} h(n-k) \cdot e(k)$$

* étant l'opérateur de convolution

Le principe de la méthode est donc de séparer ou d'isoler le signal $h(n)$, lié à la forme du conduit vocal qui est

l'information qui nous intéresse le plus car elle caractérise ce qui a été dit .

L'information phonétique se trouve en effet dans ce qu'on appelle les formants (resonances du conduit vocal) .

11)-3-1 Systèmes homomorphiques:

Le problème de la séparation des signaux combinés par convolution est un cas particulier d'un problème beaucoup plus général en l'occurrence , la séparation des signaux combinés suivant une lois quelconque .

Le problème a été abordé par certain chercheurs dont notamment OPPENHEIM dans ce que l'on appelle les systèmes homomorphiques

Soient donc 2 signaux $x_1(k)$ et $x_2(k)$ et soient les lois \square et \oplus respectivement , les lois de combinaison des signaux entre eux à l'entrée et à la sortie du système H et \blacksquare , \oplus les lois de combinaison d'un signal avec un scalaire respectivement à l'entrée et à la sortie du même système .

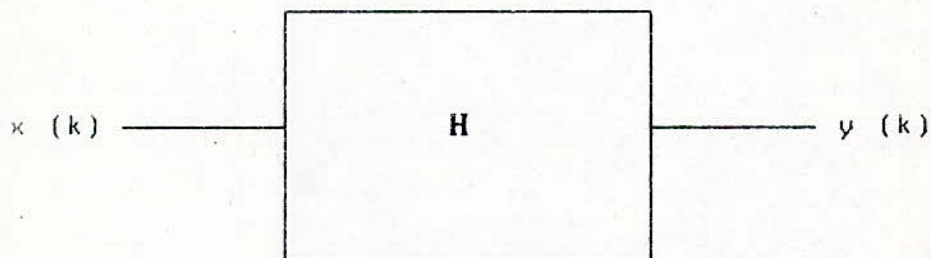


Fig (2-1)

Nous avons les relations suivantes :

$$H \left[x_1(k) \square x_2(k) \right] = H \left[x_1(k) \right] \ominus H \left[x_2(k) \right]$$

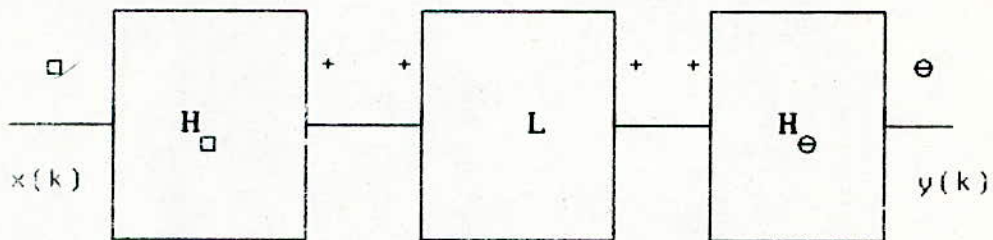
et

$$H \left[A \blacksquare x(k) \right] = A \oplus H \left[x(k) \right]$$

Ce système est dit homomorphique, c'est une généralisation des systèmes linéaires. On retrouve d'ailleurs ces derniers en faisant :

$\square = \ominus = +$ et $\blacksquare = \oplus = \times$ (multiplication)

Un résultat très important et extrêmement intéressant démontré par OPPENHEIM est la possibilité de représenter un système homomorphique quelconque par 3 systèmes en cascade dont celui du milieu est un système linéaire



L'utilité d'une telle représentation tient au fait qu'il est désormais possible d'appliquer tous l'arsenal mathématique développé pour les systèmes linéaires, pour l'étude des systèmes homomorphiques complexes.

II)-3-2 Systèmes convolutifs :

Dans le cas particulier qui nous intéresse nous rappelons que

:

$$S(n) = h(n) * e(n)$$

* : opérateur de convolution

e(n) : l'excitation

h(n) : échantillons de la réponse impulsionnelle du filtre représentant le conduit vocal

S(n) : échantillons du signal composite

L'opération de déconvolution doit donc séparer les 2 signaux e(n) et h(n)

Formellement on a :

$$\begin{aligned} D[S(n)] &= D[e(n) * h(n)] \\ &= D[e(n)] + D[h(n)] \end{aligned}$$

$$\hat{S}(n) = \hat{e}(n) + \hat{h}(n)$$

$\hat{S}(n)$, $\hat{e}(n)$, $\hat{h}(n)$ sont respectivement les résultats de l'application de l'opérateur de déconvolution aux signaux S(n), e(n) et h(n).

Un autre intérêt du passage d'une combinaison par convolution à une combinaison par addition est la possibilité de les séparer simplement par multiplication par une fonction porte, à condition bien sûr que les deux signaux occupent des domaines différents (le domaine dont il question ici est le domaine où évoluent les signaux : domaine temporel, fréquentiel, qu'éfrentiel ect.....).

Nous avons ensuite la possibilité de retrouver les signaux originaux par application simple de l'opérateur inverse c'est

à dire D^{-1}

Ainsi nous aurons :

$$h(n) = D^{-1} \left[\hat{h}(n) \right]$$

II)-3-3 Définition de l'opérateur D

Pour trouver l'opérateur mathématique capable de satisfaire à nos besoins, nous partons de la remarque que la fonction logarithme permet de transformer un produit en une somme :

$$\text{Ln}(x.y) = \text{Ln}(x) + \text{Ln}(y)$$

propriété caractéristique de la fonction logarithme .

Or la transformée en Z de deux signaux combinés par convolution est justement le produit des deux transformés en Z individuelles .

$$T_z \left[h(n) * e(n) \right] = T_z \left[h(n) \right] \cdot T_z \left[e(n) \right]$$

$$S(Z) = H(Z) \cdot E(Z)$$

Donc en calculant successivement la transformée en Z du signal de la parole et en calculant le logarithme on aura réalisé l'opérateur D cherché .

$$S(n) \xrightarrow{\text{TZ}} S(Z) \xrightarrow{\text{log}} \text{Ln} \left[S(Z) \right]$$

II)-3-4 Définition du Cepstre

Le cepstre réel d'un signal est la transformée de fourier

inverse du logarithme du module du spectre du signal .

On le définit mathématiquement par :

$$C(n) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} \text{Ln} | S(e^{j\nu}) | e^{jn\nu} d\nu$$

II)-3-5 Propriétés des coefficients cepstraux

pour mettre en évidence certaines propriétés du cepstre nous allons considérer un exemple qui se prête bien au calcul analytique .

Soit un signal $S(n)$ dont la transformée en Z est $S(Z)$

$$S(Z) = \frac{\prod_{k=1}^{K1} (1 - a_k Z^{-1}) \prod_{k=1}^{K2} (1 - b_k Z)}{\prod_{k=1}^{K3} (1 - c_k Z^{-1}) \prod_{k=1}^{K4} (1 - d_k Z)}$$

a_k , b_k , c_k et d_k sont tel que leurs modules soient inférieur à 1 .

$$S(Z) = \text{Ln}(A) + \sum_{k=1}^{K1} \text{Ln}(1 - a_k Z^{-1}) + \sum_{k=1}^{K2} \text{Ln}(1 - b_k Z) - \sum_{k=1}^{K3} \text{Ln}(1 - c_k Z^{-1}) - \sum_{k=1}^{K4} \text{Ln}(1 - d_k Z)$$

En développant les termes de cette relation en série de puissance sachant que :

$$\ln(1-x) = - \sum_{k=1}^{+\infty} \frac{x^k}{k} \quad \text{pour } |x| < 1$$

On obtient :

$$\ln(1 - a_k Z^{-1}) = - \sum_{n=1}^{\infty} \frac{a_k^n Z^{-n}}{n}$$

$$\ln(1 - b_k Z) = - \sum_{n=1}^{\infty} \frac{b_k^n Z^n}{n}$$

$$\ln(1 - c_k Z^{-1}) = - \sum_{n=1}^{\infty} \frac{c_k^n Z^{-n}}{n}$$

$$\ln(1 - d_k Z) = - \sum_{n=1}^{\infty} \frac{d_k^n Z^n}{n}$$

On reconnaît dans ces expressions les transformés en Z des signaux numériques :

$$\begin{aligned} & - \frac{a_k^n}{n} e^{(n-1)} \\ & \frac{b_k^n}{n} e^{(-n+1)} \\ & - \frac{c_k^n}{n} e^{(n-1)} \\ & \frac{d_k^n}{n} e^{(-n+1)} \end{aligned}$$

d'ou :

$$\hat{S}(n) = \begin{cases} \ln(A) & \text{pour } n=0 \\ \sum_{k=1}^{K1} \frac{a_k^n}{n} + \sum_{k=1}^{K3} \frac{c_k^n}{n} & n > 0 \\ \sum_{k=1}^{K2} \frac{b_k^{-n}}{n} - \sum_{k=1}^{K4} \frac{d_k^{-n}}{n} & n < 0 \end{cases}$$

Pour les systèmes à phase minimale ($b_k = d_k = 0$) on a :

$$\hat{S}(n) = 0 \text{ pour } n < 0$$

La propriété très intéressante qui apparait ici est la décroissance en $1/n$ des coefficients cepstraux .

n est le rang du coefficient , ce qui nous permettra de nous contenter d'un nombre réduit de ces coefficients pour représenter un mot .

C'est un des atouts essentiels de l'analyse cepstrale en reconnaissance globale , dans la perspective de l'implantation de ce travail dans un petit système à base d'un microprocesseur où la contrainte d'espace mémoire joue un rôle déterminant sur ce type de choix.

Le cepstre est la transformée de Fourier inverse du logarithme du module du spectre du signal .

Pour un son voisé , par exemple , des ondulations rapides dues au pitch (période de vibration des cordes vocales) sont

superposés aux ondulations moins rapides correspondant à la structure formantique du conduit vocal .

Dans le domaine cepstral où les signaux occupent des domaines différents , les premiers coefficients sont caractéristiques de la structure du conduit vocal correspondant au mot prononcé . Plus le nombre de coefficients est élevé mieux est la représentation du conduit vocal .

Toutefois lorsque le rang des coefficients est assez élevé on risque de prendre des informations dépendant du locuteur .

En pratique on prend les 8 premiers coefficients à l'exception du coefficient d'ordre 0 $C(0)$ qui représente l'information sur l'énergie .

II)-4 MISE EN OEUVRE DE L'ANALYSE CEPSTRALE

Le conduit vocal évolue au cours de la prononciation des mots , c'est à dire que ses dimensions changent dans le temps .

Cependant cette évolution est suffisamment lente pour considérer que pendant des périodes d'environ 20 ms on peut l'admettre comme stationnaire .

L'analyse se fera donc sur des trames de 20 ms, soit un ensemble de 256 échantillons pour une fréquence d'échantillonnage de 12,8 kHz . Le spectre étant limité à 6 kHz .

Mais avant d'extraire les paramètres pertinents du signal vocal par analyse cepstrale , le signal subit un traitement préliminaire qui se fait en deux étapes successives :

1-- PREACCENTUATION

Le signal de la parole subit un affaiblissement dans la partie haute fréquence du spectre ,par suite de la désadaptation des deux milieux où il évolue : le conduit vocal et l'air extérieur .

De plus les informations contenues dans cette partie du spectre sont souvent déterminantes .

La préaccentuation a pour but de rétablir le niveau énergétique de cette partie du spectre .

L'affaiblissement est estimé à une pente décroissante de 6 dB/octave [6]

2-- FENETAGE

Le fenêtrage a pour but de limiter les effets néfastes de la limitation temporelle du signal .

En effet le spectre d'un signal multiplié par une fonction porte subit des distortions dont il faudra chercher à atténuer de la manière la plus grande possible .

Il existe plusieurs sortes de fonctions fenêtre dont la plus utilisée en traitement de la parole est sans doute la fenêtre de Hamming définie par :

$$w_h(n) = 0.54 - 0.46 \cos(2\pi (n-1) / N)$$
$$n = 1 \dots \dots \dots N$$

L'analyse acoustique est mise en oeuvre en suivant les étapes dans l'ordre :

--- Prise des trames de 256 échantillons

--- Préaccentuation : cette opération est réalisée par programme suivant la formule $S_p(n) = S(n) - S(n-1)$

--- Fenêtrage $S_p(n) = S_p(n) * W_h(n)$

--- Calcul de la T.F.D par la F.F.T. et calcul du logarithme du module des échantillons de la T.F.D.

$$S(k) = \sum_{n=0}^{N-1} S_p(n) e^{-jnk2\pi/N} \quad \text{et} \quad \text{Ln}|S(k)|$$

$N = 256$

--- Calcul de la T.F.I (transformée inverse de Fourier) du logarithme du module de S(k) .

$$C(n) = \frac{1}{N} \sum_{k=0}^{N-1} \text{Ln}|S(k)| e^{jnk\frac{2\pi}{N}}$$

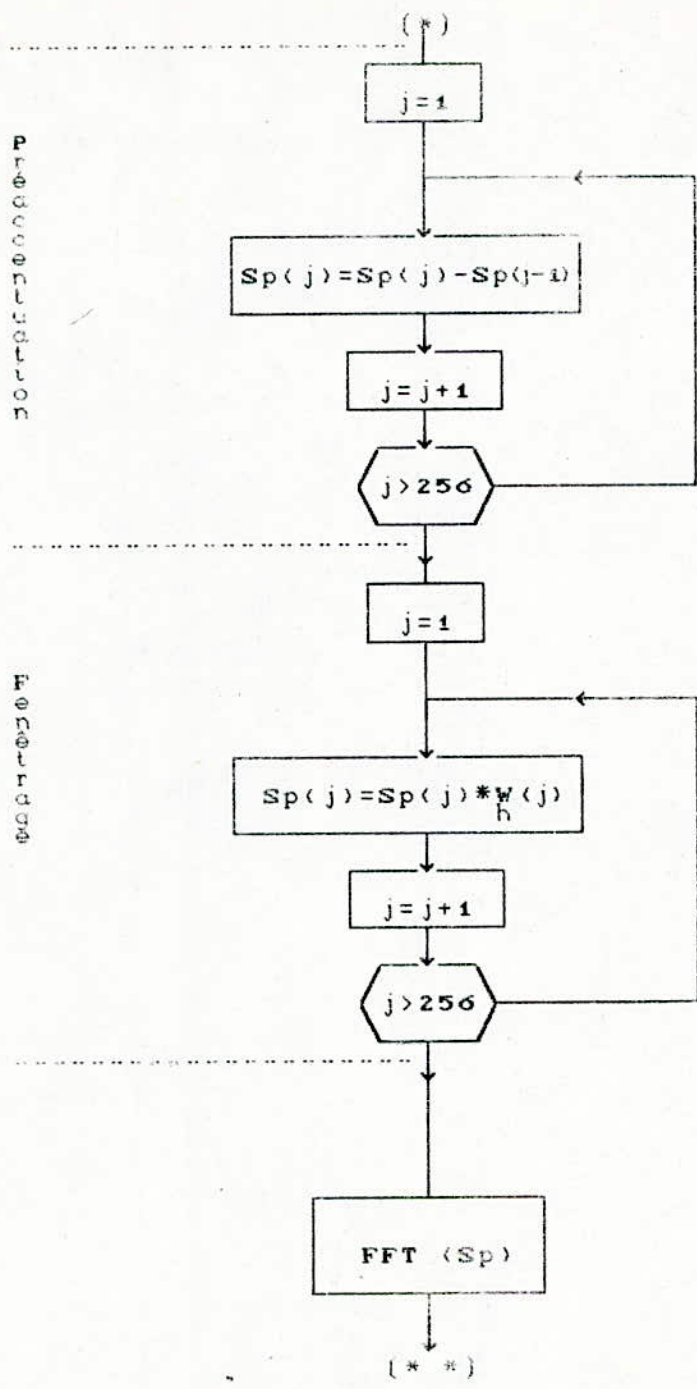
--- Enfin prise des 16 premiers coefficients comme paramètres représentatifs .

II)-5 ORGANIGRAMME DE L'ANALYSE

Dans notre programme de simulation les échantillons des mots sont lus dans des fichiers que nous appellons ici ECH et mis dans le tableau Sp pour traitement .

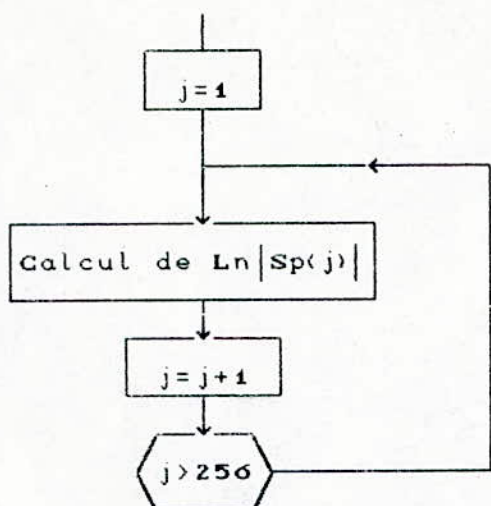
Nech : nombre d'échantillons du signal

Les coefficients cepstraux calculés sont stockés dans des fichiers ANA

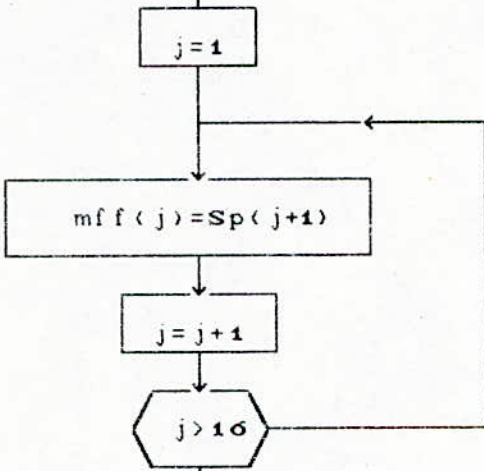


Calcul de la fonction de transfert
des coefficients de la fonction de transfert

(**)



$FFT^{-1}(Sp)$



(**)

III-1 INTRODUCTION

Après avoir analysé et codé le signal acoustique émis par l'opérateur humain, nous devons maintenant chercher un moyen pour comparer ce mot aux différentes empreintes vocales stockées en mémoire dans le dictionnaire comme références.

Dans cet effort nous devons tenir compte de certains aspects caractérisant la parole : c'est la variabilité extrêmement grande qui se manifeste même pour un locuteur donné et cette variabilité est de 4 sortes :

-- Variabilité fréquentielle :

Tout locuteur possède une identité formantique propre à chaque son qu'il prononce. La source de cette variabilité est la différence des dimensions des systèmes phonétiques d'une personne à une autre, ceux-ci, en effet ne sont pas rigoureusement identiques.

-- Variabilité énergétique :

Se manifeste par ce qu'il est courant d'appeler le volume du son c'est à dire son intensité. En effet cette intensité varie suivant que le locuteur crie ou chuchotte.

-- Variabilité prosodique :

Les informations prosodiques sont des informations qui dépendent du locuteur, elles sont donc dépendantes de la personne qui parle.

-- Variabilité temporelle :

La source de cette variabilité est la vitesse d'élocution qui diffère d'une personne à une autre et d'une situation à une autre. Toutes ces variabilités sont prises en charge dans les systèmes de reconnaissance globaux à différents niveaux. Pour l'énergie, nous avons le droit de penser que ce problème a été pris en charge automatiquement lors de la phase d'analyse du signal. Nous avons en effet fait remarquer (il est facile de le montrer) que les coefficients cepstraux retenus comme paramètres représentatifs avaient l'avantage d'être normalisés en énergie.

Les variabilités prosodiques sont prises en charge au niveau du module analyse où seules les informations propres au mot prononcé sont gardées (coefficients cepstraux).

Les variabilités fréquentielles seront traitées notamment pendant la phase de classification. Reste le problème des variabilités temporelles dont le module de la comparaison dynamique des mots est justement destiné à régler. Nous savons que les constituants phonétiques d'un mot ne subissent pas le même degré de déformation. Certaines parties du mot sont plus stables donc facilement extensibles ou compressibles sans que cela ne nuise à leur intelligibilité que certains phénomènes acoustiques beaucoup moins sujet à de telles variations à cause de leur nature même, tels que les sons plosifs.

Ces dernières constatations mettent à jour un besoin de normalisation non-linéaire.

III)-2 POSITION DU PROBLEME

Soit à comparer un mot prononcé par un locuteur (mot test T) avec un mot référence R appartenant au dictionnaire préalablement stocké en mémoire .

Après analyse , les deux mots se présentent comme des suites de vecteurs acoustiques :

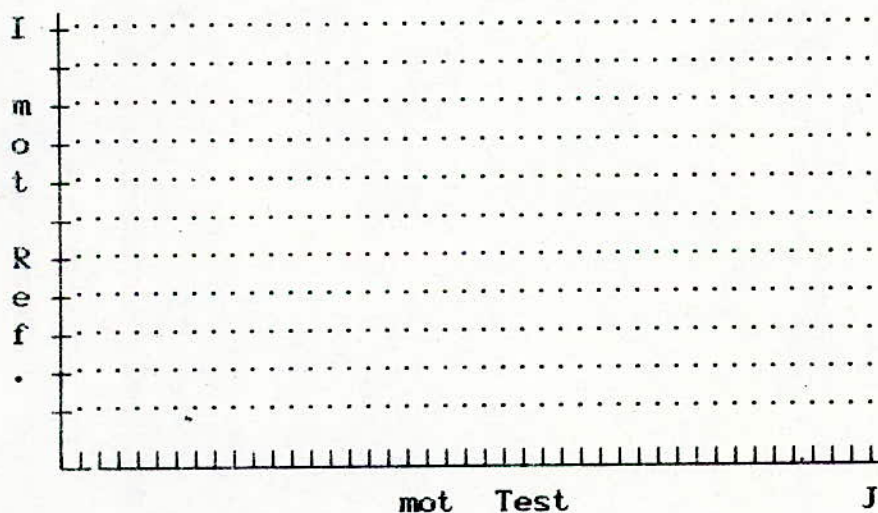
$$T \quad : \quad \left[\begin{array}{c} T(i) \end{array} \right] \quad i = 1 \dots I$$

$$R \quad : \quad \left[\begin{array}{c} R(j) \end{array} \right] \quad j = 1 \dots J$$

I , J dependent de la longueur des mots Test et Référence
Chaque vecteur acoustique contient les huit coefficients cepstraux relatifs à une trame donnée .

Pour mieux illustrer les idées qui viendront adoptons la représentation graphique suivante :

Fig (3-1)



Le rectangle dont les cotés sont I et J est appelé par la suite espace de recherche .

Le problème qui se pose est comment mesurer la proximité ou la similitude des deux mots , en d'autres termes il faudra définir une procédure ou une mesure qui , par exemple , sera d'autant plus petite que les mots sont plus proches phonétiquement et en particulier cette mesure devra être idéalement nulle si les mots à comparer T et R sont les mêmes avec cependant des longueurs différentes .

III)-3 DEFINITIONS

III)-3-1 Distance locale :

La distance locale entre une trame du mot Test $T(i)$ et une trame du mot Référence $R(j)$ notée $d(i,j)$ est une mesure de la dissemblance (ou ressemblance) locale entre les deux trames. Nous avons plusieurs choix dans la définition d'une telle distance , parmi lesquelles nous avons opté pour la distance géométrique en raison de sa simplicité :

$$d(i,j) = \sum_{k=1}^P | i(k) - j(k) |$$

P étant le nombre de coefficients cepstraux retenus pour une trame .

III)-3-2 Chemin d'alignement :

Nous appelons ainsi un ensemble F de points appartenant à

l'espace de recherche $F = C(1), \dots, C(K)$. Chaque point est en fait un couple de trames l'une appartenant au mot Référence , l'autre au mot Test.

$$C(k) = \left[i(k) , j(k) \right]$$

Le chemin d'alignement définit une sorte de projection du mot Test sur le mot Référence .

III)-3-3 Distance globale :

La distance globale est la somme pondérée des distances locales sur un chemin d'alignement , elle mesure la dissemblance globale entre les mots Test T et Référence R .

$$D_g = \sum_{k=1}^K d[C(k)] W(k)$$

$W(k)$: coefficients de pondération .

Les mots étant de longueurs différentes , la distance globale doit donc être normalisée en longueur pour pouvoir être comparable .

Pour cette raison on prend :

$$D_g = \frac{\sum_{k=1}^K d[C(k)] W(k)}{\sum_{k=1}^K W(k)} \quad (3-1)$$

III)-4 PROGRAMMATION DYNAMIQUE

Le problème central est donc de trouver un chemin qui relie un certain nombre de points $C(k)$ appartenant à l'espace de recherche pour lequel la distance globale normalisée définie par (3-1) est minimale. Le chemin d'alignement optimal est ce que l'on appelle la fonction de déformation.

Cependant il serait vain d'essayer de trouver le chemin de déformation en procédant à une recherche exhaustive de tous les chemins possibles, le temps de calcul excessif nécessaire à une telle opération enlèverait tout intérêt pratique à notre système, qui plus est aspire à une réalisation temps réel.

Les chercheurs ont toutefois trouvé le moyen de contourner cette difficulté en adoptant la méthode de la programmation dynamique dans la comparaison des mots.

Ce concept général a été originalement énoncé par BELLMAN.

Sa formulation générale est :

Une politique est optimale si, à un instant donné, quelque soient les décisions précédentes prises, les décisions qui restent à prendre constituent une politique optimale en regard des décisions précédentes.

III)-5 DETERMINATION DU CHEMIN DE DEFORMATION ET LA DISTANCE GLOBALE ASSOCIEE

La détermination de la fonction de déformation ou chemin optimal dans l'espace de recherche par la méthode de la programmation dynamique doit obéir aux restrictions suivantes :

Contraintes locales

1-- Conditions aux limites : Le chemin de déformation doit de toute évidence démarrer du point (1,1) et aboutir au point (I,J) .

2-- Le signal de la parole progresse d'une manière monotone ce qui implique que , sur le chemin de déformation on doit avoir :

$$i(k+1) \geq i(k) \text{ et } j(k+1) \geq j(k)$$

3 -- Conditions de continuité :

$$i(k+1) - i(k) \leq 1 \text{ et } j(k+1) - j(k) \leq 1$$

La combinaison de ces deux dernières contraintes nous amènent à trois sous chemins possibles partant d'un point donné comme le montre la Fig (3-2)

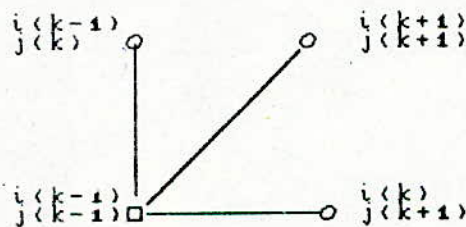


Fig (3-2)

4 -- Condition de pente :

La condition de pente a été introduite pour éviter la

coïncidence de certains phénomènes acoustiques courts avec d'autres plus longs et différents phonétiquement . Cette contrainte consiste à imposer à la fonction de déformation lorsqu'elle a parcouru m points consécutifs dans la direction des i ou celle des j de faire n points dans la direction diagonale avant de reprendre éventuellement la direction i ou celle des j

La contrainte de pente est définie par $P = n/m$

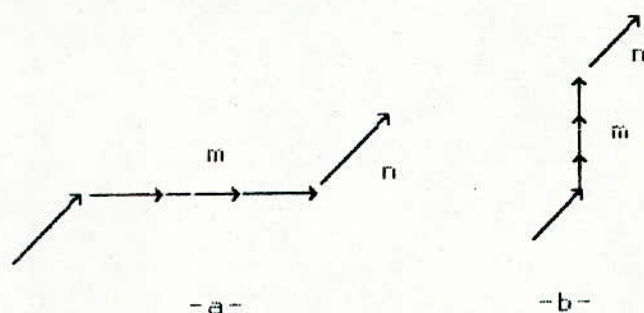


Fig (3-3)

La prise en compte de toutes ces contraintes peut se faire d'une manière asymétrique ; c'est une approche due à ITTAKURA . Les dédoublements et les omissions de trames ne sont possibles que sur le mot Référence , le principal avantage de la forme asymétrique est son gain en temps de traitement . Nous avons d'autre part la forme symétrique proposée par les chercheurs japonais SAKOE et CHIBA, où les dédoublement et les omissions de trames se font aussi bien sur l'axe du mot Référence que sur celui du mot Test .

Des discussions théoriques et considérations expérimentales [12] montrent la supériorité des contraintes symétriques .

Contrainte globale

La contrainte globale constitue un moyen très utile pour éliminer les mots qui sont trop courts ou trop longs que le mot prononcé , cela nous permet un gain en temps de calcul , donc des performances meilleures pour notre système .

Pour notre part , puisque la contrainte de pente retenue est $P = 1$, donc le mot test est comparé au mot référence si et seulement si sa longueur est comprise dans un intervalle dont les bornes sont respectivement la moitié et le double de la longueur du mot test .

Exemple de contraintes locales avec les coefficients de pondération associés

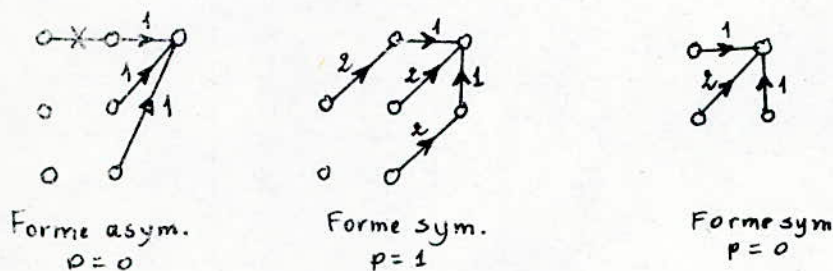
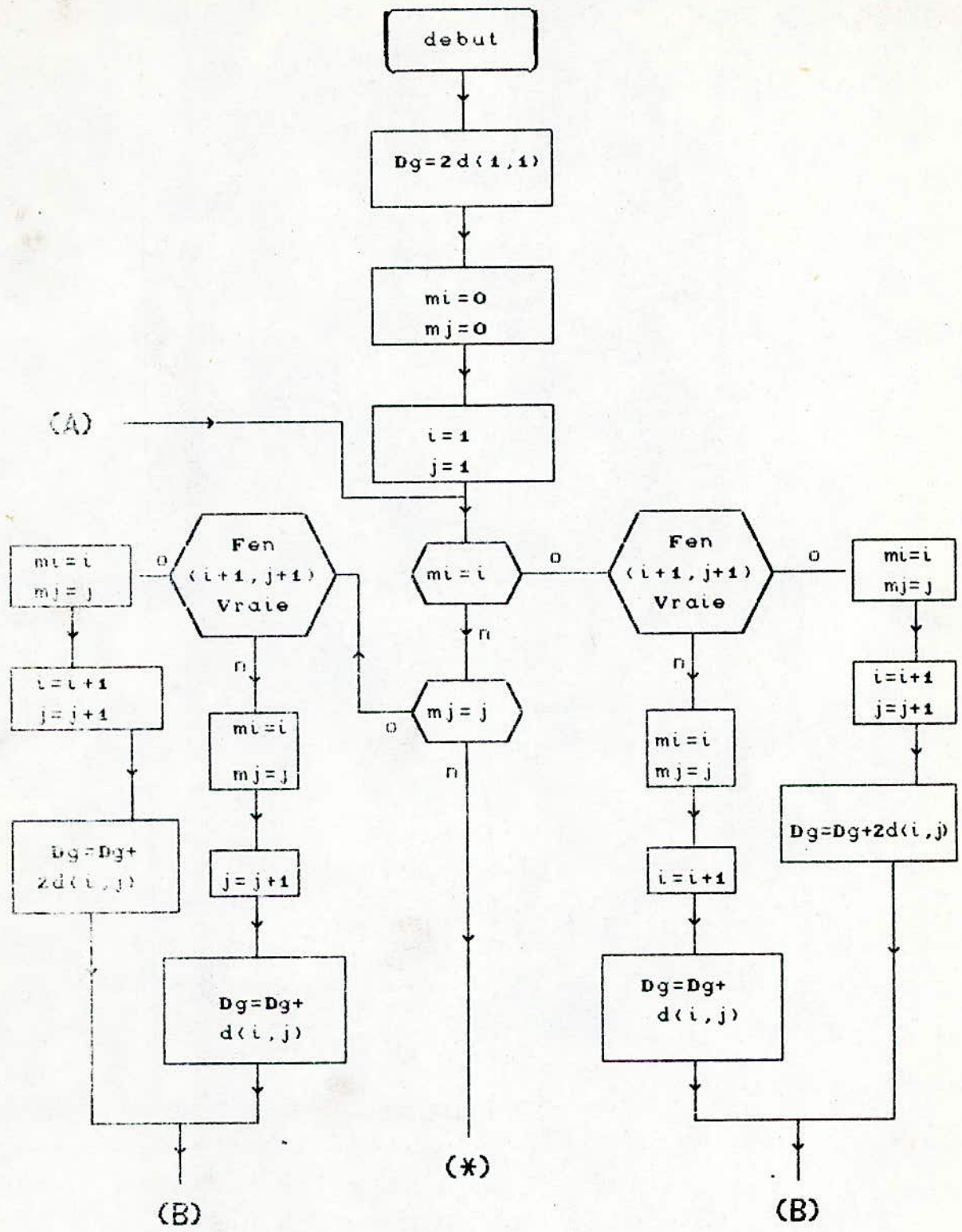


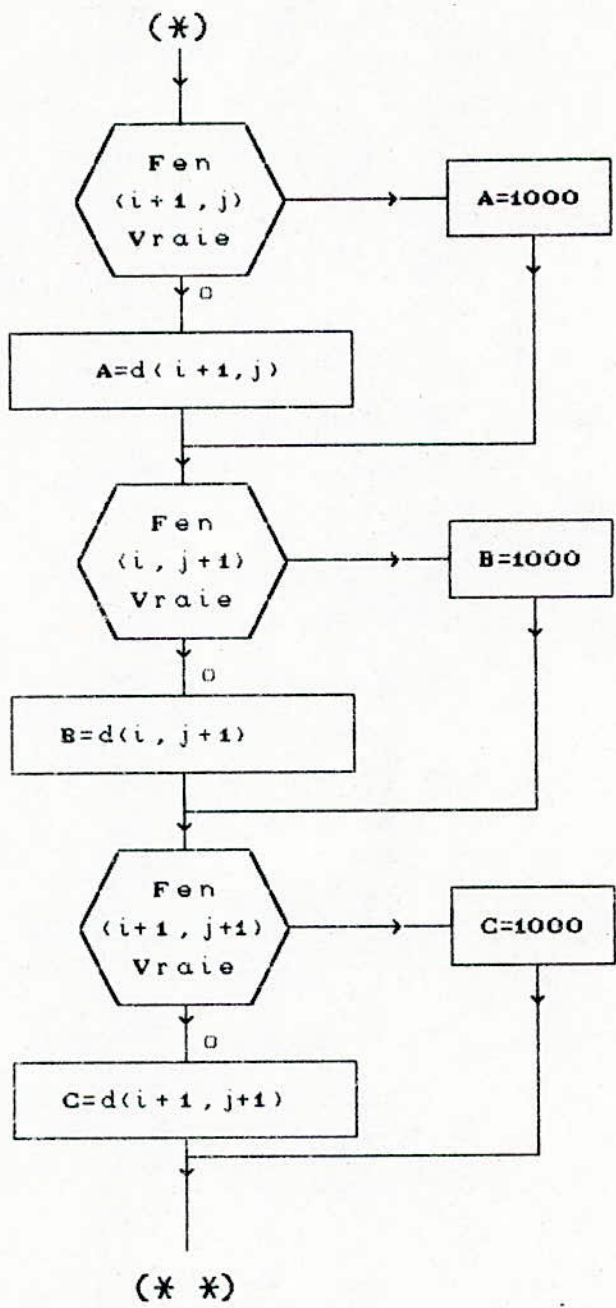
Fig (3-4)

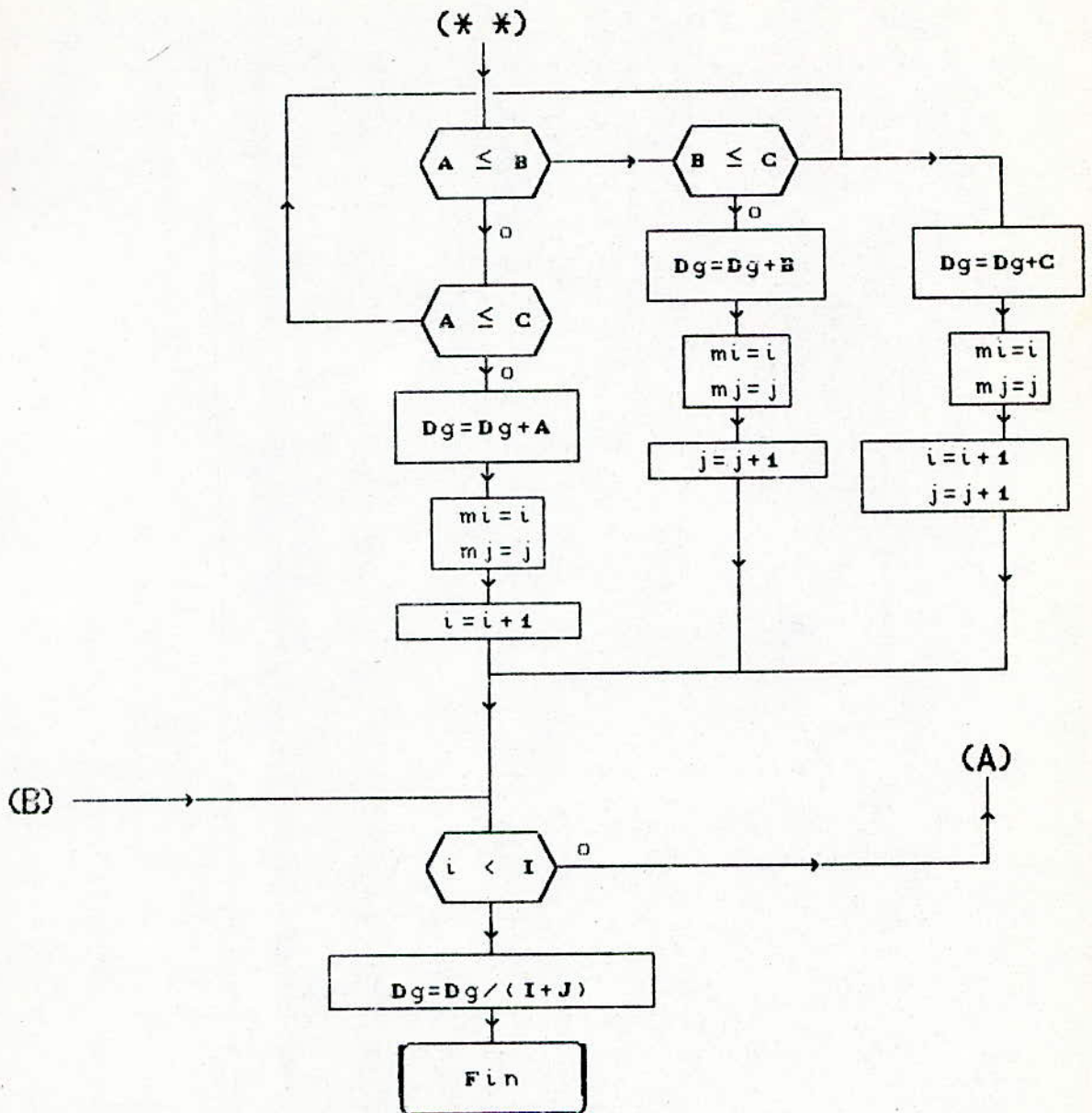
Pour notre travail nous avons choisi d'appliquer les contraintes locales de type symétrique avec une pente $p = 1$. Ce choix trouve sa justification dans le taux de reconnaissance meilleur obtenu avec ce type de contraintes [12] .

III)-6 ORGANIGRAMME

- Dg Dittance globale
- d(i,j) Fonction qui retourne entre la trame i et la trame j
- Fen(i,j) Est une fonction qui retourne la valeur vraie si le point $C(k) = \left[i(k), j(k) \right]$ est à l'intérieur du domaine défini par la contrainte globale
- mi et mj Sont des variables qui mémorisent les positions précédentes des variables i et j représentant les trames des mots référence et test

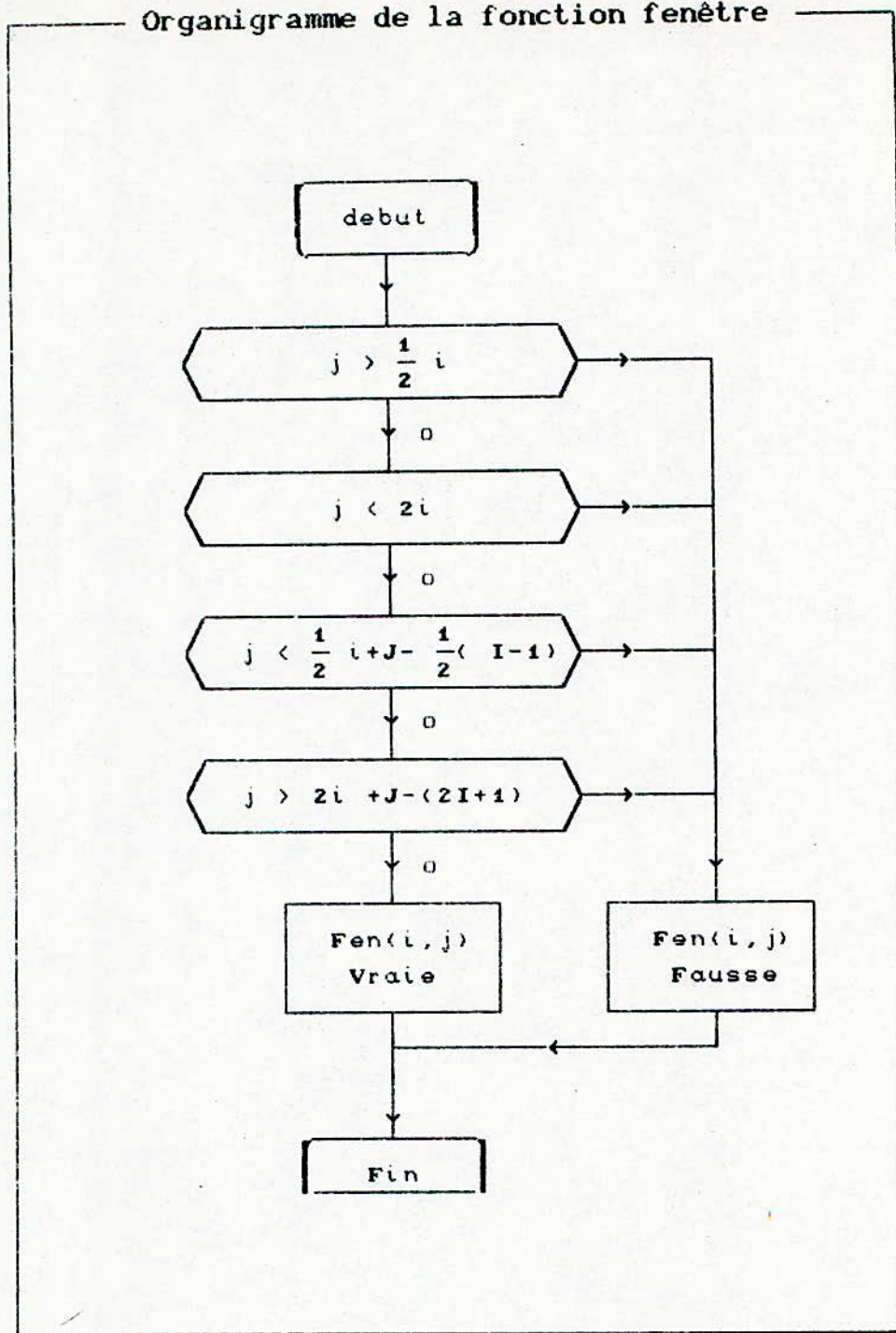




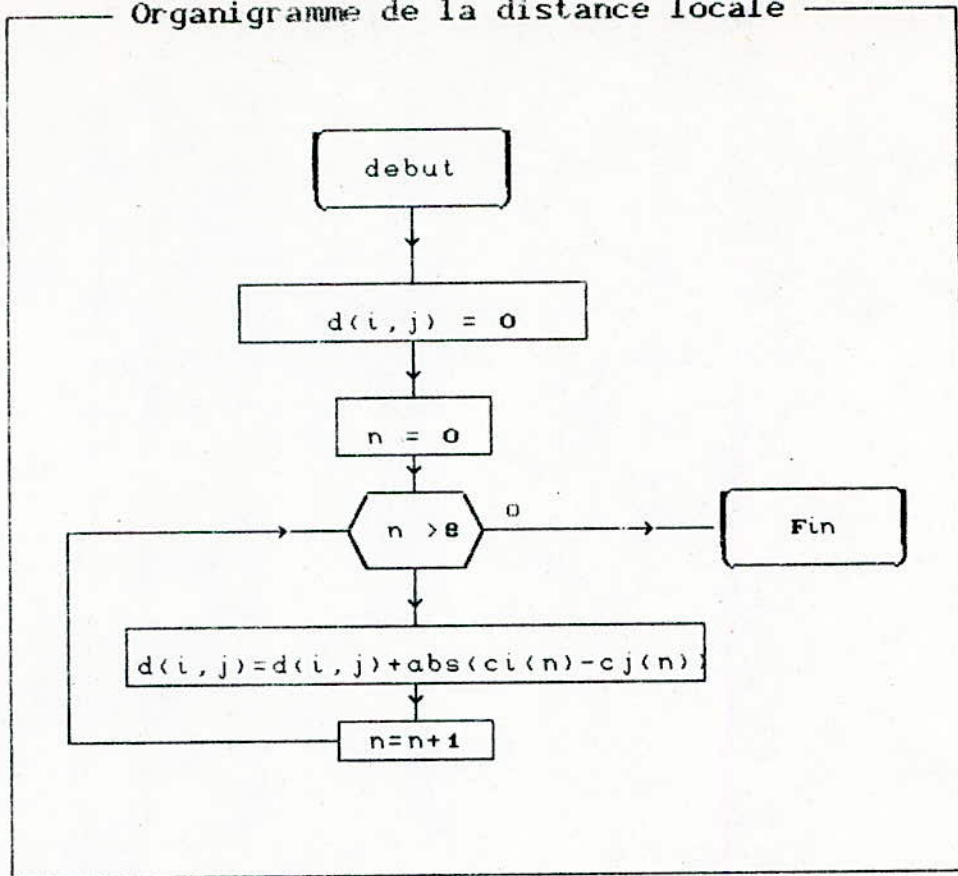


Organigramme de normalisation temporelle.

Organigramme de la fonction fenêtre



Organigramme de la distance locale



IV)-1 INTRODUCTION

La décision est la dernière étape dans le processus de reconnaissance de la parole .Elle consiste à prendre la décision du mot reconnu en exploitant les résultats issus de la comparaison dynamique des mots (distances globales) .

A la fin de la phase de comparaison , nous avons un ensemble de distances globales ordonnées suivant un ordre croissant .

Toutefois il peut arriver que cet ordre ne soit pas conforme à la réalité , c'est à dire que le mot qui devrait normalement être choisi ne se trouve pas à la première position et ceci peut arriver pour différentes raisons :

- mauvaise extraction des mots
- ressemblance phonétique entre deux mots du dictionnaire .
- etc

IV)-2 TECHNIQUE DES K. N. N

La technique des K.N.N (en anglais K Nearst Neighbours) est utilisée pour remédier au problème cité ci-dessus .L'idée est d'utiliser les "K" premières références d'un mot pour établir le classement des mots et non plus un classement référence par référence .Rappelons que les critères temps de réponse et les contraintes d'espace mémoire sont très importants pour les systèmes de reconnaissance automatique de la parole , aussi

toute conception doit veiller à optimiser ces deux paramètres . Dans notre système qui rappelle est multilocuteurs , chaque mot référence est représenté par plusieurs élocutions issues de la phase d'apprentissage . L'expérience montre que plus le nombre de références par mot augmente , mieux est le taux de reconnaissance , le nombre typique de représentants par mot est de 10 .

Dans notre système , la reconnaissance se fait en deux passages . Pendant le premier passage le mot Test est comparé avec un nombre réduit mais suffisant de références par mots , après ce passage on dégagera un certain nombre de candidats les plus probables .

Au deuxième passage , on comparera le mot Test avec les références restantes des mots issus du premier passage .

Cette astuce nous permet de réduire considérablement la quantité de traitement et donc le temps de réponse sans que la qualité du système ne subisse la moindre dégradation .

IV- 3 REJET

Certains mots font l'objet d'une décision de rejet de la part du système c'est à dire que le système ne répond pas à l'ordre lorsque la plus petite distance est plus grande qu'une distance seuil qu'on fixe à l'avance .

Cette procédure est utile et est même nécessaire . Elle a pour but d'écartier une éventuelle détection de bruit ou la prononciation d'un mot étranger c'est à dire n'appartenant pas

au dictionnaire de référence .

Il n'y a pas un moyen de prédire à l'avance la distance seuil acceptable ,sa détermination est une opération empirique qui se fait sur un grand nombre de données issues de comparaisons entre divers mots.

IV)- ORGANIGRAMME

Dans notre programme chaque voyelle est repéré par un numéro d'ordre de la façon suivante (A,1) , (O,2) , (I,3) , (E,4) , (U,5) , (Y,6) .

$R(n,k)$ est le $k^{\text{ème}}$ représentant de la voyelle n .

T est le mot Test .

Vect1 et Vect2 sont des vecteurs qui contiennent respectivement les distances globales moyennes après le premier et le deuxième passage .

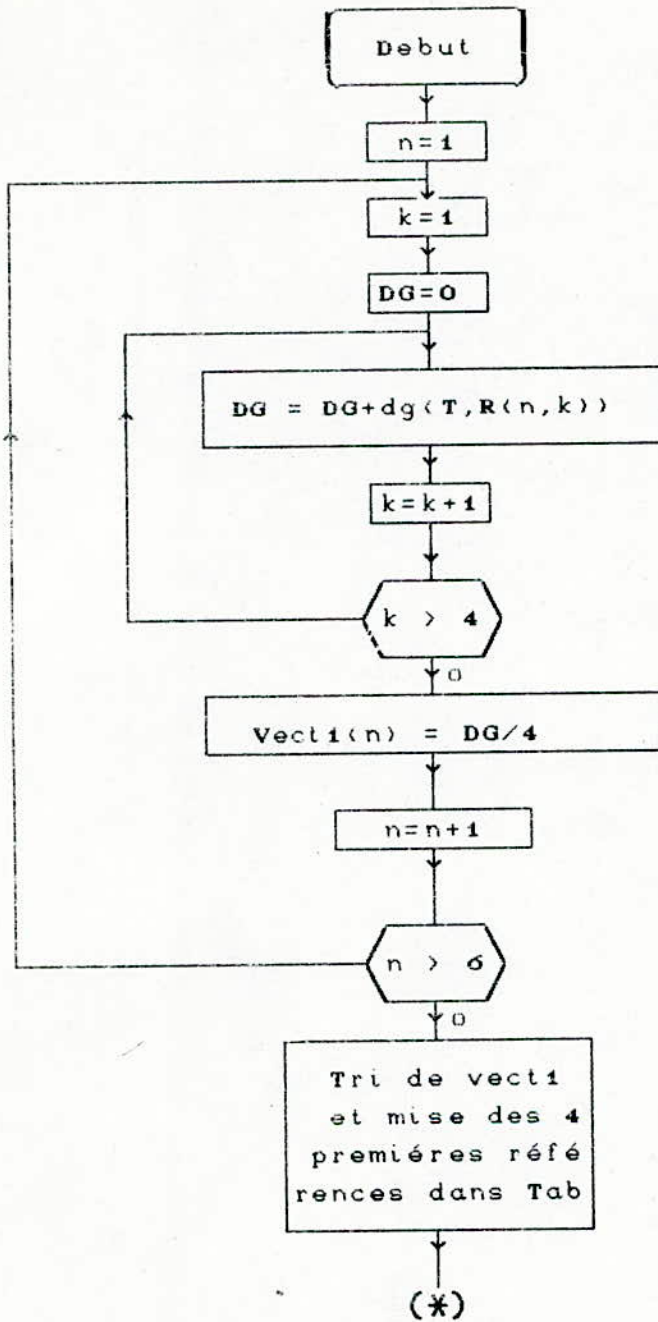
Tab est un tableau qui est formé après le premier passage ,il contient les numéros correspondants aux voyelles :

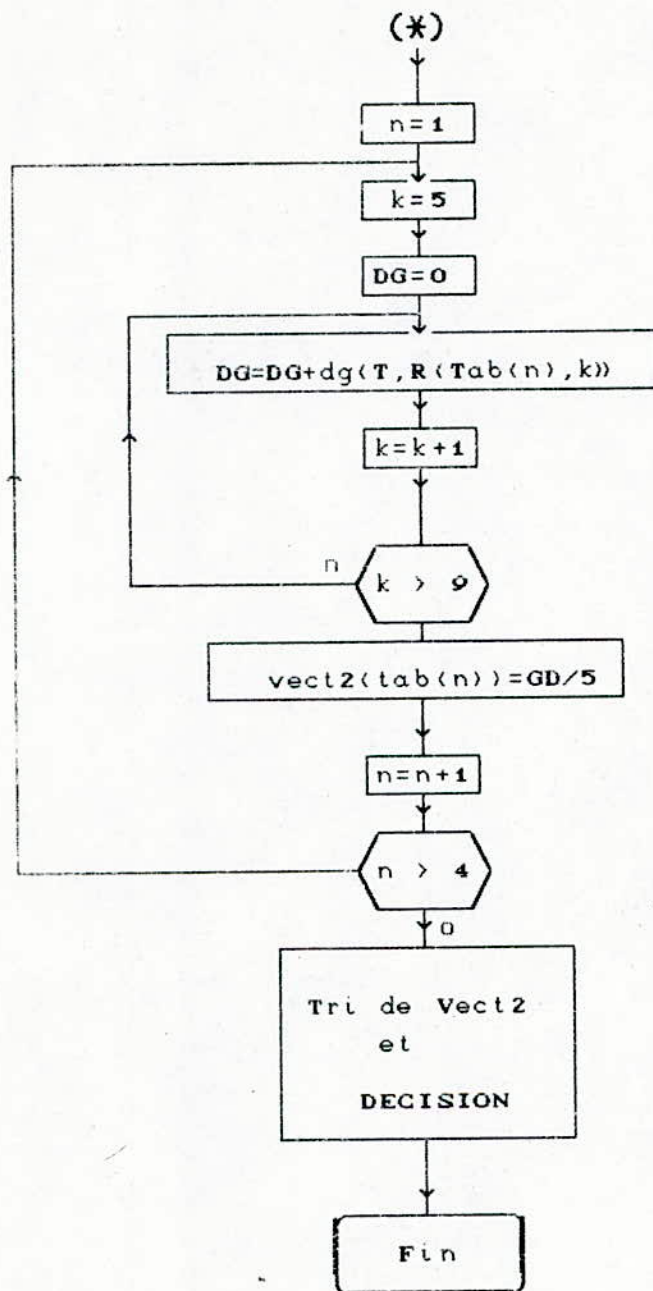
exemple ,si après le premier passage la voyelle O est classé première suivie par exemple par la voyelle U on aura $Tab[1]=2$ et $Tab[2]=5$

DG est la distance globale cumulée et dg la distance globale simple .

Le programme compare le mot Test avec les 4 premières références de chaque voyelle lors du premier passage ,les quatre meilleures voyelles sont ensuite reprises pour les comparer une nouvelle fois avec les cinq références restantes lors du second passage .

Organigramme Decision





VD-1 INTRODUCTION

La reconnaissance multilocuteur qui est un aspect de notre travail peut être atteinte suivant plusieurs méthodes dont on peut citer à titre d'exemple l'adaptation automatique aux nouveaux locuteurs. Cette idée est bonne mais sa mise en oeuvre est difficile et lourde. Il existe une autre façon d'aborder la question : c'est la multiplication des références pour un même mot.

Ces références devront être choisies de façon à intégrer au mieux les différences statistiques des élocutions d'un mot issues d'une large population d'individus.

Jusqu'à maintenant nous avons, tout au long de ce travail, employé le terme "mot Référence", nous avons dit que c'étaient des mots constituant le dictionnaire avec lesquels le mot test sera comparé, sans toutefois préciser comment ces mots ont été obtenus.

S'agit-il de locutions qu'on choisit au hasard ou alors d'une sorte de moyenne de locutions ?

En fait il ne s'agit ni de l'un ni de l'autre, les mots référence du dictionnaire sont en réalité le résultat d'un long et laborieux travail qui consiste en un grand nombre de comparaisons et de calculs qui se fait dans le cadre de ce que l'on appelle la classification automatique des données.

Celle-ci consiste à répartir un grand nombre de données brutes

(les différentes élocutions pour ce qui nous concerne) en classes plus au moins homogènes et à partir de ces classes , choisir les représentants suivant des critères pertinents . Nous avons admis l'existence de ces classes , il est cependant légitime de se demander si ces classes existent réellement . Cette hypothèse a été validée dans le cadre de l'Analyse Factorielle [7] appliquée au signal de la parole .

V)-2 SITUATION DU PROBLEME

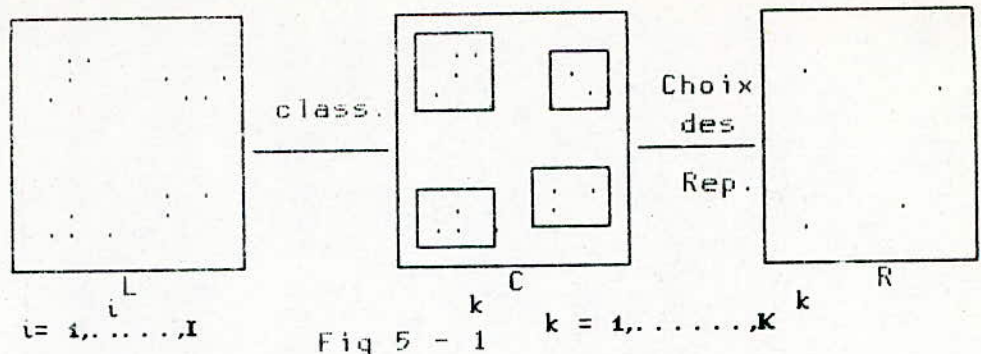
Soit donc $\{ L_i \}$ $i = 1, 2, \dots, I$

un ensemble assez grand de locutions d'un mot faites par une population de personnes .

A partir de cet ensemble de départ on utilisera un algorithme de classification qui nous donnera dans une première étape un ensemble de classes C_k dont le nombre peut être choisi arbitrairement .

Ce nombre est un compromis entre le taux de performance du système envisagé et les contraintes d'espace mémoire inhérentes à ce type de système .

Dans la deuxième étape , il s'agira ensuite de prendre dans chaque classe un représentant R_k qui rassemble les différences entre les locutions de cette même classe . Ce processus peut être schématisé de la manière suivante :



V)-3 ALGORITHME DE CLASSIFICATION

Il existe plusieurs algorithmes de classification basés sur différentes techniques :

Les techniques basées sur les méthodes hiérarchiques [6]

Les techniques basées sur les méthodes par partitionnement [7]

Ces dernières méthodes consistent à choisir un ensemble de classes de départ .Ce choix peut être tout à fait arbitraire , à affiner ces classes en transférant des éléments entre elles jusqu'à obtenir les classes les plus homogènes possibles.

La critique que l'on peut faire à cette méthode est la fixation à priori du nombre de classes K .

Cette technique nous permet certes de trouver les K classes optimales mais ne nous permet pas nécessairement de retrouver la répartition naturelle des locutions , où le nombre de classes serait bien entendu inconnu à priori .

Toutefois les résultats expérimentaux obtenus avec ces méthodes sont tout à fait satisfaisants [7] .

Avant d'aborder l'algorithme nous aurons besoin de quelques

outils mathématiques que nous définirons ci-après.

U)-3-1 Définitions

Appelons L l'ensemble $\{L_i\}$ $i = 1, \dots, I$
 ensemble qui contient toutes les prononciations d'un même mot.
 L'ensemble des classes C_k $k = 1, \dots, K$ forment une
 partition de L c'est à dire :

$$\begin{aligned} C_k &\neq \emptyset & \forall k = 1, \dots, K \\ C_k \cap C_l &= \emptyset & \forall k \neq l, k, l = 1, \dots, K \\ \bigcup_{k=1}^K C_k &= L \end{aligned}$$

Appelons également $d(L_i, L_k)$ la distance globale normalisée
 entre les prononciations L_i et L_k , et n_k le nombre d'éléments de
 la classe C_k

U)-3-1-1 Définition de la métrique

Soit une classe C_k .

Pour avoir une idée qualitative sur la situation d'un élément
 par rapport aux autres éléments d'une classe donnée C_k , on
 définit une métrique comme suit :

$$m(L_i, C_k) = \left(\frac{1}{n_k - 1} \sum_{\substack{k=1 \\ k \neq i}}^{n_k} d^2(L_i, C_k) \right)^{1/2}$$

Cette métrique représente la distance quadratique moyenne de
 L_i à tous les autres L_k , $L_k \in C_k$ et $L_k \neq L_i$

V)-3-1-2 Définition de la fonction d'homogénéité d'une classe

Cette fonction est destinée à nous donner une mesure de la qualité d'une classe, c'est à dire son degré d'homogénéité ou, en termes concrets, le degrés d'éparpillement des éléments entre eux dans la classe.

nous la définirons simplement par :

$$H(C_k) = \frac{1}{n_k} \sum_{i=1}^{n_k} m(L_i, C_k)$$

Nous pouvons remarquer que la fonction d'homogénéité $H(C_k)$ est d'autant plus petite que les éléments de la classe sont plus proches les uns des autres.

V)-3-1-3 Fonction critère

Cette fonction est destinée à mesurer la qualité relative

d'une partition $\{C_k\}$ $k = 1, \dots, K$

Pour notre part nous choisissons une fonction critère

$F\left\{\left\{C_k\right\}\right\}$ définie comme suit :

$$F = \sum_{k=1}^K H(C_k) (n_k - 1)$$

F étant par définition une somme de quantité positives. Sa diminution, donc la diminution des $H(C_k)$ entraîne la

formation de meilleures partitions et par voie de conséquence
la formation de meilleures classes

U)-3-2 Algorithme

- 1 Choix d'une partition initiale (choix arbitraire)
- 2 Trouver un élément L_i tel que son transfert d'une classe à une autre provoque la diminution de la fonction critère

$$F \left[\left\{ \left\{ C'_k \right\} \right\} \right] < F \left[\left\{ \left\{ C_k \right\} \right\} \right] \quad \text{sinon aller en } \underline{5}$$

- 3 Transférer cet élément L_i à une classe tel que la diminution de F est maximale
- 4 Aller en 2
- 5 Fin

V)-4 CHOIX DES REPRESENTANTS

Les classes étant prêtes , il s'agit de tirer de chaque classe un représentant en vue de l'intégrer dans le dictionnaire .

Ce représentant peut être choisi de deux manières :

-- soit choisir un représentant réel c'est à dire issu d'une locution faite par une personne .

Ce représentant devra répondre à un critère donné comme par exemple minimiser sa distance moyenne à tous les autres éléments .

-- soit un représentant 'moyen' qui n'a pas d'existence réelle

et qui est en quelque sorte , le centre de gravité de la classe .

L'expérience montre que les résultats obtenus avec le deuxième cas présentent une légère supériorité par rapport à ceux du premier cas .Cependant la somme de calcul supplémentaire qu'il apporte contrebalance son intérêt en reconnaissance .

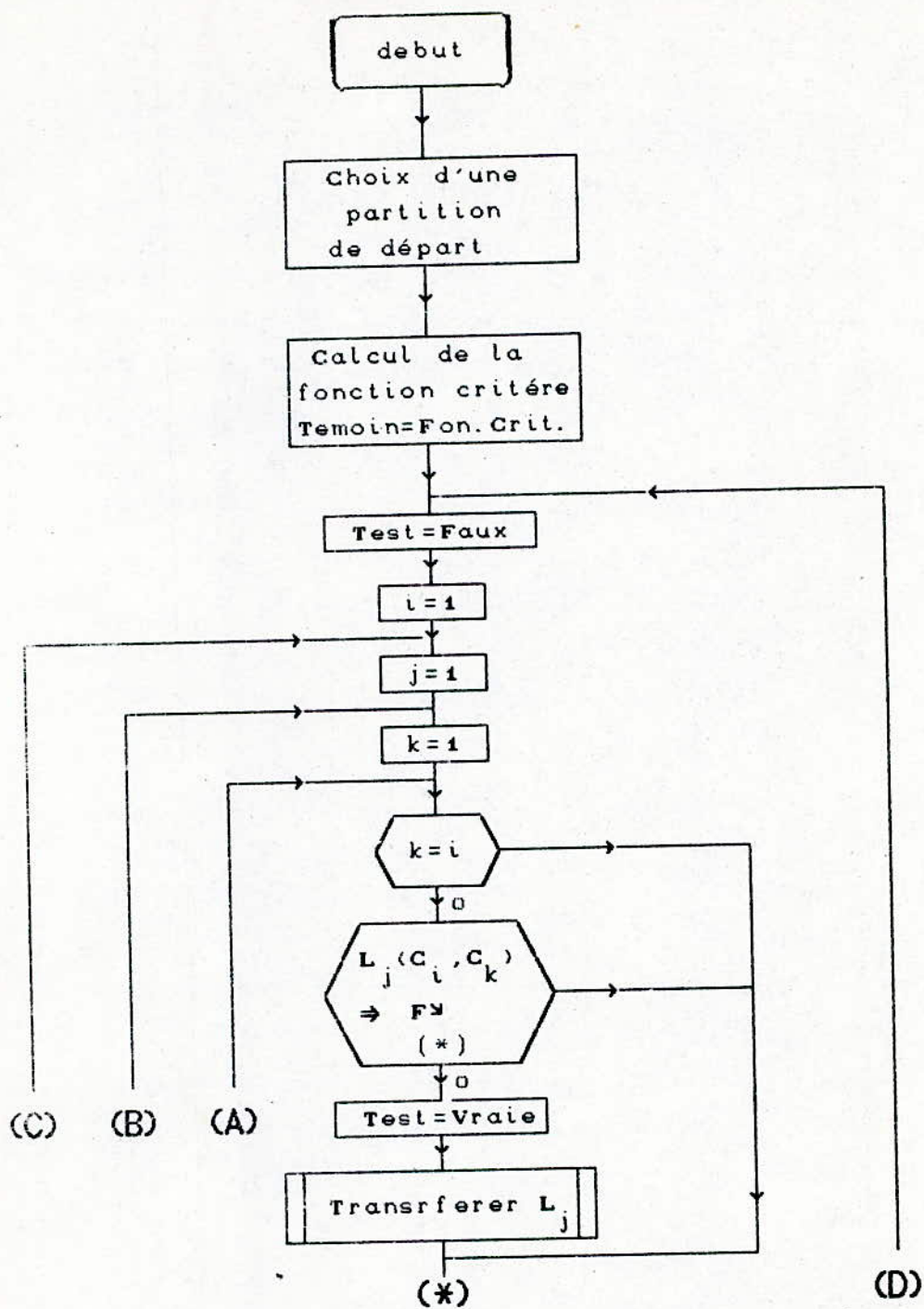
De plus ce paramètre est peu critique en matière de performance . C'est pour cette raison que nous avons choisi comme représentant d'une classe C_k l'élément R_k dont la distance moyenne à tous les autres éléments de la classe est minimum .

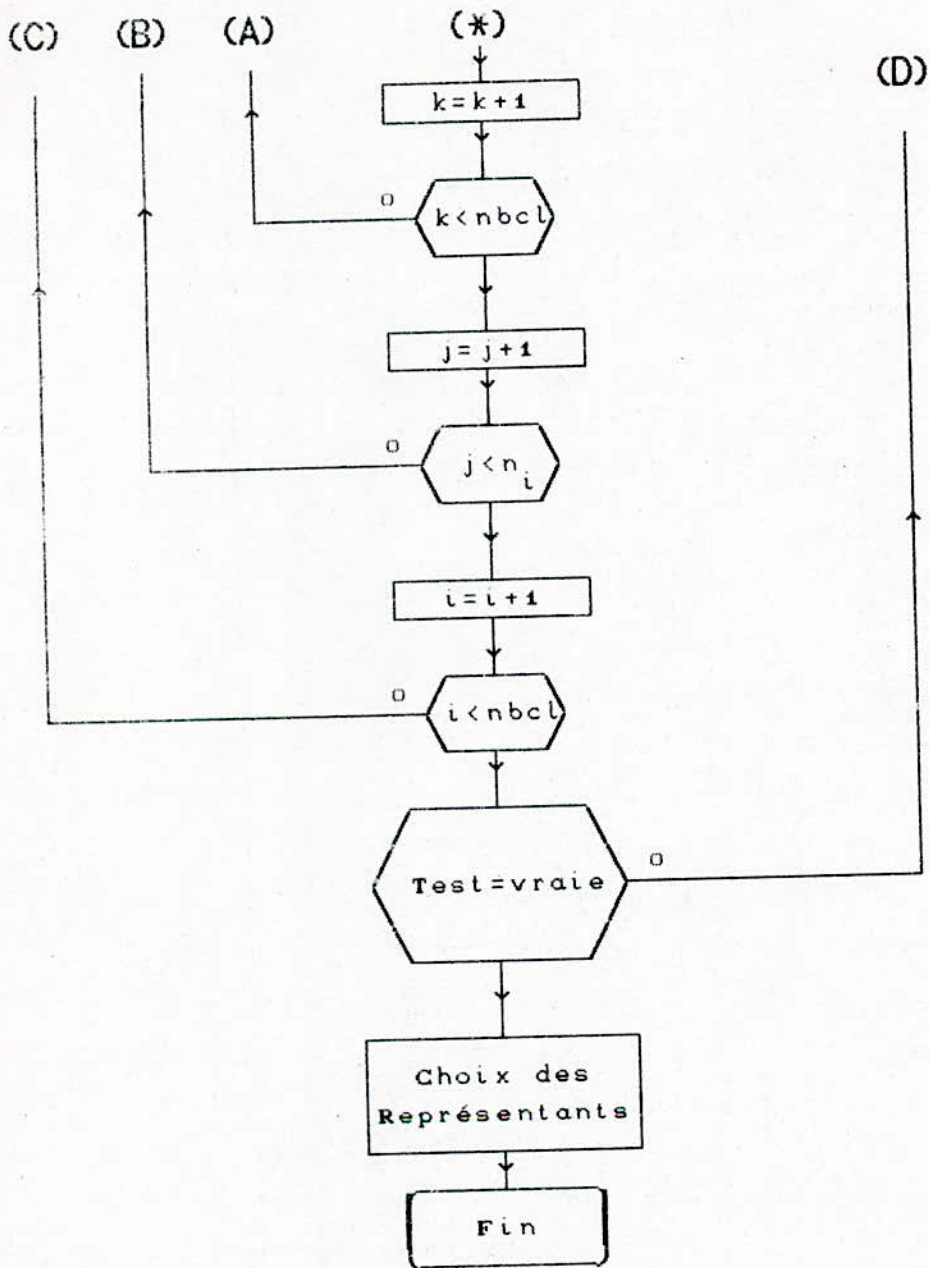
$$R_k / m(R_k, C_k) = \min_{L_i \in C_k} m(L_i, C_k)$$

VD-5 ORGANIGRAMME

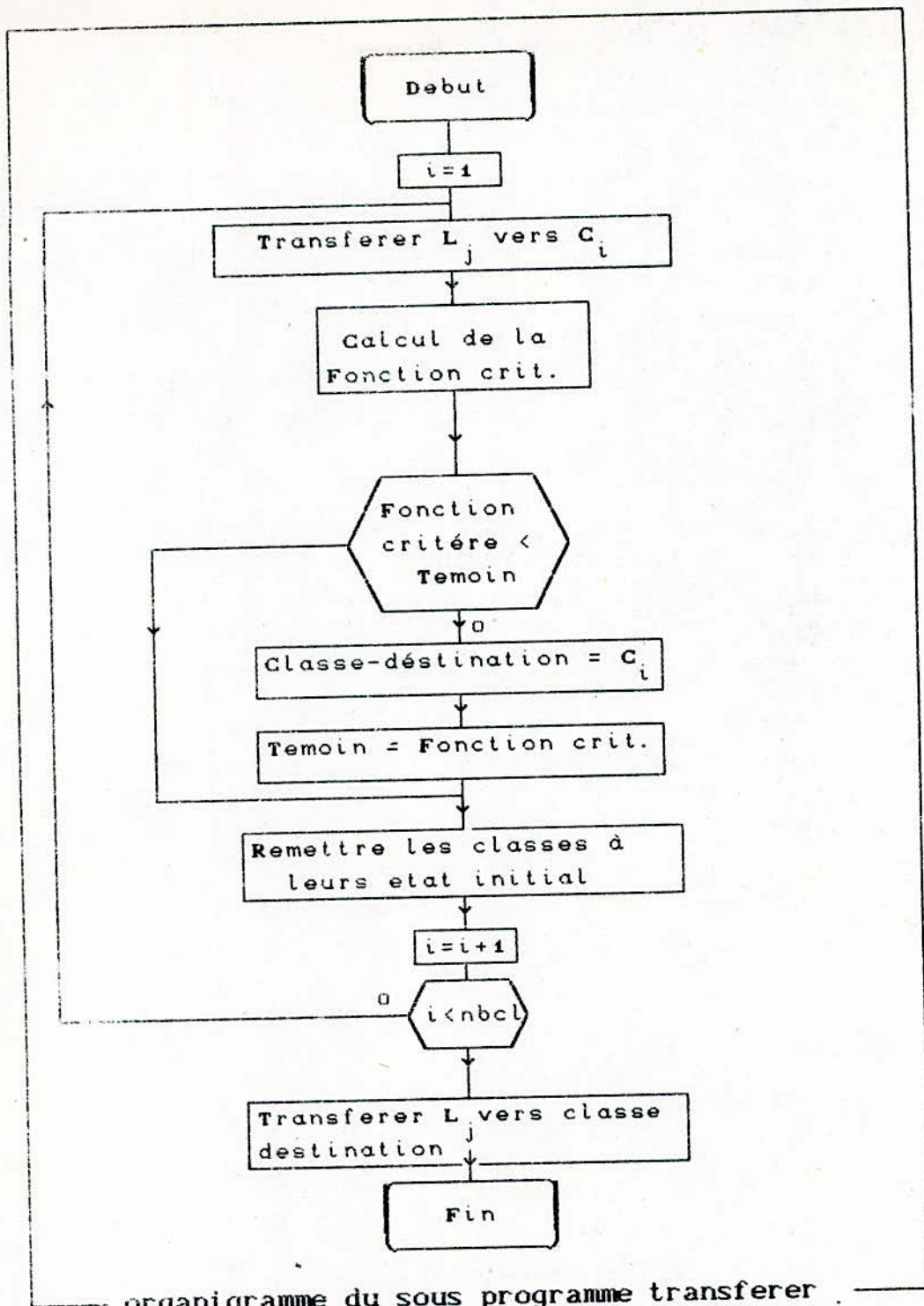
- n_i nombre d'éléments de la classe i
- n nombre de classe

(*) $L_j(C_i, C_k) \Rightarrow F_j$: le transfert de L_j de C_i vers C_k provoque t-il la diminution de la fonction critère (voir organigramme).





Organigramme Classification



organigramme du sous programme transferer

VI-1 INTRODUCTION

Le but de ce chapitre est d'établir un certain nombre de tests pour voir le comportement de nos différents programmes de reconnaissance .

pour ceci nous avons modélisé les voyelles voisées A O I E U Y en adoptant les formules mathématiques suivantes pour générer les échantillons temporels de ces voyelles :

$$\begin{aligned}
 V(n) = & A_v \sin(\text{coef } n F_1) + A_v/\sqrt{2} \sin(\text{coef } n (F_1+10)) \\
 & + A_v/\sqrt{2} \sin(\text{coef } n (F_1-10)) + A_v \sin(\text{coef } n F_2) \\
 & + A_v/\sqrt{2} \sin(\text{coef } n (F_2+10)) + A_v/\sqrt{2} \sin(\text{coef } n (F_2-10))
 \end{aligned}$$

V voyelle considérée

A_v amplitude associée

coef = $2\pi / \text{Nech}$

Nech nombre d'échantillons

F_1 et F_2 sont les deux premiers formants caractérisant la voyelle .

Ce modèle pourrait être amélioré (rapproché de la réalité) . Les voyelles étant voisées , il aurait fallu , pour générer les échantillons , en plus les convoluer avec les échantillons d'une autre sinusoïde pour simuler le pitch , chose qui n'a pas été faite en ce qui concerne notre travail .

VI)-2 TESTS EN MODE MONOLOGUEUR

L'ensemble de tests mis au point en mode monologueur a pour but de tester l'influence de divers paramètres sur les performances du système et aussi le comportement des différents modules de notre programme.

Les données nécessaires pour simuler les voyelles, à savoir F_1 , F_2 , Av et Nech, ont été tirées à partir de [5] pour les formants F_1 et F_2 .

En ce qui concerne l'amplitude et le nombre d'échantillons, en l'absence de documentation adéquate, nous avons fixé ces deux paramètres d'une manière arbitraire :

Voyelle	F_1	F_2	Nech	Av
A	750	1350	4096	70
O	375	750	3900	70
I	280	2500	5000	70
E	400	2200	2250	70
U	250	600	3584	70
Y	250	1800	3300	70

Tableau (6-1)

Les références du dictionnaire ont été calculées en utilisant les données de ce tableau par un programme spécialement mis au point, qu'on a appelé CREA_FCH.PAS. Les références (représentées par des matrices de coefficients cepstraux) ont

été stockées dans des fichiers .REF , exemple A.REF , O.REF

VI)-2-1 Influence de l'énergie

Le but de ce test est d'avoir une idée sur le module analyse acoustique ; en effet théoriquement les coefficients cepstraux sont normalisés en énergie.

Pour ceci nous avons maintenu les formants F_1 , F_2 ainsi que N_{ech} constants , et nous avons fait varier l'amplitude A_v .

Premier test : $A_v = 5$

R	T	A	O	I	E	U	Y
A		<u>0.0020</u>	2.2979	2.9261	2.5630	2.1653	2.9675
O		2.2975	<u>0.0018</u>	1.8656	1.8326	1.1109	2.9027
I		2.9256	1.8652	<u>0.0008</u>	long	1.6278	2.4327
E		2.5630	1.8363	court	<u>0.0067</u>	1.3865	3.7293
U		2.1649	1.1114	1.6277	1.3834	<u>0.0013</u>	2.6117
Y		2.9661	2.9026	2.4305	3.7232	2.6463	<u>0.0032</u>

Tableau 6-2

Taux de reconnaissance partiel $TR_1 = 100 \%$

deuxième Test : Av = 40

R	T	A	O	I	E	U	Y
A		<u>0.0006</u>	2.2976	2.8260	2.5630	2.1649	2.9660
O		2.2977	<u>0.0005</u>	1.8654	1.8877	1.1112	2.9029
I		2.9261	1.8655	<u>0.0002</u>	long	1.6277	2.4295
E		2.5631	1.8365	court	<u>0.0021</u>	1.3861	3.7286
U		2.1651	1.1110	1.6277	1.3877	<u>0.0004</u>	2.6461
Y		2.9665	2.9029	2.4301	3.7307	2.6464	<u>0.0022</u>

Tableau 6-3

Taux de reconnaissance partiel TR₂ = 100 %

TAUX DE RECONNAISSANCE SUR LES DEUX ESSAIS TR = 100 %

VI)-2-2 Déplacement des formants

Nous avons réalisé 2 tests en décalant à chaque test les formants F₁ et F₂ et en gardant les autres paramètres constants c'est à dire Nech et Av .

Test1	F ₁	F ₂	Test2	F ₁	F ₂
A	760	1360	A	754	1352
O	365	740	O	382	757
I	255	2509	I	258	2500
E	450	2230	E	400	2202
U	240	590	U	255	609
Y	260	1800	Y	260	1806

Resultats avec test1 :

R	T	A	O	I	E	U	Y
A		<u>0.841</u>	2.247	2.677	2.349	2.036	2.828
O		2.116	<u>0.608</u>	2.763	1.833	1.165	2.798
I		2.684	1.828	1.935	long	1.695	2.261
E		2.405	1.828	court	<u>1.119</u>	1.340	3.448
U		1.923	1.089	2.520	1.582	<u>0.473</u>	2.562
Y		3.020	2.819	<u>1.625</u>	3.598	2.808	<u>0.547</u>

Tableau 6-4

Taux de reconnaissance partiel $TR_1 = 83.4 \%$

Resultats avec test2

R	T	A	O	I	E	U	Y
A		<u>0.898</u>	2.208	2.925	2.523	2.267	2.785
O		2.103	<u>0.704</u>	1.816	1.718	1.118	2.803
I		2.560	1.821	<u>0.540</u>	long	1.668	2.254
E		2.308	1.878	court	<u>0.423</u>	1.550	3.617
U		1.923	1.277	1.729	1.298	<u>0.569</u>	2.556
Y		3.098	2.691	2.376	3.617	2.745	<u>0.936</u>

Tableau 6-5

Taux de reconnaissance partiel $TR_2 = 100 \%$

TAUX DE RECONNAISSANCE SUR LES DEUX ESSAIS $TR = 100 \%$

VI)-2-3 Influence de la longueur des mots

Ce test a été élaboré pour voir comment le module normalisation temporelle se comporte vis-à-vis des variabilités temporelles. Dans ce but nous avons créé des fichiers tests en gardant les formants F_1 et F_2 constants ainsi que l'amplitude A_v c'est-à-dire celles affichées dans le tableau 6-1, mais en faisant varier à chaque fois le nombre d'échantillons N_{ech} .

On rapporte dans ce qui suit les résultats obtenus avec 2 essais $Test_1$ et $Test_2$:

$Test_1$	N_{ech}	$Test_2$	N_{ech}
A	4500	A	4097
O	3600	O	3130
I	4790	I	5280
E	2900	I	2455
U	4000	U	3647
Y	3000	Y	3400

Resultats avec test1 :

R	T	A	O	I	E	U	Y
A		<u>1.553</u>	2.572	2.801	2.936	2.381	2.607
O		2.402	<u>0.735</u>	2.666	<u>1.976</u>	1.133	3.015
I		2.975	1.969	2.078	2.726	1.701	2.383
E		2.499	1.877	court	2.350	1.724	3.902
U		2.302	1.464	2.580	2.342	<u>1.108</u>	2.983
Y		3.036	3.014	<u>1.348</u>	3.537	2.779	<u>1.931</u>

Tableau 6-6

Taux de reconnaissance partiel TR₁ = 66.7 %

Resultats avec test2

R	T	A	O	I	E	U	Y
A		<u>0.772</u>	2.336	2.820	2.986	2.333	3.053
O		2.205	<u>0.789</u>	2.837	<u>1.703</u>	1.164	2.660
I		2.698	1.271	2.142	2.443	1.642	2.168
E		2.466	1.818	long	1.979	1.765	3.478
U		1.956	1.106	2.698	1.762	<u>0.814</u>	2.633
Y		2.919	3.025	<u>1.347</u>	3.768	2.905	<u>1.098</u>

Tableau 6-7

Taux de reconnaissance partiel TR₂ = 66.7 %

TAUX DE RECONNAISSANCE SUR LES DEUX ESSAIS TR 66.7 %

Interpretation

Le taux de reconnaissance qu'on juge très faible au vu de la dimension du dictionnaire ne peut pas à notre sens constituer un indice de faiblesse du module de la comparaison dynamique, en ce sens que les voyelles I et Y qui sont confondues sont proches l'une de l'autre (se referer aux données du tableau 6-1) .Les formants de la voyelle E et ceux de U sont assez éloignés ,mais il ne faut pas oublier l'effet de l'échantillonnage qui n'a pas été réalisé avec la même fréquence pour toutes les voyelles étudiées

VI)-2-4 Influence des paramètres decallage des formants et longueur des mots combinés :

Dans la réalité les paramètres cités précédemment sont toujours combinés simultanément , c'est pour cette raison qu'on a mis au point les deux tests qui suivent .

Cependant l'influence de l'énergie était comme on l'a vu paragraphe VI)-2-1, négligeable .On n'en n'a donc pas tenu compte pour les tests qui suivent.

Les fichiers tests créés sont les suivants :

Test1	F1	F2	Nech	Test2	F1	F2	Nech
A	755	1354	4600	A	755	1362	4228
O	380	770	4100	O	387	756	4218
I	241	2489	4800	I	251	2506	5091
E	408	2210	3000	E	401	2201	2643
U	240	590	4000	U	250	601	3681
Y	255	1809	3000	Y	255	1800	3653

Resultats avec test1

R	T	A	O	I	E	U	Y
A		<u>1.573</u>	2.309	2.946	2.144	2.552	2.531
O		2.393	<u>0.698</u>	2.868	<u>2.496</u>	1.312	2.950
I		2.747	1.803	2.247	2.922	1.643	2.360
E		2.384	2.029	court	2.805	1.636	3.756
U		2.345	1.152	2.716	2.720	<u>1.092</u>	3.047
Y		3.016	2.835	<u>1.469</u>	3.412	2.942	<u>2.029</u>

Tableau 6-8

Taux de reconnaissance partiel $TR_1 = 66.7 \%$

Resultats avec test2

R	T	A	O	I	E	U	Y
A		<u>0.881</u>	2.275	2.947	2.455	2.254	3.146
O		2.235	<u>0.810</u>	2.863	1.813	1.028	2.858
I		2.782	1.651	2.318	3.039	1.718	2.252
E		2.425	1.620	court	<u>1.584</u>	1.712	3.487
U		2.056	1.684	2.719	1.888	<u>0.872</u>	2.879
Y		2.857	2.745	<u>1.613</u>	4.182	2.827	<u>1.566</u>

Tableau 6-9

Taux de reconnaissance partiel $TR_2 = 83.3 \%$

TAUX DE RECONNAISSANCE SUR LES DEUX ESSAIS $TR = 75 \%$

VI)-2-5 essais de quelques mots étrangers

Cet essais a été mis au point pour voir le comportement de notre programme en face d'un mots ne figurant pas dans le dictionnaire .

La réponce normale devrait ,bien entendu, etre le rejet en appliquant la contrainte de distance seuil .

On a donc simulé 2 mots hypothétique de la façon suivante :

Mot M ₁			Mot M ₂		
F ₁	F ₂	Nech	F ₁	F ₂	Nech
1000	2000	3600	1200	1600	3900

et on a obtenu les résultats suivants :

Mot test M₁

Référence	Distance
A	3.03744
O	3.16816
I	3.02060
E	3.34620
U	3.15914
Y	2.85211

Distance minimale
2.52811

Mot test M₂

Référence	Distance
A	2.25866
O	2.78583
I	2.91966
E	3.34784
U	2.71715
Y	2.69650

Distance minimale
2.25866

Interpretation

Les distances globales qui ressortent de la comparaison d'un mot étranger aux voyelles de référence sont sensiblement plus grandes que celles issues d'une comparaison avec les voyelles appartenant au dictionnaire , cependant les distances minimales demeurent encore faibles .Ceci peut être expliqué par le modèle adopté pour représenter les mots étrangers qui a la même forme que les voyelles de référence .

VI)-3 TESTS EN MODE MULTILOCUTEUR

Les tests en mode multilocuteur ont porté seulement sur l'influence du nombre de représentants par voyelle sur le taux de reconnaissance .

Pour mettre en oeuvre ce test nous avons procédé à la création de 9 représentants pour chaque voyelle , que nous avons stocké sur disque sous forme de fichiers .REF ,exemple , pour la voyelle A les neufs représentants sont : A₁.REF , A₂.REF , . . . , A₉.REF Nous avons ensuite créé des fichiers tests ,5 en tout que nous avons appelé Test10 , Test11 , . . . ,Test14 Les fichiers tests ont été créé d'une manière aléatoire :
exemple , pour créer la voyelle A du fichier Test10 c'est-à-dire Test10_A nous avons utilisé dans le programme les formules :

Pour les formants $F_1 = 750 + \text{Random}(20)$

$F_2 = 1350 + \text{Random}(20)$

Pour le nombre d'échantillons $N_{ech} = 4096 + \text{Random}(400)$

Où Random(Num) est une fonction du langage T.Pascal qui génère des nombres aléatoires inférieurs à Num .

Les fichiers tests ainsi obtenus sont les suivants :

Test10	F1	F2	Nech	Test11	F1	F2	Nech
A	765	1362	4288	A	768	1369	4131
O	378	753	3936	O	375	751	4075
I	251	2503	5167	I	267	2511	5092
E	416	2215	2292	E	421	2211	2363
U	258	611	3603	U	262	622	3815
Y	268	1805	3344	Y	275	1811	3635

Test12	F1	F2	Nech	Test13	F1	F2	Nech
A	786	1386	4224	A	773	1374	4231
O	376	788	4180	O	381	764	4146
I	259	2517	4972	I	282	2538	5069
E	433	2213	2473	E	426	2219	2330
U	281	629	3646	U	251	636	3790
Y	252	1821	3334	Y	252	1800	3423

Test14	F1	F2	Nech
A	753	1363	4337
O	389	779	4204
I	285	2517	5161
E	402	2201	2570
U	277	624	3790
Y	259	1813	3637

Influence du nombre de représentants

L'influence du nombre de représentants a été étudié à l'aide du fichier Test14 .

Resultats avec 4 représentants en référence

Premier passage :

R	A	O	I	E	U	Y
A	<u>1.172</u>	<u>2.130</u>	2.912	<u>2.294</u>	<u>2.174</u>	2.957
O	<u>2.404</u>	<u>0.685</u>	<u>2.816</u>	<u>1.778</u>	<u>1.049</u>	<u>2.392</u>
I	2.932	2.280	<u>1.255</u>	9999	2.485	<u>1.419</u>
E	<u>2.157</u>	<u>1.471</u>	3.127	<u>1.156</u>	<u>1.454</u>	3.160
U	<u>2.207</u>	<u>0.966</u>	<u>2.819</u>	<u>1.677</u>	<u>0.803</u>	<u>2.594</u>
Y	3.067	2.515	<u>1.272</u>	3.316	2.816	<u>1.032</u>

Deuxieme passage :

test14	A	O	I	E	U	Y
A	1.587	0.666	0.712	1.587	0.778	1.064
E	2.024	0.803	1.133	1.737	0.944	1.067
U	2.136	1.532	2.855	1.800	1.454	2.481
O	2.235	2.112	2.803	2.235	2.095	2.544

Resultats avec 6 representants :

Premier passage

R	T	A	O	I	E	U	Y
A	<u>1.146</u>	<u>2.137</u>	2.907	<u>2.283</u>	<u>2.178</u>	2.981	
O	<u>2.411</u>	<u>0.692</u>	<u>2.849</u>	<u>1.791</u>	<u>1.022</u>	<u>2.451</u>	
I	2.933	2.373	<u>1.116</u>	9999	2.580	<u>1.317</u>	
E	<u>2.147</u>	<u>1.533</u>	3.127	<u>1.325</u>	<u>1.479</u>	3.200	
U	<u>2.165</u>	<u>0.945</u>	<u>2.786</u>	<u>1.663</u>	<u>0.792</u>	<u>2.551</u>	
Y	3.056	2.506	<u>1.270</u>	3.348	2.801	<u>1.099</u>	

Deuxieme passage

test14	A	O	I	E	U	Y
A	1.095	0.677	0.688	1.911	0.803	1.028
E	2.262	1.044	1.170	1.663	0.953	1.062
U	2.123	1.604	2.768	1.728	1.419	2.436
O	2.388	2.189	2.859	2.257	2.223	2.498

TAUX DE RECONNAISSANCE TR = 83.3 %

Resultats avec 9 representants

Premier passage

R	T	A	O	I	E	U	Y
A	<u>1.114</u>	<u>2.163</u>	2.907	<u>2.469</u>	<u>2.200</u>	2.981	
O	<u>2.374</u>	<u>0.655</u>	<u>2.845</u>	<u>1.591</u>	<u>0.998</u>	<u>2.440</u>	
I	2.933	2.373	<u>1.037</u>	9999	2.580	<u>1.301</u>	
E	<u>2.180</u>	<u>1.486</u>	3.127	<u>1.107</u>	<u>1.416</u>	3.200	
U	<u>2.155</u>	<u>0.967</u>	<u>2.767</u>	<u>1.446</u>	<u>0.786</u>	<u>2.537</u>	
Y	3.056	2.506	<u>1.253</u>	3.348	2.801	<u>1.066</u>	

Deuxieme passage

test14	A	O	I	E	U	Y
A	1.093	0.648	0.663	1.357	0.770	0.927
E	2.164	1.039	1.118	1.439	1.087	1.049
U	2.335	1.576	2.854	1.742	1.442	2.311
O	2.413	2.213	2.782	2.361	2.259	2.593

TAUX DE RECONNAISSANCE TR = 100 %

Interprétation des résultats :

Le taux de reconnaissance a été dans tous les cas 100% sauf dans le test avec 6 représentants où la voyelle E a été confondue avec la voyelle U , ceci peut être expliqué par la présence d'une ou plusieurs mauvaises références de la voyelle E dans le dictionnaire

VI)-4 CONCLUSION

Le taux de reconnaissance obtenu à travers les différents tests que nous avons mis au point peuvent être jugés satisfaisants si l'on considère le modèle mathématique retenu pour générer les échantillons des voyelles A O I E U Y . En effet la forme générale du modèle est la même pour toutes les voyelles simulées , seuls les formants F_1 , F_2 et le nombre d'échantillons changent . On comprend donc que le taux est faible dans certains cas .

De toute façon pour tirer des conclusions objectives , il aurait fallu faire beaucoup de tests avec chacun des paramètres , parcequ'en reconnaissance seules les données statistiques (donc beaucoup de données) ont une signification valable .

Nous avons d'autre part remarqué que le nombre de coefficients cepstraux utilisés ainsi que leur rang n'avaient pratiquement aucune influence , contrairement à ce que la théorie prévoit . Les résultats ne mettent toutefois pas en cause d'une manière décisive l'insuffisance des algorithmes que nous avons choisi

serait en effet inutile de développer des algorithmes puissants d'analyse et de normalisation si le module détection S / B ne suit pas .

L'une des conclusions les plus importantes qu'on peut tirer au terme de ce travail est qu'un sujet comme celui-ci exige un minimum de moyens pour qu'on puisse arriver à un résultat quelconque .La reconnaissance de la parole est un domaine essentiellement ,pour ne pas dire totalement ,expérimental .La procédure que les chercheurs emploient dans ce domaine est d'émettre des hypothèses et de procéder immédiatement à l'expérience pour confirmation ou infirmation en l'absence presque totale de données théoriques relatives à la parole .

Les résultats obtenus à travers les quelques tests (insuffisants) ont été rapportés dans le chapitre VI) , ce qu'il faut dire c'est qu'il ne faut pas trop se fier à ces résultats , la valeur de notre travail ne peut, à notre avis , être jugée positivement ou négativement qu'à travers des tests réels .c'est à dire ,implanter les programmes mis au point ici ,sur une carte autonome et voir le comportement du système dans des conditions réelles de fonctionnement .

En effet une simulation du signal de la parole ne peut prendre en compte toutes les subtilités et toute la dynamique naturelle qui caractérisent le signal acoustique .

LISTING DES PROGRAMMES
DE
RECONNAISSANCE

```

(*****)
(* projet de fin d'etude : *)
(* RECONNAISSANCE DE LA PAROLE *)
(* EN MODE MULTILOCUTEUR *)
(* realise par : *)
(* BENACHOUR ABDELAZIZ *)
(* Ecole Nationale Polytechnique *)
(* Juin 1990 *)
(*****)

```

```
PROGRAM RECONNAISSANCE ;
```

```
uses crt,graph;
```

```
CONST
```

```

nmff =16;
seuil =1e-4;
x1=200;y1=260;
lx=300;ly=150;
radius =3;
nTmax =25;
npT =256;
lettres='AOIEUY';

```

```
type
```

```

liste =Array[1..nmff] of REAL;
Lliste=Array[1..ntmax] of liste;
Vect1 =Array[1..6] of INTEGER;
Vect2 =Array[1..4] of REAL;
ref =File of liste;
strin =STRING[2];

```

```
VAR
```

```

tabT,tabR:Lliste;
FichierRef:ref;
deb,pai,paj,fin,i,j,k,ntT,ntR,Long_Court,nb_cl:INTEGER;
Dist_Glo:REAL;
Stri:STRING[20];
Con:liste;
Dist1:Array[1..6] of REAL;
Dist2:Vect2;
T =Vect1;
Ch,Mot,Ch1,Mt:STRING;
c:char;

```

```

FUNCTION FENETRE(i,j:INTEGER):boolean ;
LABEL 1,2;

```

```

BEGIN
  if j<0.5*i then goto 1;
  if j>2*i then goto 1;
  if j>0.5*i+ntR-0.5*(ntT-1) then goto 1;
  if j<2*i+ntR-(2*ntT+1) then goto 1;
  FENETRE:=true;goto 2;
1: FENETRE:=false;
2: END;

```

```

PROCEDURE GRAPHE(ch:Strin);
  VAR i,j,x,y:INTEGER;

```

```

BEGIN          (*  debut PROCEDURE graphe  *)
  cleardevice;
  moveTo(x1,y1-ly);lineTo(x1,y1);lineTo(x1+lx,y1);
  pai:=trunc(lx/ntt);paj:=trunc(ly/ntr);
  FOR i:=1 TO ntt do
  BEGIN
    x:=x1+i*pai;line(x,y1+2,x,y1-2)
  END;
  FOR j:=1 TO ntr do
  BEGIN
    y:=y1-j*paj;line(x1+2,y,x1-2,y)
  END;
  FOR i:=1 TO ntt do BEGIN
    x:=x1+i*pai;
    FOR j:=1 TO ntr do BEGIN
      y:=y1-j*paj;
      if FENETRE(i,j) then circle(x,y,radius)
      else putpixel(x,y,white);
    END;
  END;
  setTextStyle(defaultFont,VertDir,1);
  SetTextJustify(leftText,centerText);
  outTextXY(x1-2,(2*y1-ly) div 2,'mot ref. ');
  SetTextStyle(defaultFont,HorizDir,1);
  SetTextJustify(centerText,TopText);
  OutTextXY((2*x1+lx) div 2,y1+2,'mot test ');
  setTextStyle(gothicfont,horizdir,0);
  SetTextJustify(leftText,TopText);
  OutTextXY(x1,10,'normalisation temporelle ');
  settextstyle(smallfont,horizdir,0);
  settextjustify(centerText,centerText);
  outTextXY(getmaxX div 2,getmaxY-5,'...appuyer sur enter... ');

```



```

settextstyle(defaultfont,horizdir,1);
settextjustify(lefttext,toptext);
outtextXY(0,150,'resultats');
outtextXY(0,130,'lettre reference: '+ch);
outtextXY(0,110,'Lettre Test      : '+ch1);
END;

```

PROCEDURE SAISIE_DONNEE ;

```

BEGIN
ClrScr;
GoToXY(32,4);TextColor(White+Blink);
WriteLn('... SAISIE ...');
GoToXY(1,10);
NormVideo;
WriteLn('Choisir un Fichier Test [TEST_X.DAT]  X = A,O,I,E,U,Y = ');
ReadLn(Stri); Stri:=Stri+'.DAT';
Ch1:= Copy(Stri,7,1);
GoToXY(1,15);
WriteLn('Entrez les bornes des coefts cepstraux deb et fin');
WriteLn('NB : Choisir des chiffres entre 1 et 16 ');
WriteLn;WriteLn;
WriteLn('DEB                                FIN');
Read(deb);GoToXY(whereX+38,whereY-1);ReadLn(fin);
GoToXY(1,25);
Write('Donner le nombre de representants par voyelle :');
ReadLn(nb_cl);
ClrScr;
END;

```

```

{*****}
{*                                *}
{**          ANALYSE ACOUSTIQUE    **}
{*                                *}
{*****}

```

PROCEDURE ANALYSE_ACOUSTIQUE(NPTT:INTEGER);

```

CONST
    p=8;
TYPE
    Tableau=ARRAY[1..npT] of REAL;
    Fichier=FILE of REAL;
VAR

```

```

    FichierTest:Fichier;
    w,sp:Tableau;
    mff:Liste;
    anc,nouv:REAL;
    n,m,i,j,choix:INTEGER;
}-----}
PROCEDURE FFT(VAR xr: Tableau ; n,Choix:INTEGER);
LABEL .1,2,3;
TYPE
    Vecteur=ARRAY[1..10] of INTEGER;
    Matrix=ARRAY[1..2,1..2] of REAL;
VAR
    Iox : Vecteur;
    . xi : Tableau ;
    wk , a , q : Matrix;
    i,j,ii,j0,j1,j2,noo,l,fl,np,
    j00,fnp,mp1,moit,mex2,nmex2:INTEGER;
    u,as,ac,res1,res2:REAL;
PROCEDURE RCOM(r,x:REAL ; VAR a:Matrix);
BEGIN
    a[1,1]:=r;
    a[1,2]:=x;
    a[2,1]:=-x;
    a[2,2]:=r
END;
PROCEDURE CMULT(a,b:Matrix ; VAR c:Matrix);
BEGIN
    c[1,1]:=a[1,1]*b[1,1]+a[1,2]*b[2,1];
    c[1,2]:=a[1,1]*b[1,2]+a[1,2]*b[2,2] ;
    c[2,2]:=c[1,1];
    c[2,1]:=-c[1,2]
END;
FUNCTION PUISS(p,q:INTEGER):INTEGER;
VAR p1,i:INTEGER;
BEGIN
    p1:=p;
    FOR i:=1 TO q-1 do p:=p*p1;
    puiss:=p;
    if q=0 then puiss:=1
END;
BEGIN
    np:=puiss(2,n);
    FOR i:=1 TO np do xi[i]:=0;
    FOR i:=1 TO n do iox[i]:=puiss(2,n-i);

```

```

FOR mp1:=1 TO n do
BEGIN
mex2:=puiss(2,mp1-1);
nmex2:=np div mex2;
moit:=nmex2 div 2;
l:=0;
FOR j00:=1 TO mex2 do
BEGIN
fl:=1;
fnp:=np;
v:=choix*6.2831853*fl/fnp;
ac:=cos(v);
as:=sin(v);
RCOM(ac,as,wk);
j0:=nmex2*(j00-1);
FOR i:=1 TO moit do
BEGIN
j1:=j0+i;
j2:=j1+moit;
RCOM(xr[j2],xi[j2],a);
CMULT(a,wk,q);
xr[j2]:=xr[j1]-q[1,1];
xi[j2]:=xi[j1]-q[1,2];
xr[j1]:=xr[j1]+q[1,1];
xi[j1]:=xi[j1]+q[1,2];
END;
FOR i:=2 TO n do
BEGIN
ii:=i;
if l<iox[i] then goto 1;
l:=l-iox[i];
END;
1: l:=l+iox[ii];
END;
END;
l:=0;
FOR j:=1 TO np do
BEGIN
if l<j then goto 2;
res1:=xr[j];
res2:=xi[j];
xr[j]:=xr[l+1];
xi[j]:=xi[l+1];
xr[l+1]:=res1;

```



```

xi[l+1]:=res2;
2: FOR i:=1 TO n do
BEGIN
ii:=i;
if l<iox[i] then goto 3;
l:=l-iox[i]
END;
3: l:=l+iox[ii];
END;
if choix>0 then
FOR i:=1 TO np do
xr[i]:=ln(sqrt(sqr(xr[i])+sqr(xi[i])))
else FOR i:=1 TO np do
xr[i]:=xr[i]/fnp;
END;      (* fin de PROCEDURE fft *)
}-----}

BEGIN (* debut de PROCEDURE analyse acoustique *)
assign(FichierTest,stri);
reset(FichierTest);
n:=FileSize(FichierTest);
if (n/nptt)=(n div nptt) then ntT:=n div nptt
else ntT:=(n div nptt)+1;
FOR i:=1 TO nptt do w[i]:=0.54-0.46*cos(6.28*(i-1)/nptt);
FOR i:=1 TO nptt do sp[i]:=0;
anc:=0;
FOR j:=1 TO ntT do
BEGIN
Read(FichierTest,nouv);
sp[1]:=nouv-anc;
FOR i:=1 TO (nptt-1) do
BEGIN
if not eof(FichierTest) then
BEGIN
anc:=nouv;
Read(FichierTest,nouv);
sp[i+1]:=nouv-anc
END
else sp[i+1]:=0 ;
anc:=nouv
END;
FOR i:=1 TO nptt do sp[i]:=sp[i]*w[i];
choix:=1;
fft(sp,p,choix);

```

```

choix:=-1;
fft(sp,p,choix);
FOR i:=1 TO nmff do
mff[i]:=sp[i+1];
tabT[j]:=mff
END;
close(FichierTest)
END;      (*   fin de PROCEDURE analyse   *)
{-----}
{*****}
{*                                     *}
{**      NORMALISATION TEMPORELLE      **}
{*                                     *}
{*****}
{-----}
PROCEDURE NOR_TEMP(N:INTEGER);
  LABEL branch;
  VAR
  mi,mj,i,j:INTEGER;
  a,b,c:REAL;
{-----}
  FUNCTION intostr(x:REAL):STRING;
  VAR s:STRING[11];
  BEGIN
  Str(x:5:4,s);
  intostr:=s;
  END;
{-----}
  FUNCTION Dist_Loc(test,ref:liste;deb,fin:INTEGER):REAL;
  VAR
  i:INTEGER;
  x:REAL;
  BEGIN
  x:=0;
  FOR i:=deb TO fin do x:=x+abs(test[i]-ref[i]);
  Dist_Loc:=x
  END;
{-----}

BEGIN      (*   debut PROCEDURE norm_tem   *)
Dist_Glo:=2*Dist_Loc(tabt[1],tabr[1],deb,fin);
mi:=0 ; mj:=0;
i:=1 ; j:=1 ; if n=1 then MoveTo(x1+pai,y1-paj);

```

```

while (i<ntT) or (j<ntR) do
BEGIN
  if mi=i then
  BEGIN
    if FENETRE(i+1,j+1) then
    BEGIN
      mi:=i ; mj:=j ;
      i:=i+1 ; j:=j+1 ; if n=1 then lineTo(x1+i*pai,y1-j*paj);
      Dist_Glo:=Dist_Glo+2*Dist_Loc(tabT[i],tabR[j],deb,fin)
    END
    else
    BEGIN
      mi:=i ; mj:=j;
      i:=i+1; if n=1 then lineTo(x1+i*pai,y1-j*paj);
      Dist_Glo:=Dist_Glo+Dist_Loc(tabt[i],tabr[j],deb,fin)
    END;
    goto branch
  END;
  if mj=j then
  BEGIN
    if FENETRE(i+1,j+1) then
    BEGIN
      mi:=i ; mj:=j ;
      i:=i+1 ; j:=j+1 ; if n=1 then lineTo(x1+i*pai,y1-j*paj);
      Dist_Glo:=Dist_Glo+2*Dist_Loc(tabt[i],tabr[j],deb,fin)
    END
    else
    BEGIN
      mj:=j ; mi:=i ;
      j:=j+1; if n=1 then lineTo(x1+i*pai,y1-j*paj);
      Dist_Glo:=Dist_Glo+Dist_Loc(tabt[i],tabr[j],deb,fin)
    END;
    goto branch
  END;
  if FENETRE(i+1,j) then
  a:=Dist_Loc(tabT[i+1],tabR[j],deb,fin)
  else a:=1000;
  if FENETRE(i+1,j+1) then
  c:=Dist_Loc(tabT[i+1],tabR[j+1],deb,fin)
  else c:=1000;
  if FENETRE(i,j+1) then
  b:=Dist_Loc(tabT[i],tabR[j+1],deb,fin)
  else b:=1000;
  (* calcul du minimum *)

```



```

    if a<=b then
    BEGIN
    if a<=c then
    BEGIN
    Dist_Glo:=Dist_Glo+a;
    mi:=i ; mj:=j ; i:=i+1;if n=1 then lineTo(x1+i*pai,y1-j*paj);
    END
    else
    BEGIN
    Dist_Glo:=Dist_Glo+2*c;
    mi:=i;mj:=j;i:=i+1;j:=j+1;if n=1 then lineTo(x1+i*pai,y1-j*paj);
    END
    END
    else
    BEGIN
    if b<=c then
    BEGIN
    Dist_Glo:=Dist_Glo+b;
    mj:=j ; mi:=i ; j:=j+1;if n=1 then lineTo(x1+i*pai,y1-j*paj);
    END
    ELSE
    BEGIN
    Dist_Glo:=Dist_Glo+2*c;
    mi:=i;mj:=j;i:=i+1;j:=j+1;if n=1 then lineTo(x1+i*pai,y1-j*paj);
    END
    END;
branch: END;      (* fin de la boucle while *)
Dist_Glo:=Dist_Glo/(ntR+ntT);
if n=1 then BEGIN
outtextXY(0,200,'dist.glob.: '+intostr(Dist_Glo));
readln
END;
END;      (* fin de PROCEDURE nor_temp *)

```

```

<----->
{*****}
{*                                     *}
;  {**          DECISION          **}
{*                                     *}
{*****}
<----->

```

PROCEDURE PASSAGE1(KVAR T:VECT1);

```

LABEL 1;
VAR   i,j,k,Max :INTEGER;
      st         :STRING;

```

```
ch      :STRING;
Temp    :REAL;
```

```
BEGIN
Max:=nb_cl div 2 + 1;
FOR k:=1 TO 6 do BEGIN
ch:=Copy(Lettres,k,1);Write(' ',Ch);
Temp:=0 ; Long_Court:=0 ;
FOR j:=1 TO Max do BEGIN
str(j,St) ; st:=ch+st+'.Ref';
Assign(FichierRef,st) ; Reset(FichierRef);
i:=1;
Repeat
Read(FichierRef,Con); TabR[i]:=Con;
i:=i+1;
Until EOF(FichierRef);
ntR:=FileSize(FichierRef);
Close(FichierRef);
If (ntR>2*ntT) OR (ntR<ntT/2) then Long_Court:=Long_Court+1
Else BEGIN Nor_Temp(0);Temp:=Temp+Dist_Glo END;
END;
If Max>Long_Court then Dist1[k]:=Temp/(Max-Long_Court)
Else Dist1[k]:=9999;
END;
(* CLASSEMENT DES DISTANCES *)
FOR i:=1 TO 6 do T[i]:=0;
FOR i:=1 TO 6 do BEGIN
Temp:=Dist1[i];j:=1;
FOR k:=1 TO 6 do If Temp>Dist1[k] Then j:=j+1;
1: If T[j]=0 Then T[j]:=i Else BEGIN j:=j+1;GoTo 1 END;
END; (* Fin de Classement *)
END; (* Fin de PROCEDURE PASSAGE1 *)
```

```
PROCEDURE PASSAGE2(VAR Dist2:VECT2);
```

```
VAR Gd,Gm,i,j,k,Min :INTEGER;
Temp :REAL;
st,Ch :STRING;
```

```
BEGIN
Gd:=Detect;
InitGraph(Gd,Gm,'');
Min:=nb_cl div 2+1;
FOR i:=1 TO 4 do BEGIN
Ch:=Copy(Lettres,T[i],1);
```

```

Temp:=0 ; Long_Court:=0 ;
FOR j:=Min TO nb_cl do BEGIN
str(j,St);st:=Ch+st+'.Ref';
Assign(FichierRef,St);Reset(FichierRef);
k:=1;
Repeat
Read(FichierRef,Con); TabR[k]:=Con;
k:=k+1;
Until EOF(FichierRef);
ntR:=FileSize(FichierRef);
Close(FichierRef);
If (ntR>2*ntT) OR (ntR<ntT/2) then Long_Court:=Long_Court+1
Else BEGIN Graphe(Ch+Copy(St,2,1));Nor_temp(1);Temp:=Temp+Dist_Glo
END;
If Nb_Cl>(Long_Court+Min) then Dist2[i]:=Temp/(Nb_Cl-(Long_Court+Min)
Else Dist2[i]:=9999;
END;
CloseGraph;
END;

```

PROCEDURE RESULTATS:

```

VAR i,j :INTEGER;
      ch :STRING;
      A :REAL;
BEGIN
GoToXY(20,2);Write(Chr(218));
FOR i:=2 TO 38 do Write(Chr(196));
WriteLn(Chr(191));
GoToXY(20,3);WriteLn('                RESULTATS                ');
GoToXY(20,4);WriteLn('                DE                ');
GoToXY(20,5);WriteLn('                RECONNAISSANCE                ');
GoToXY(20,6);Write(Chr(192));
FOR i:=2 TO 38 do Write(Chr(196));WriteLn(Chr(217));
GoToXY(5,10);WriteLn('Voyelle Test = ',Ch1);
GoToXY(5,12);WriteLn('Premier Passage'); WriteLn;
FOR i:=1 TO 6 do BEGIN
Ch:=Copy(Lettres,i,1);
If i in [ T[1],T[2],T[3],T[4] ] Then BEGIN
HighVideo;WriteLn(' ',Ch,' ',Dist1[i]:4:4);NormVideo END
Else WriteLn(' ',Ch,' ',Dist1[i]:4:4);
END;
WriteLn;
WriteLn(' Les Voyelles les plus');
WriteLn(' Probables sont =');

```



```

FOR i:=1 TO 4 do BEGIN
Ch:=Copy(Lettres,T[i],1);Write(' ',Ch);
END;
GoToXY(25,12);WriteLn('Deuxieme Passage');
highVideo;
FOR i:=1 TO 4 do BEGIN
Ch:=Copy(Lettres,T[i],1);GoToXY(25,i+13);
Write(Ch,' ',Dist2[i]:4:4)
END;
(*      CALCUL DU MINIMUM      *)
A:=1000;
FOR j:=1 TO 4 do If dist2[j]<A Then BEGIN A:=Dist2[j];i:=j END;
Ch:=Copy(Lettres,T[i],1);
GoToXY(5,24);
Write('Voyelle Reconnue : ');Textcolor(White+Blink);
Write(ch);
Readln;
NormVideo;
END;      (*      Fin de PROCEDURE RESULTAT      *)

```

```

BEGIN      (*      programme principal      *)

```

```

Repeat

```

```

SAISIE_DONNEE;

```

```

ANALYSE_ACOUSTIQUE(NPT);

```

```

PASSAGE1(T);

```

```

PASSAGE2(DIST2);

```

```

RESULTATS;

```

```

GoToXY(40,25);Write('Voulez vous Reessayer [ O/N ] : ');

```

```

c:=ReadKey;

```

```

Until UpCase(c)='N';

```

```

NormVideo;

```

```

END.

```

```

(*****
(*) PROGRAMME CLASSIFICATION *)
(*) Ecole Nationale Polytechnique *)
(*) Réalisé par : BENACHOUR Abdelaziz *)
(*) dans le cadre du projet de fin d'étude *)
(*) JUN 90 *)
(*****

```

PROGRAM CLASSIFICATION:

```

USES CRT;
LABEL loop;
CONST deb=1;
      nmff =16;
      fin=8;
      nTmaX =20;
TYPE liste=array[1..nmff] of real;
      tableau=array[1..nTmaX] of liste;
      Enr = record
              ceps:tableau;
              elimine:boolean;
              nT:byte;
            END;
      Classe = File of Enr;
VAR Index,j,k,Fsize,Fpos,compteur,nb_cl:integer;
      Classe_S,Classe_D:string;
      Tab:Enr;
      Fichier:Classe;
      Test:boolean;
      Temoin:real;
      Lettre:char;

```

FUNCTION DISTANCE(TABT,TABR:ENR):REAL;

```

      LABEL branch;
      VAR mi,mj,i,j:integer;
          a,b,c,dist:real;
; {-----}
      FUNCTION Fenetre(i,j:integer):boolean ;
      LABEL 1,2;
      BEGIN
      if j<0.5*i then goto 1;
      if j>2*i then goto 1;
      if j>0.5*i+TabR.nT-0.5*(TabT.nT-1) then goto 1;
      if j<2*i+TabR.nT-(2*TabT.nT+1) then goto 1;

```

```

Fenetre:=true;goto 2;
1: Fenetre:=false;
2: END;
{-----}
FUNCTION Dist_Loc(test,ref:liste):real;
VAR i:integer;
    x:real;
BEGIN
x:=0;
FOR i:=deb TO fin DO x:=x+abs(test[i]-ref[i]);
dist_loc:=x
END;
{-----}
BEGIN (* debut PROCEDURE norm_tem *)
dist:=2*dist_loc(tabt.Ceps[1],tabr.Ceps[1]);
mi:=0 ; mj:=0;
i:=1 ; j:=1 ;
while (i<TabT.nT) or (j<TabR.nT) DO
BEGIN
if mi=i then
BEGIN
if fenetre(i+1,j+1) then
BEGIN
mi:=i ; mj:=j ;
i:=i+1 ; j:=j+1 ;
dist:=dist+2*dist_loc(tabt.Ceps[i],tabr.Ceps[j])
END
else
BEGIN
mi:=i ; mj:=j;
i:=i+1;
dist:=dist+dist_loc(tabt.Ceps[i],tabr.Ceps[j])
END;
goto branch
END;
if mj=j then
BEGIN
if fenetre(i+1,j+1) then
BEGIN
mi:=i ; mj:=j ;
i:=i+1 ; j:=j+1 ;
dist:=dist+2*dist_loc(tabt.Ceps[i],tabr.Ceps[j])
END
else

```



```

BEGIN
mj:=j ; mi:=i ;
j:=j+1;
dist:=dist+dist_loc(tabt.Ceps[i],tabr.Ceps[j])
END;
goto branch
END;
if fenetre(i+1,j) then
a:=dist_loc(tabT.Ceps[i+1],tabR.Ceps[j])
else a:=1000;
if fenetre(i+1,j+1) then
c:=dist_loc(tabT.Ceps[i+1],tabR.Ceps[j+1])
else c:=1000;
if fenetre(i,j+1) then
b:=dist_loc(tabT.Ceps[i],tabR.Ceps[j+1])
else b:=1000;
(* calcul du minimum *)
if a<=b then
BEGIN
if a<=c then
BEGIN
dist:=dist+a;
mi:=i ; mj:=j ; i:=i+1;
END
else
BEGIN
dist:=dist+2*c;
mi:=i;mj:=j;i:=i+1;j:=j+1;
END
END
else
BEGIN
if b<=c then
BEGIN
dist:=dist+b;
mj:=j ; mi:=i ; j:=j+1;
END
else
BEGIN
dist:=dist+2*c;
mi:=i;mj:=j;i:=i+1;j:=j+1;
END
END;
branch: END;      (* fin de la boucle while *)

```

```
Distance:=dist/(TabT.nt+TabR.nt);  
END;
```

```
FUNCTION FONCTION_CRITERE:REAL;
```

```
VAR i,n:integer;  
    y:real;  
    clas :string;  
    F:Classe;
```

```
FUNCTION HOMOGENIETE:real;
```

```
VAR i1,n1:integer;  
    Y1 :real;  
    Tab1 :Enr;
```

```
FUNCTION METRIQUE(X:Enr):real;
```

```
    LABEL saut;
```

```
    VAR i2:integer;  
        Y2:real;  
        Tab2:Enr;
```

```
    BEGIN (* Fonction Metrique *)
```

```
        Y2:=0;
```

```
        Seek(F,0);
```

```
        FOR i2:=0 TO n-1 DO BEGIN
```

```
            read(F,Tab2);
```

```
            if (i2=n1-1) OR (Tab2.elimie) then GoTo saut;
```

```
            Y2:=Y2+DISTANCE(Tab2,X);
```

```
        saut:END;
```

```
        Seek(F,n1);
```

```
        METRIQUE:=Y2/(n-1);
```

```
        END; (* Fin de la fonction Metrique *)
```

```
    BEGIN
```

```
        Y1:=0;
```

```
        FOR i1:=1 TO n DO BEGIN
```

```
            read(F,Tab1);
```

```
            if not Tab1.elimie then BEGIN
```

```
                n1:=FilePos(F);
```

```
                Y1:=Y1+METRIQUE(Tab1) END;
```

```
            END;
```

```
        HOMOGENIETE:=Y1/n;
```

```
    END;
```

```
BEGIN
```

```
Y:=0;
```

```
FOR i:=1 TO nb_cl DO BEGIN
```

```

str(i,clas);clas:='Classe_'+clas;
Assign(F,Clas);reset(F);n:=FileSize(F);
Y:=Y+HOMOGENIETE*(n-1);
close(F)          END;
FONCTION_CRITERE:=Y
END;

```

PROCEDURE ECRAN;

```

VAR i:integer;

BEGIN
ClrScr;
GoToXY(20,2); write(chr(218));
FOR i:=2 TO 39 DO write(chr(196));WRITELN(chr(191));
GoToXY(20,3);WRITELN('');
GoToXY(20,4);WRITELN('          CLASSIFICATION');
GoToXY(20,5);WRITELN('');
GoToXY(20,6);write(chr(192));
FOR i:=2 TO 39 DO write(chr(196));WRITELN(chr(217));
GoToXY(31,10);WRITELN('Programme Realisé');
GoToXY(39,11);WRITELN('Par');
GoToXY(30,12);highvideo;write('Benachour Abdelaziz');
GoToXY(22,13);NormVideo;
write('Dans le cadre du projet de fin d''etude');
GoToXY(27,14);HighVideo;write('RECONNAISSANCE AUTOMATIQUE');
GoToXY(34,15);write('DE LA PAROLE');
GoToXY(28,16);write('PAR LES METHODES GLOBALES');
NormVideo;
GoToXY(36,17);write('Juin 90');
delay(5000);
Window(21,11,59,15);textbackground(red);
FOR i:=1 TO 195 DO write(' ');
HighVideo;GoToXY(3,2);
write('Donner la lettre a classifier ');
readln(lettre);
GoToXY(3,3);WRITELN('Donner le nombre de representant: ');
GotoXY(2,4);readln(nb_cl);
NormVideo;
Window(1,1,80,25);
END;

```

PROCEDURE MESSAGE(M:STRING);


```

VAR i:integer;
BEGIN
ClrScr;
window(21,11,59,15);textbackground(red);
FOR i:=1 TO 195 DO write(' ');textcolor(white);
GoToXY(12,3);write(M);
sound(1500);
delay(500);
nosound;
normVideo;
END;

```

PROCEDURE DEBUT;

```

VAR F1,F2:Classe;
    n,i,j: integer;
    stri : string;
    Tab  : Enr;
BEGIN
ClrScr;
gotoXY(1,24);WRITELN('Donner le fichier de depart ');
readln(stri);
assign(F1,stri);reset(F1);
n:=fileSize(F1);n:=n div nb_cl;
FOR i:=1 TO nb_cl DO BEGIN
str(i,stri);stri:='classe_'+stri;
assign(F2,stri);rewrite(F2);
FOR j:=1 TO n DO BEGIN
read(F1,Tab);write(F2,Tab);
END;
if i=nb_cl then
while not eof(F1) DO BEGIN
read(F1,tab);write(F2,Tab) END;
close(F2);
END;
close(F1);
END;

```

PROCEDURE ECHANGE(X:ENR:C_S;STRING);

```

LABEL saut;
VAR F2 :Classe;
    i :integer;
    Fch_Destination,Fch : string;

```

```

BEGIN
FOR i:= 1 TO nb_cl DO BEGIN
if i=Index then goto saut;
str(i,Fch);Fch:='Classe_'+Fch;
assign(F2,Fch);reset(F2);
Seek(F2,FileSize(F2));write(F2,X);
(*   elimination de X   *)
Seek(Fichier,Fpos-1);X.elimine:=true;write(Fichier,X);
close(F2);close(Fichier);
if Fonction_Critere<Temoin then BEGIN
Fch_Destination:=Fch;
Temoin:=Fonction_Critere           END;
Assign(F2,Fch);reset(F2);Seek(F2,filesize(f2)-1);
Truncate(F2);
Assign(Fichier,C_S);reset(Fichier);
X.elimine:=false;Seek(Fichier,Fpos-1);write(Fichier,X);
close(F2);           END;
Assign(F2,Fch_Destination);reset(F2);
Seek(F2,FileSize(F2));write(F2,X);
close(F2);
X.elimine:=true;Seek(Fichier,Fpos-1);
write(Fichier,X);
saut=END;           (* FIN DE PROCEDURE ECHANGE *)

```

PROCEDURE REPRESENTANTS;

```

VAR i,j,n,n1:integer;
    stri:String;
    Temoin:real;
    Tab,TabR:Enr;
    F :Classe;
    Fch:File of Liste;
FUNCTION METRIQUE(X:Enr):real;
    LABEL saut;
    VAR i2:integer;
        Y2:real;
        Tab2:Enr;
    BEGIN           (* Fonction Metrique *)
    Y2:=0;
    Seek(F,0);
    FOR i2:=0 TO n-1 DO BEGIN
    read(F,Tab2);
    if (i2=n1-1) OR (Tab2.elimine) then GoTo saut;
    Y2:=Y2+DISTANCE(Tab2,X);

```

```

saut=END;
    Seek(F,n1);
    METRIQUE:=Y2/(n-1);
    END;
BEGIN
FOR i:=1 TO nb_cl DO BEGIN
    Temoin:=999999;
    str(i,stri);stri:='classe_'+stri;
    Assign(F,stri);Reset(F);
    n:=FileSize(F);
    FOR j:=1 TO n DO BEGIN
        Read(F,Tab);
        If NOT Tab.elimine then BEGIN
            n1:=FilePos(F);
            If METRIQUE(Tab)<Temoin then BEGIN
                Temoin:=METRIQUE(Tab);
                TabR:=Tab;
            END;
        END;
        Close(F);
        Str(i,stri);Stri:=Lettre+stri+'.REF';
        Assign(Fch,stri);Rewrite(Fch);
        FOR j:=1 TO TabR.nT DO Write(Fch,TabR.Ceps[j]);
    END;
END;
(* Fin de la PROCEDURE REPRESENTANTS *)

```

FUNCTION TRANSFERER(X:ENR:C_S,C_T:STRING):BOOLEAN;

```

VAR F1 :Classe;
    Vrai_Faux :boolean;
BEGIN
Assign(F1,C_T);reset(F1);
Seek(F1,FileSize(F1));write(F1,X);
X.elimine:=true;Seek(Fichier,Fpos-1);
write(Fichier,X);close(Fichier);close(F1);
if Fonction_Critere < Temoin then Vrai_Faux:=true
else Vrai_Faux:=false;
Assign(F1,C_T);reset(F1);
Seek(F1,FileSize(F1)-1);Truncate(F1);
assign(Fichier,C_S);reset(fichier);
X.elimine:=false;Seek(Fichier,Fpos-1);write(Fichier,X);
close(F1);
TRANSFERER:=Vrai_Faux
END; (* FIN DE LA FONCTION TRANSFERER *)

```



```

BEGIN (* Programme Principal *)
ECRAN:
DEBUT:
MESSAGE('CLASSIFICATION');
Window(1,1,80,25);
compteur:=0;
Temoin:=Fonction_Critere;
repeat
Test:=False;
FOR Index:=1 TO nb_cl DO BEGIN
str(Index,Classe_S);
Classe_S:='Classe_'+Classe_S;
assign(Fichier,Classe_S);
Reset(Fichier);
Fsize:=FileSize(Fichier);
FOR j:=1 TO Fsize DO BEGIN
read(Fichier,Tab);
if NOT Tab.elimine then BEGIN
Fpos:=FilePos(Fichier);
FOR k:=1 TO nb_cl DO BEGIN
if k=Index then GoTo loop;
Str(k,classe_D);Classe_D:='Classe_'+Classe_D;
if TRANSFERER(Tab,Classe_S,Classe_D) then
BEGIN
Test:=true;
compteur:=compteur+1;
gotoXY(44,25);write('Nombre d'elements transferes: ',compteur);
ECHANGE(Tab,Classe_S);
END;
loop:=END;
END;
END;
close(Fichier);
END;
until(not Test);
MESSAGE('REPRESENTANT');
REPRESENTANTS:
window(1,1,80,25);ClrScr;
WRITELN('EXECTION TERMINEE ');
WRITELN('APPUYER sur Return pour revenir au systeme');
readln;
END.

```

annexe

exemples d'execution

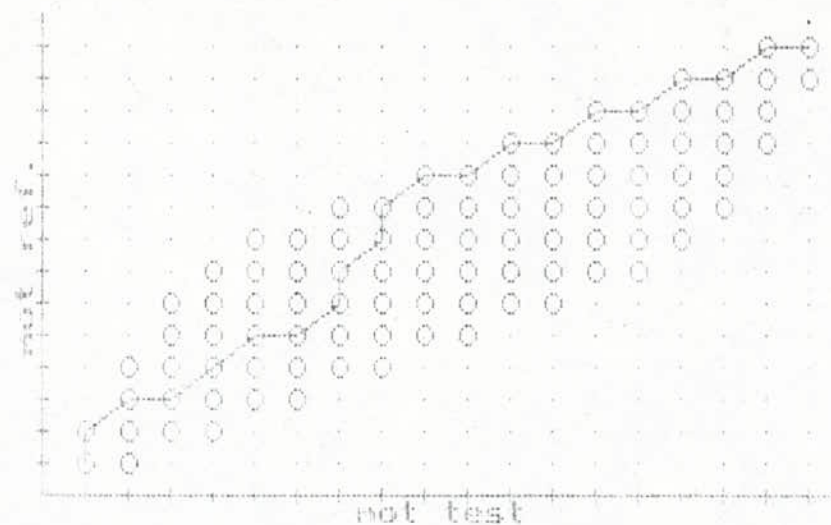
normalisation temporelle

Lettre Test : A

lettre reference: Y

resultats

dist.glob.: 3.3143



comparaison de la voyelle A avec la voyelle Y
FONCTION DE DEFORMATION

Exemple d'exécution en mode monolocuteur.

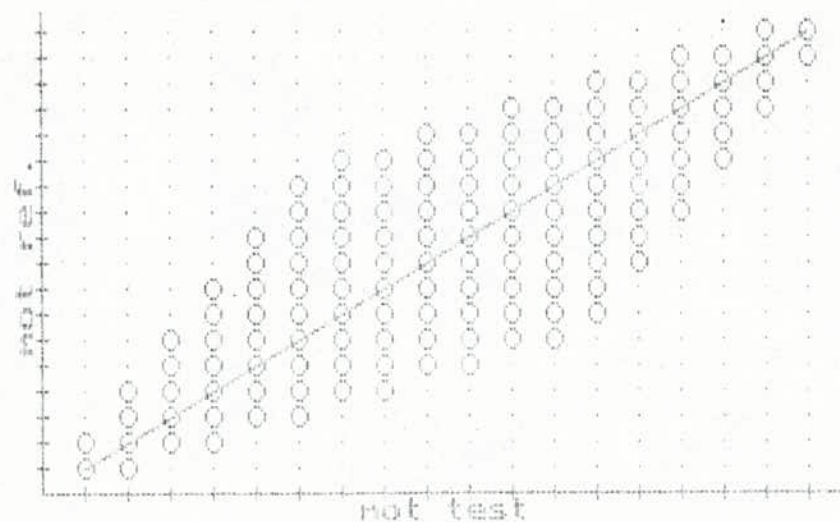
normalisation temporelle

Lettre Test : A

lettre reference: A

resultats

dist.glob.: 0.0000



Ici le fichier est le meme que celui
qui a servi à la creation de la référence

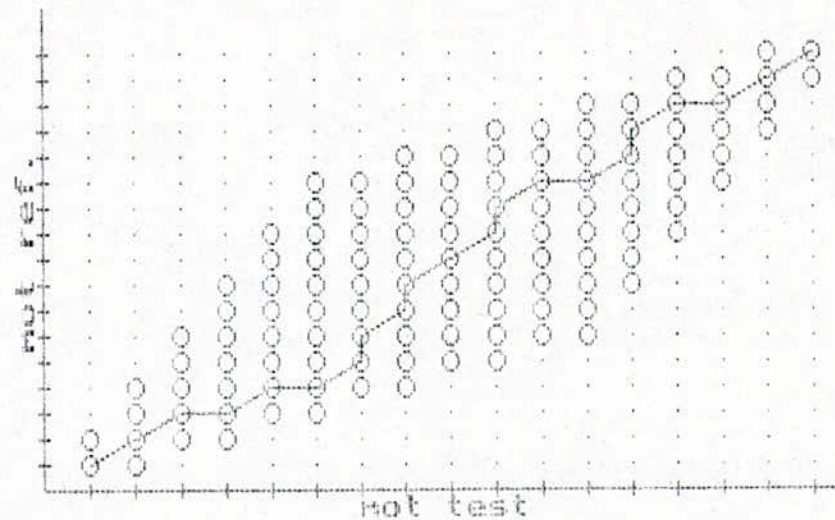
normalisation temporelle

Lettre Test : 0

lettre reference: 03

resultats

dist.glob.: 0.8580



...Comparaison de la voyelle 0 avec ses differents representants...

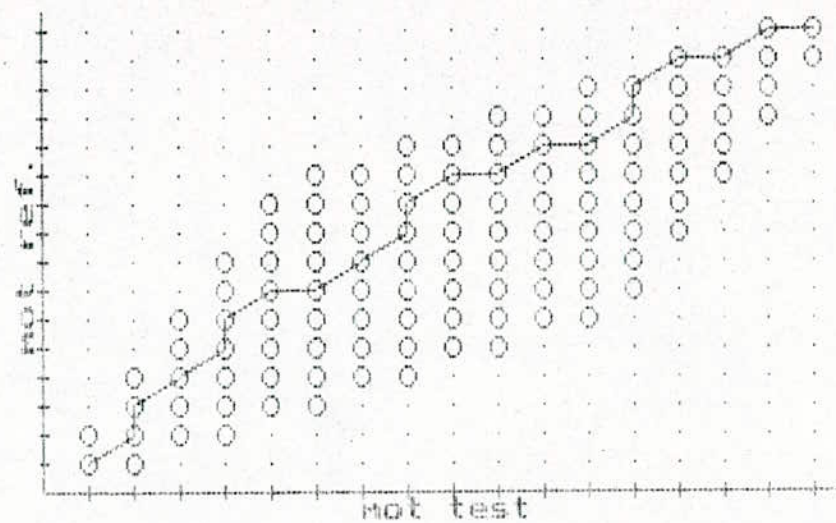
TEST MULTILOCUTEURS

Nbre de representants 4

Résultats du deuxième passage.

normalisation temporelle

Lettre Test : 0
lettre reference: 04
resultats
dist.glob.: 0.8715



...Comparaison de la voyelle 0 avec ses differents representants...

TEST MULTILOCUTEURS

Nbre de representants 4

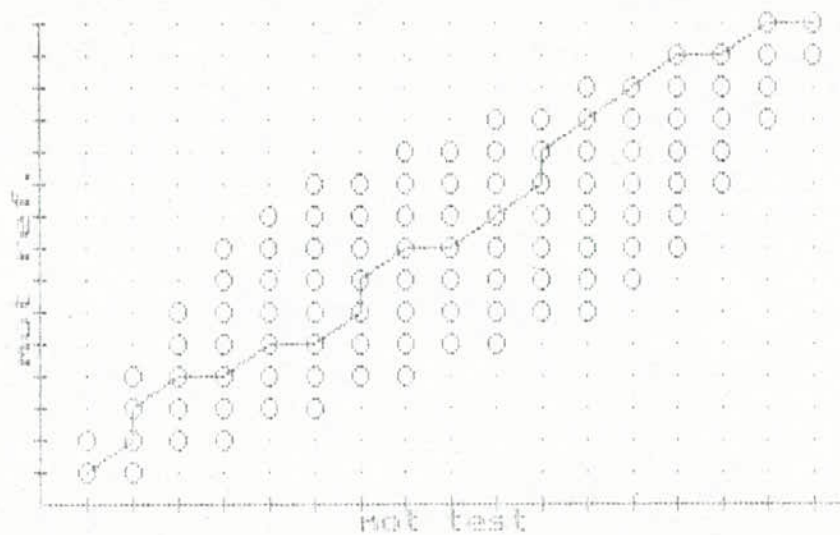
normalisation temporelle

Lettre Test : 0

lettre reference: U3

resultats

dist.glob.: 0.8032

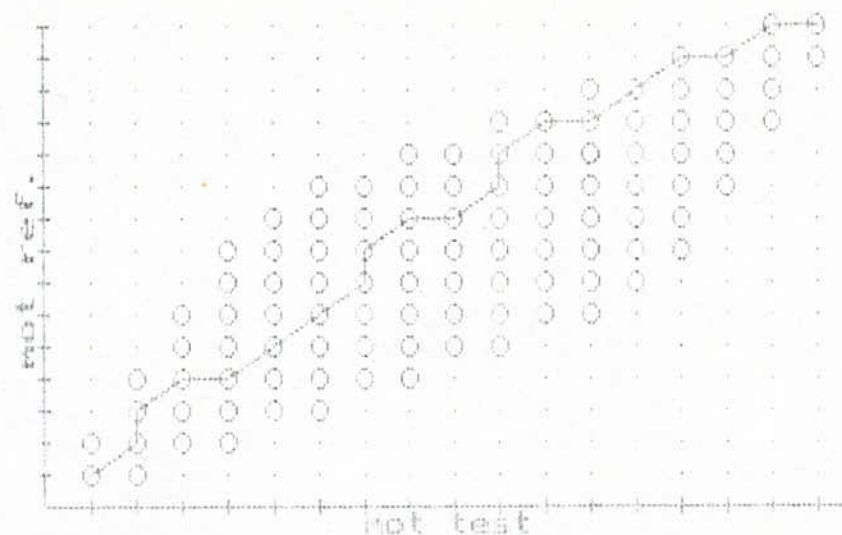


..Comparison de la voyelle 0 avec les representants de la voyelle U.
TEST MULTILOCUTEURS

normalisation temporelle

Lettre Test : O
lettre reference: U4
resultats

dist.glob.: 1.0167



...Comparaison de la voyelle O avec les représentants de la voyelle U.
TEST MULTILOCUTEURS

BIBLIOGRAPHIE

- [1] M. KUNT , Traitement numérique des signaux DUNOD
- [2] R. BOITE & M. KUNT , Le traitement de la parole
- [3] J. GUIBERT , La parole Collection "Que sais-je"
- [4] A. CHEVALIER , La programmation dynamique et ses applications
- [5] E. EMERIT , Cours de phonétique acoustique SNED
- [6] A. MENACER, "Reconnaissance de la parole en mode multilocuteurs" Thèse de docteur-ingénieur.
- [7] A. MOKKEDEM, "Analyse de la parole , reconnaissance multilocuteur des mots isolés" Thèse de doctorat
- [8] G. MERCIER, "Le décodage acoustico-phonétique de la parole, Note technique NT/LAA/TSS/194
- G. MERCIER, "La reconnaissance de la parole - Etat actuel de la technique et problèmes à résoudre" NT/LAA/TSS 194
- [9] C. GAGNOULET, "Système séraphine" NT/LAA/TSS 189
- [10] J. MARIANI, "La reconnaissance de la parole" La recherche N° 170 Octobre 1985
- [11] J. TREMOLIRE, "La synthèse de la parole", Electronique applications Oct-Nov 1988
- [12] SAKOE & CHIBA, "Dynamic programming algorithme optimisation for spoken word recongnition IEEE ASSP 26 N°1 Fev 1978

