

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT D'ELECTRONIQUE

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

PROJET DE FIN D'ETUDES

S U J E T

**Contribution à la Reconnaissance
et la Compréhension de la
Parole Continue**

Proposé par :

B. BOUSSEKSOU

Etudié par :

M. NOUACER

A. DAOUDI

Dirigé par :

B. BOUSSEKSOU

PROMOTION : JUIN 1986

DEDICACES

=====

Je dédie ce modeste travail:

- * A la mémoire de mon père.
- * A ma mère, mon frère et mes soeurs pour leurs sacrifices.
- * A M'Barak et F/A.
- * A mes amis.
- * ET à tous ceux qui me sont chers.

A. Daoudi

REMERCIEMENTS

=====

Nous remercions vivement notre promoteur Mr B. BOUSSEKSOU pour ses conseils éclairés et son aide précieuse tout au long de l'élaboration de ce projet de fin d'étude.

Nous tenons à exprimer notre réelle et sincère reconnaissance à tous les professeurs de l'E.N.P.A ainsi que tous ceux qui ont contribué à notre formation depuis le primaire.

Nos remerciements à tous ceux qui nous ont aidé de près ou de loin dans la préparation de ce projet. Ainsi qu'aux membres de jury pour l'honneur qu'ils nous font en jugeant notre modeste travail.

TABLES DES MATIERES

Pages

PREMIERE PARTIE :

CHAPITRE 0	INTRODUCTION.....	1
CHAPITRE 1	PHONATION.....	3
1.1	Anatomie de l'appareil phonatoire.....	3
1.2	Production de la parole.....	3
1.3	Simulation du signal parole.....	5
1.4	Conclusion.....	6
CHAPITRE 2	ANALYSE.....	7
2.1	Prétraitement.....	8
2.1.1	Détection bruit parole	
2.1.2	Préaccentuation	
2.1.3	Fenetrage	
2.2	Traitement homomorphique et cepstre.....	10
2.3	Analyse cepstrale.....	10
2.3.1	Propriétés du cepstre	
2.3.2	Ecelle MEL des coefficients cepstraux	
2.4	Conclusion.....	13
CHAPITRE 3	APPRENTISSAGE.....	14
3.1	Analyse factorielle.....	14
3.1.1	Principe de la méthode	
3.1.2	Recherche de la solution	
3.1.3	Conclusion	
3.2	Classification.....	16
3.2.1	Notions mathématiques	
3.2.2	L'algorithme d'échange basé sur une fonction critère (A.E.C)	

3.2.3 Conclusion

CHAPITRE 4	PROGRAMMATION DYNAMIQUE.....	19
4.1	Distance cepstrale.....	19
4.2	Fonction de coïncidence.....	20
4.3	Fonction de pondération.....	20
4.4	Contraintes.....	21
4.5	Programmation dynamique.....	23
CHAPITRE 5	RECONNAISSANCE DES MOTS CONNECTES.....	25
5.1	L'alignement DTW entre T et R ^S	25
5.2	Préliminaires.....	26
5.3	Comparaison de la forme test à plusieurs formes de référence.....	27
5.4	Le backtracking.....	28
5.5	L'algorithme LB-DTW.....	30
5.6	Certaines modifications de l'algorithme LB- DTW.....	31
5.6.1	Réduction du champ de recherche de la fonc- tion de déformation.	
5.6.2	Technique de réduction d'intervalle.	
5.6.3	Tolérance sur le dernier point du test.	
5.6.4	Introduction des flous.	
5.7	L'algorithme LB-DTW modifié.....	36
5.8	Conclusion.....	37

DEUXIEME PARTIE :

I.	COMMENTAIRES.....	38
II.	ORGANIGRAMMES.....	40
	A). Organigramme de reconnaissance type " LB-DTW ".....	40
	B). Organigramme de détection de mots dans la parole continue " sous programme DDTW "	

c). Fonction calculant la distance entre
deux échantillons de parole A et B...48

III. PROGRAMME.....49

CONCLUSION GENERALE.....50

BIBLIOGRAPHIE

CHAPITRE : ()

INTRODUCTION

Depuis l'arriver des ordinateurs, la parole est sortie du domaine de la linguistique essentiellement descriptive pour être explorée par tous les scientifique (acousticiens, physiciens, physiologistes, informaticiens, ...) et devenir une science exacte d'une grande importance surtout en intelligence artificielle.

Le signal vocal possède des propriétés très spécifiques. Il est extrêmement variable et multi-forme, du fait des processus physiques très divers mis en œuvre lors de la production de la parole: sons vocaliques pseudo-périodiques, bruit de friction, ect... C'est pourquoi dans un premier chapitre on va donner certains aspects de la phonation.

L'analyse acoustique permet la saisie du signal vocal et son prétraitement pour l'extraction des paramètres pertinents, par des techniques de traitement du signal. Dans le second chapitre on décrira brièvement une de ces méthodes qui est l'analyse cepstrale. Celle-ci est surtout intéressante en reconnaissance, grâce à son économie de représentation.

La constitution du dictionnaire de référence est une phase très importante de l'apprentissage dont dépendra la reconnaissance correcte des formes prononcées par un locuteur. Ainsi un troisième chapitre présente les deux étapes fondamentales de l'apprentissage. La première étant l'analyse factorielle qui permet une réduction de données considérable par le passage d'une représentation dans un espace à une représentation plane. La seconde est une classification de l'ensemble de données selon un algorithme d'échange basé sur une fonction critère.

Au cours d'une phase préalable d'apprentissage, les formes de référence correspondant aux différents mots du vocabulaire sont mémorisées. La reconnaissance d'un mot inconnu se fait alors par comparaison de sa forme aux différentes formes de référence. Dans le cas d'un système monolocuteur le principal problème est un problème de normalisation temporelle. En effet, le mot peut subir des distorsions temporelles dues à des variations du rythme d'élocution. L'algorithme de comparaison de deux mots doit prendre en compte ces fluctuations non linéaires de façon à donner une mesure optimale de la distance entre ces mots. La programmation dynamique, thème du 4^{ème} chapitre, fournit à ce problème une solution optimale largement utilisée.

Dans le 5^{ème} chapitre, la technique de construction par niveau (level building) est proposée pour la reconnaissance de mots connectés. Les problèmes du traitement en temps réel, d'encombrement mémoire et de flexibilité exigent des modifications de l'algorithme LB-DTW. Ainsi l'algorithme LB modifié est discuté dans ce chapitre.

En deuxième partie nous présenterons un programme LB-DTW modifié écrit en FORTRAN et contenant un sous programme de détection de mots dans la parole continue, suivie d'un sous programme fonction calculant la distance entre deux échantillons de parole.

Le champ de recherche sur la parole est vaste. Il exige des connaissances très variées et multidisciplinaires. Tout ces conditions s'ajoutent au temps qui nous est approprié pour nous obliger à arrêter en ce point notre étude, très soigneusement élaborée, en espérant qu'elle soit bénéfique à tout ceux qui s'intéressent à la communication parlée.

الدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

CHAPITRE : 1

CHAPITRE : 1

PHONATION

L'étude de la voix et des paroles émises par les structures anatomiques de l'être humain permet d'expliquer le pourquoi et le comment de l'expression vocale. Elle apporte les moyens de corriger éventuellement les troubles qui font l'objet de toute la pathologie de la phonation. C'est un apport très important pour la rééducation puisque l'on sait utilement mettre en œuvre, ensemble ou séparément, les structures anatomiques, les cavités, les muscles concourant à la prononciation d'une parole.

1.1 Anatomie de l'appareil phonatoire :

Il est intéressant de savoir par quel mécanisme nous prononçons. Dans ce but, simplifiée à l'extrême, l'anatomie de l'appareil vocal humain a été représentée dans la figure 1.1, à gauche. A côté, un modèle mécanique, ou plus exactement pneumatique, comme on le monterait dans un robot devant être capable de parler sans utiliser un haut-parleur.

La cavité nasale, indéformable, est couplée au pharynx par la voile du palais; la position de la langue règle le volume et le couplage des deux autres cavités. A savoir la cavité du pharynx et la cavité bucale. Chaque cavité, prise isolément, résonne à des fréquences déterminées par sa forme et son volume; le couplage des deux cavités fait apparaître des fréquences déplacées par rapport aux fréquences propres de chacune d'elles.

1.2 Production de la parole :

Lors d'une l'occurrence, les poumons fournissent un flux d'air qui va traverser le larynx puis le conduit vocal. La production des sons s'effectue lorsque le flux précédent subi une modulation au niveau des cordes vocales dont le mécanisme de vibration est assez

1. Cavité nasale
2. Cavité orale
3. Voile du palais
4. Larynx
5. Langue
6. Cordes vocales
7. Trachée
8. Poumons

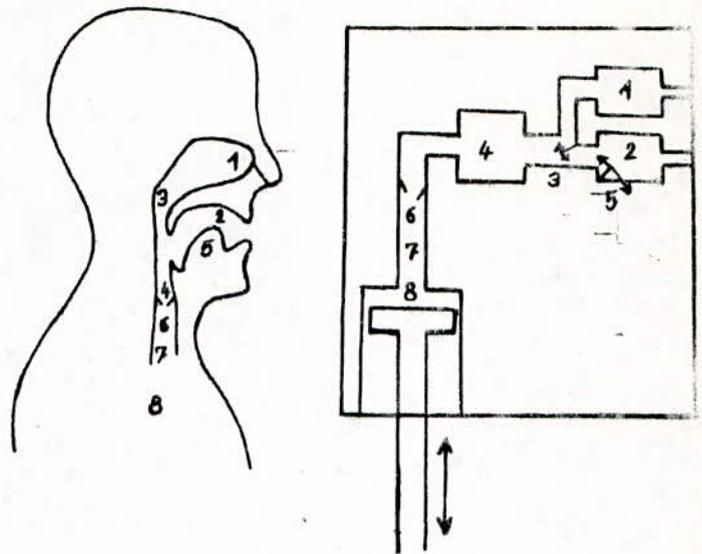


Figure 1.1 Imitation de l'appareil vocale humain par un robot pneumatique

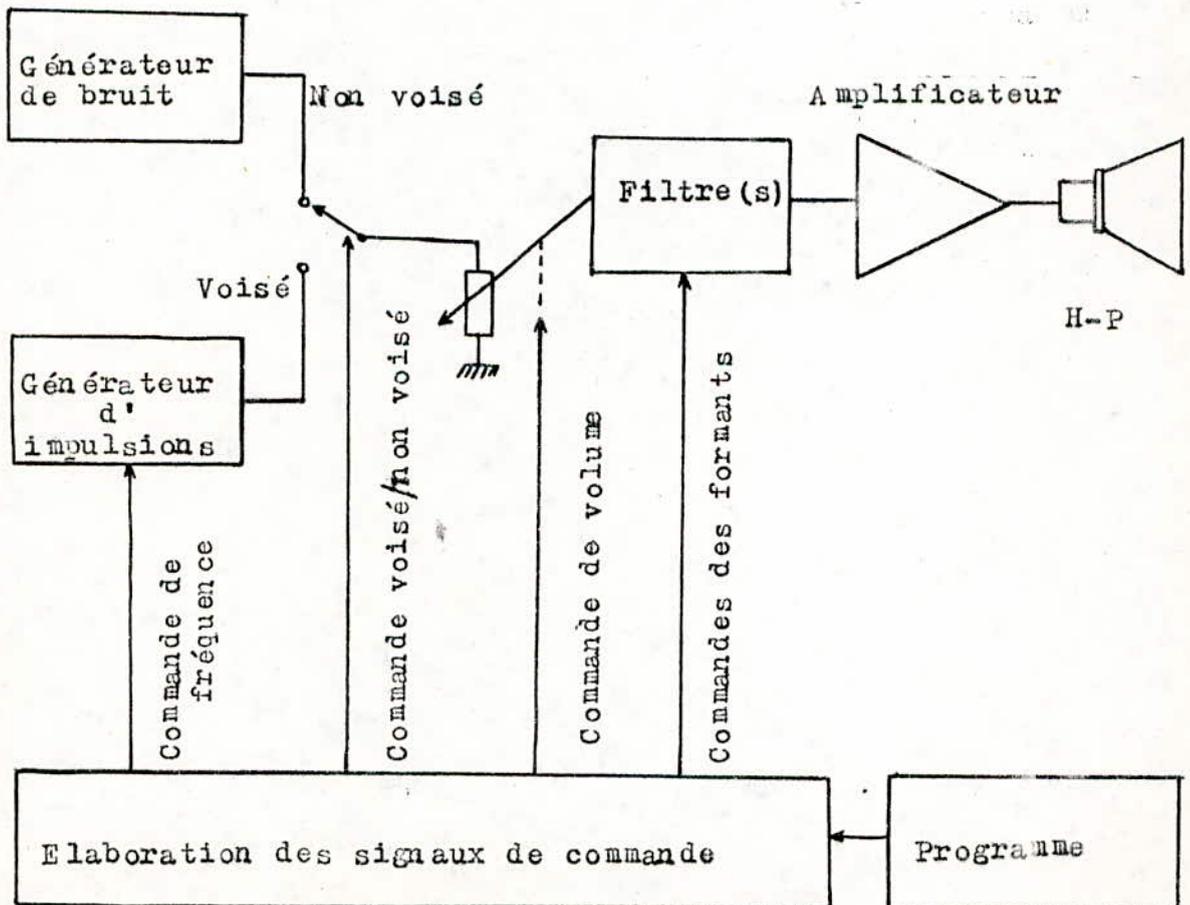


FIGURE 1.2 Organigramme décrivant la synthèse par formants aussi bien que celle à prédiction

complexe. Quand elles sont accolées, la pression augmente et les contraint à s'écarter mais des forces de rappel les amènent à se recoller la pression réaugmente alors et la vitesse de répétition de ce cycle détermine la fréquence fondamentale. L'intervention des cordes vocales dans l'obtention d'un son s'appelle le voisement et ce son est dit voisé.

Des sons peuvent être aussi obtenus lorsque les cordes vocales sont ouvertes et ceci en modulant le flux d'air par d'autres organes articulatoires tels que les lèvres et la langue. Ainsi en réduisant la dimension du conduit vocal on génère une turbulence. Le signal sonore résultant a l'aspect d'un bruit. Contrairement aux sons voisés, les sons engendrés par des flux turbulents sont apériodique. Ces deux types de sons peuvent être combiner pour obtenir d'autres sons tels que les fricatives voisées.

Un troisième type d'émission sonore est obtenu lorsqu'on libère brusquement la pression accumulée derrière un obstacle. Une obturation du conduit vocal est produite soit par la fermeture des lèvres (sons labiaux), soit par la langue contre les dents (dentales) ou le palais (palatales). Ces explosions d'énergie acoustique ont lieu lorsqu'on prononce des consonnes voisées ou non voisées.

En fin les sons nasalisés, sont obtenus lorsque le canal vocal est modifié par l'abaissement du voile du palais mettant en dérivation les cavités nasales par où va s'échapper l'air expiré.

1.3 Simulation du signal parole :

La figure 1,2 montre le principe d'un tel simulateur. Il contient un générateur d'impulsions qui imite les cordes vocales, et dont la fréquence peut-être commandée dans les limites relativement étroites qui sont celles du fondamental de la voix.

Quand nous prononçons des sons tels que ch, s, p, t, nos cordes vocales ne sont pas excitées. C'est simplement l'air sortant de nos poumons qui joue au tourbillon dans les cavités de la figure 1,1.

ceci peut-être imiter (fig. 1.2) par un générateur de bruit. De plus il faut prévoir un commutateur entre son "voisé" et son "non voisé". Après ce commutateur, on trouve une commande de volume qui traduit, en quelque sorte, la pression d'air sortant du locuteur. En suite, un filtrage réalisé par un seul filtre à pôles multiples, ou plusieurs filtres, éventuellement disposés en treillis avec des commandes individuelles de fréquence centrale et de largeur de bande.

1.4 Conclusion:

Le signal de la parole, dans son mode de production comme dans ses propriétés acoustiques, temporelles et fréquentielles est très variable; encore en donnons nous une description très simplifiée, probablement erronée aux yeux d'un phonéticien, mais suffisante pour comprendre les particularités du traitement de la parole.

CHAPITRE : 2

CHAPITRE : 2

ANALYSE

Pour analyser un signal, différentes solutions existent. De la courbe représentative du signal, un certain nombre de paramètres peuvent être tirés: soit les abscisses de passage par zéro de cette courbe, soit les coordonnées de ses extrêmes. Ce sont les méthodes d'analyse temporelles.

Mais ce sont surtout les méthodes fréquentielles ou spectrales qui jouent un rôle fondamental en analyse de la parole; les sons émis par la voix peuvent, en effet, être assez bien décrits en termes de fréquences. La parole étant un phénomène non stationnaire, il importe de faire intervenir le temps comme une troisième variable dans la représentation. La surface dans l'espace tridimensionnel de coordonnées amplitude-fréquence-temps caractérise entièrement le signal; c'est l'"image acoustique". Elle peut être analysée par un banc de filtres. Un jeu de huit filtres ou plus, effectué 50 ou 100 fois par seconde, donnera pour chaque filtre la valeur numérique de l'amplitude correspondant à la fréquence du filtre. L'ensemble des valeurs numériques ainsi obtenues pour un mot constituent l'image acoustique, ou sonagramme, de cet mot.

Le choix des paramètres pour le traitement du signal peut aussi s'inspirer des modèles physiques du mécanisme phonatoire humain. L'état des cordes vocales peut être décrit par un ensemble de paramètres numériques qui définissent un son vocal. Comme cet état varie relativement lentement, un petit nombre de bits suffit à le codifier. Les résonances des diverses cavités du canal phonatoire humain se traduisent par l'apparition de fréquences privilégiées appelées formants, lesquelles peuvent constituer des paramètres du signal vocal.

2.1 Prétraitement;

2.1.1 Détection bruit/parole:

L'algorithme de détection bruit/parole utilisant la notion de la FFT/14/, doit détecter la présence d'un signal de parole. En plus il doit localiser les débuts et les fins de mots, éliminer le bruit des lèvres et de la respiration et en fin la détection des phonèmes à faible énergie tels que les fricatives.

De ses bonnes performances dépendent la validité et la viabilité des mots qu'il serait sinon vain tant de reconnaître que d'employer comme références.

2.1.2 Préaccentuation:

L'onde sortant des lèvres subit une atténuation, puisqu'il y a désadaptation entre les deux milieux, et une distortion qui peut-être assimilée à une désaccentuation de 6dB par octave sur tout le spectre. Pour palier à cet inconvénient on préaccentue le signal de + 6 dB par octave. Ainsi on peut estimer l'évolution de la forme du conduit vocal et les lieux d'articulation et en déduire ainsi la parole prononcée.

2.1.3 Fenêtrage:

La manière brutale de limiter en fréquence un signal est de le multiplier par un signal rectangulaire appelé fenêtre rectangulaire:

$$W_R(f) = \frac{\sin \pi f N}{\sin \pi f}$$

où N est le nombre d'échantillons.

Une telle fenêtre (fig. 2.1.a) donnera la largeur (égale à $\frac{1}{N}$) du pic central la plus faible mais provoque l'apparition de pics secondaires très énergétiques dont les maxima décroissent lentement.

Pour remédier à cette situation on modifie la forme de la fonction fenêtre en super-posant trois répliques de la fonction $W_R(f)$ décalées par un pas de $\frac{1}{N}$. La fonction ainsi obtenue est dite fenêtre de Hamming $W_H(k)$.

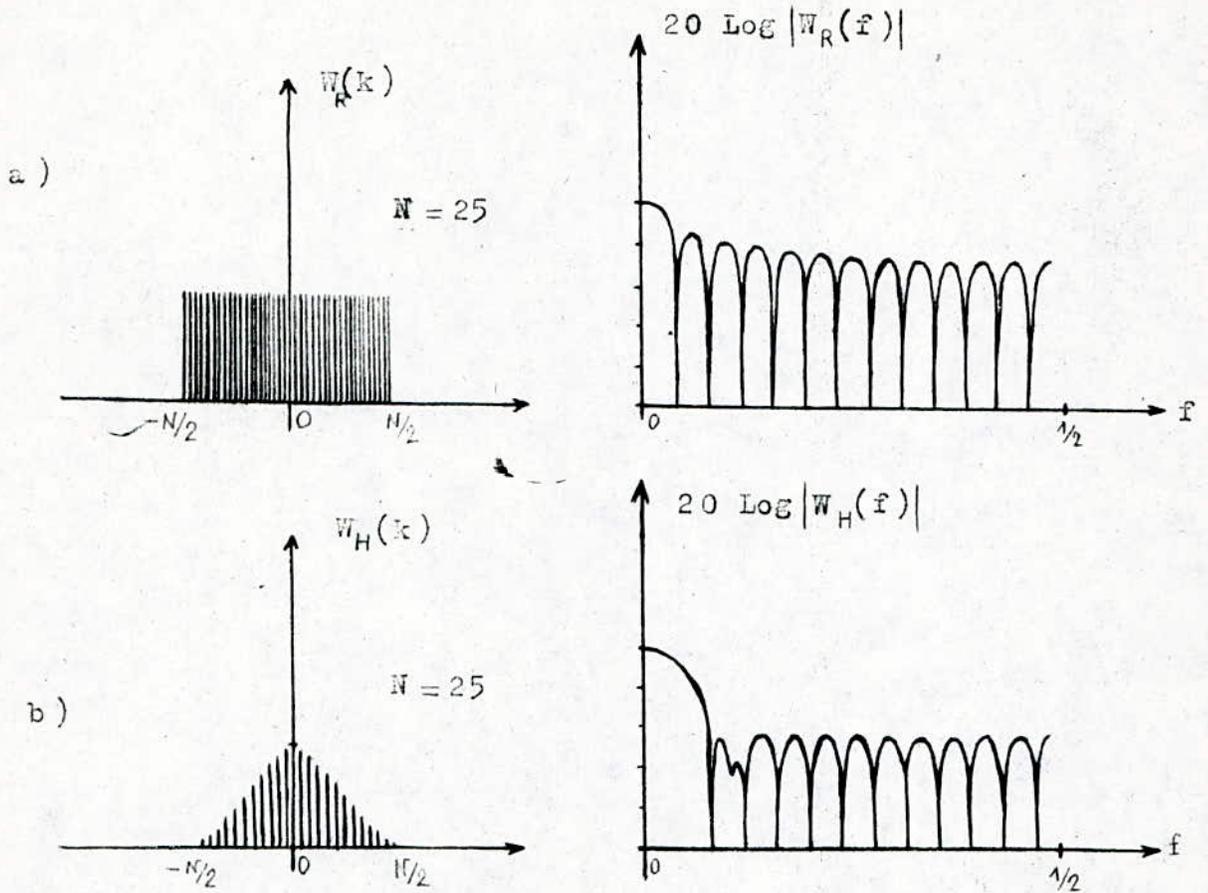


Fig. (2.1) fenêtre rectangulaire W_R
et fenêtre de Hamming W_H

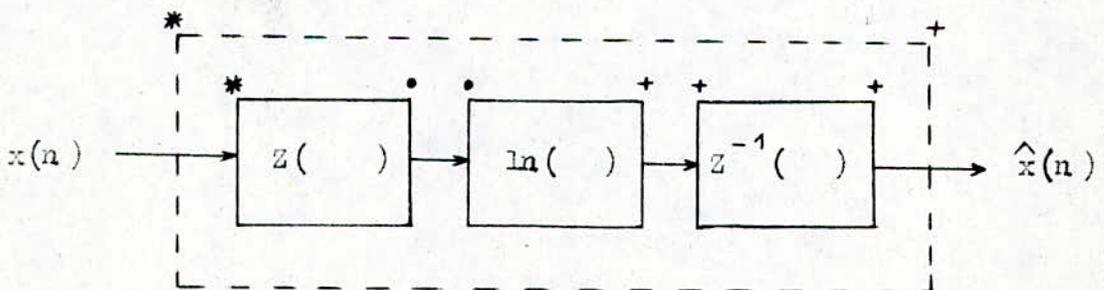


Fig. (2.2) Déconvolution homomorphique.

$$W_H = \begin{cases} \alpha + (1-\alpha) \cos(2\pi K/N) & \text{pour } |K| \leq N/2 \text{ et } \alpha = 0.54 \\ 0 & \text{ailleurs} \end{cases} \quad (2.2)$$

La fenêtre de Hamming (fig. 2.1.b) est un cas particulier de celle de Hanning ($\alpha = 0.50$) de forme sinusoidale est préférée et sera choisie par la suite. Car elle ne provoque l'apparition que de raies secondaires peu énergétiques dont les maxima décroissent vite. Toutefois la largeur du pic principal s'élargie (égale à $4/N$) par rapport à la fenêtre rectangulaire et se trouve ainsi multiplier par un facteur de 4.

2.2 Traitement homomorphique et cepstre ;

L'examen du spectre d'un son voisé montre que des ondulations, dont l'origine provient de la source d'excitation, modulent la forme du spectre caractéristique du son prononcé. L'idée est de séparer les deux composantes superposées au moyen d'un produit de convolution naturel en somme des deux composantes. Cela est réalisé par des traitements homomorphiques /14/ dont le calcul cepstral est une illustration. La figure 2.2 montre cette déconvolution.

2.3 Analyse cepstrale ;

le cepstre d'un signal est une fonction du temps définie soit comme la transformée de Fourier inverse du logarithme de la densité spectrale de puissance, soit comme le module carré de cette transformée. La figure 2.3, montre l'intérêt de la méthode. On y voit en a) l'allure du signal parole, en b) le logarithme de $|s(f)|^2$ c-à-d densité spectrale logarithmique. Il présente une périodicité spectrale, le long de l'axe des fréquences, résultant de la périodicité de la parole due au pitch, en c) le "cepstre" qui est une fonction d'une variable appelée quéfreance (mesurée en secondes).

Le cepstre complexe $y'(n)$ est définie comme étant la T.F inverse du logarithme du spectre d'un signal.

$$y'(n) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} Y(e^{j\omega}) e^{j\omega n} d\omega \quad (2.3)$$

Si seulement la partie réelle est utilisée dans le calcul, on

$$\text{obtient le cepstre } c(n) : c(n) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} \text{Log} |Y(e^{j\omega})| e^{j\omega n} d\omega \quad (2.4)$$

Le cepstre peut aussi être calculé par la transformée en cosinus du logarithme du spectre complexe du signal.

$$c(n) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} \text{Log} [Y(e^{j\omega})] \cos(\omega n) d\omega \quad (2.5)$$

2.3. Propriétés du cepstre :

En utilisant les fonctions rationnelles de la transformée en Z, équation (2.6) :

$$Y(Z) = \frac{A Z^r \prod_{k=1}^{M_1} [1 - a(k)Z^{-1}] \cdot \prod_{k=1}^{M_0} [1 - b(k).Z]}{\prod_{k=1}^{N_1} [1 - c(k).Z^{-1}] \cdot \prod_{k=1}^{N_0} [1 - d(k).Z]} \quad (2.6)$$

où

$$\begin{aligned} |a(k)| &; |c(k)| < 1 && \text{pôles et zéros intérieurs au cercle unité.} \\ |b(k)| &; |d(k)| < 1 && \text{" " " extérieurs " " " .} \end{aligned}$$

on obtient les relations suivantes :

$$\begin{cases} c(n) = 0 & \text{si } n < 0 \\ c(n) = y'(n) & \text{si } n = 0 \\ c(n) = y'(n) / 2 & \text{si } n > 0 \end{cases} \quad (2.7)$$

Donc le calcul du cepstre complexe revient ainsi à calculer le cepstre normal pour les signaux à phase minimale c-à-d ceux qui ne possèdent ni de pôles ni de zéros extérieurs au cercle unité.

2.3.2 Echelle MEL des coefficients cepstraux :

L'échelle spectrale dans le calcul du cepstre est linéaire en fréquence. Les études physiologiques et perceptives de l'oreille indiquent que cette dernière est sensible à une échelle quasi-logarithmique (dite échelle MEL) linéaire sur le premier 1 KHZ et logarithmique au delà (figure 2.4).

Les coefficients cepstraux peuvent être obtenus à l'aide d'un vocodeur à canaux dont les dix premiers filtres triangulaires

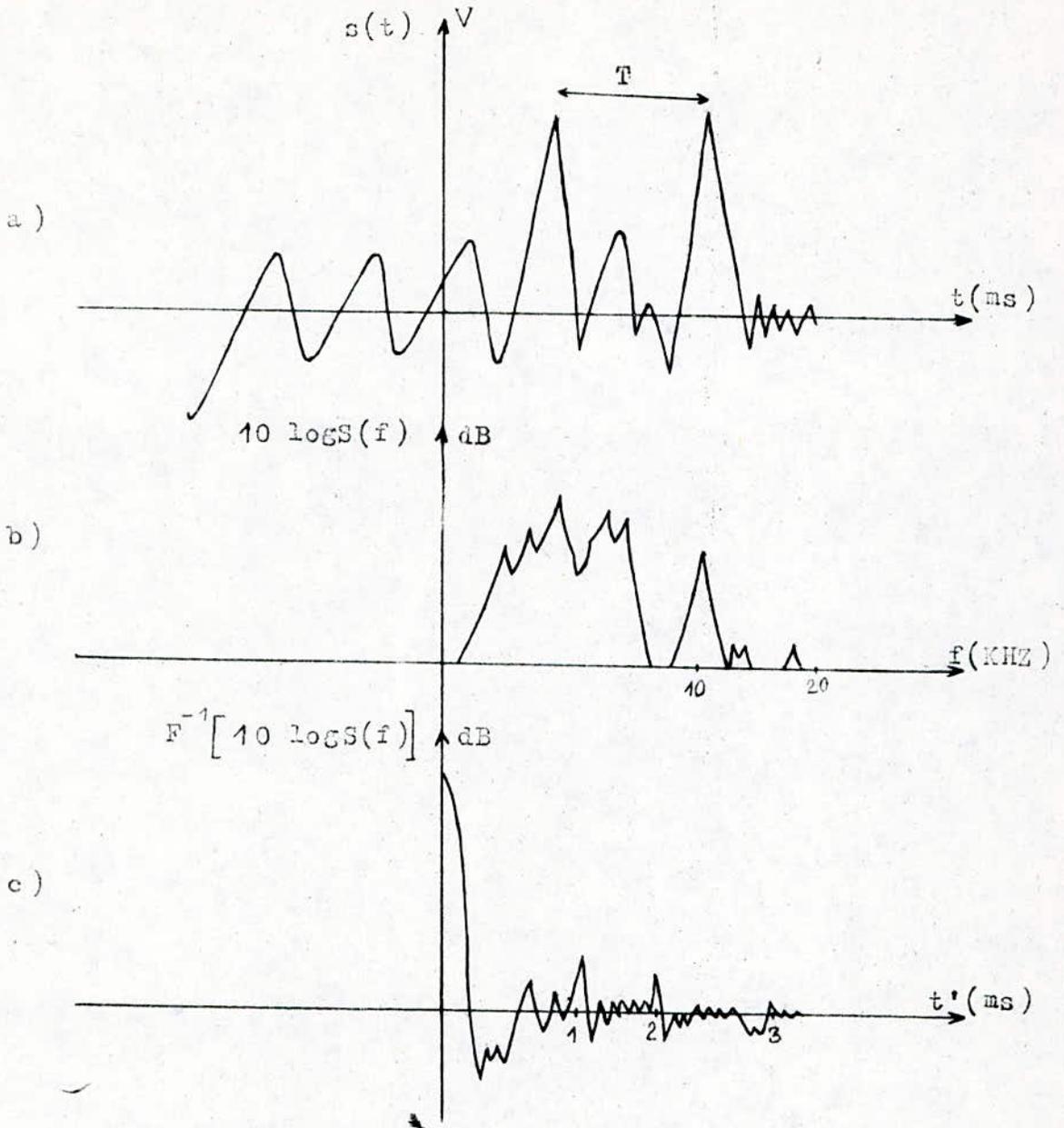


Fig. (2.3) Signal ; densité spectrale logarithmique et cepstre correspondant.

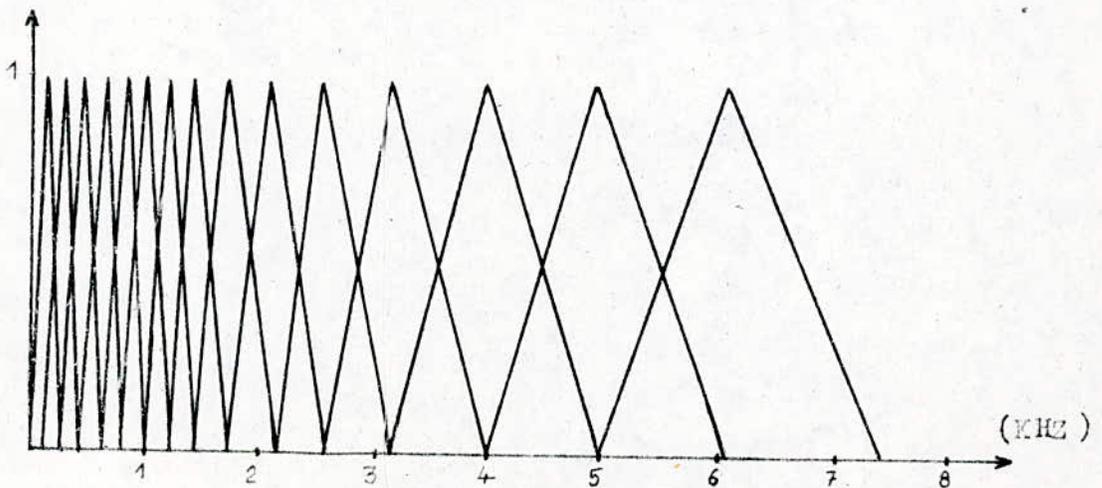


Fig. (2.4) Filtres triangulaire de l'échelle MEL.

ont une largeur uniforme et les 15 restants ont des largeurs qui croissent exponentiellement.

L'équation ci-dessous donne les coefficients cepstraux dans l'échelle MEL

$$MFCC(n) = \frac{1}{F} \sum_{k=1}^F \text{Log}[E(k)] \text{Cos}(n(k-1)/2.2\pi/F) \quad (2.8)$$

où n est le rang du coefficient cepstral, k le numéro d'un filtre d'énergie E(k) et F le nombre total des filtres.

2.4 Conclusion :

la méthode cepstrale sur une échelle MEL a été choisie car elle n'exige pas la présence du fondamental lui-même et s'accommode à la bande téléphonique normale. Elle permet en général une bonne séparation source / conduit vocal. Elle présente l'inconvénient de comporter deux T.F, ce qui allonge le temps de traitement, par contre les coefficients cepstraux ont l'avantage d'être normalisés en énergie. Peu de coefficients suffisent à bien représenter le spectre ce qui justifie leur emploi en reconnaissance.

CHAPITRE : 3

CHAPITRE : 3

APPRENTISSAGE

L'apprentissage est une étape qui suit l'analyse et consiste en la formation d'un dictionnaire qui regroupe un ensemble de référence valable.

Le nombre de références par mots correspond aux classes trouvées aux sein de l'ensemble des prononciations et le nombre de locuteurs qui utiliseront le système de reconnaissance sera évidemment beaucoup plus grand que le nombre de références par mot.

3. Analyse factorielle :

Le but de l'analyse factorielle est de représenter géométriquement, dans un sous-espace de faible dimension (par exemple, le plan $m=2$) des situations multidimensionnelles de telle sorte que l'information ne soit pas perdue. Les nouvelles composantes obtenues par l'analyse factorielle sont appelées composantes principales et permettent une analyse visuelle.

3.1.1 Principe de la méthode:

L'ensemble des I échantillons peut-être représenter dans un espace E de dimension q ($q \leq I-1$) qui conserve les distances initiales.

Soit $X_i, i=1, \dots, I$ l'ensemble des vecteurs dans E^q .

Faisons un changement d'origine en g, centre de gravité :

$$Y_i = X_i - g = X_i - \frac{1}{I} \sum_{i=1}^I X_i \quad (3.1)$$

Soit $V_1, \dots, V_m, \dots, V_q$ une base orthonormée de E^q

F^m le sous-espace engendré par V_1, \dots, V_m

$F^{m'}$ le sous-espace engendré par V_{m+1}, \dots, V_q

Comme F^m et $F^{m'}$ sont orthogonaux alors :

$$|Y_i|^2 = \sum_{k=1}^q |\langle Y_i, V_k \rangle|^2 = |\langle Y_i, F^m \rangle|^2 + |\langle Y_i, F^{m'} \rangle|^2 \quad (3.2)$$

où $\langle \cdot, * \rangle$ est la projection de \cdot sur $*$.

En sommant on a :

$$|Y_i|^2 = \sum_1 |\langle Y_i, F^m \rangle|^2 + \sum_1 |\langle Y_i, F^{m'} \rangle|^2 \quad (3.3)$$

On appelle $\sum_1 |Y_i|^2$ inertie du nuage par-rapport à g.

Ne pas perdre l'information c'est choisir un sous-espace de projection F^m dans lequel les distances seront conservées au mieux en prenant en moyenne les plus grandes. Pour cela on doit maximiser l'inertie du nuage projeté par rapport à g. Ce qui revient à rendre maximale la somme :

$$\sum_1 |\langle Y_i, F^m \rangle|^2 = \sum_{k=1}^m \sum_{i=1}^I |\langle Y_i, V_k \rangle|^2 = \sum_{k=1}^m V_k^t \cdot Y^t \cdot Y \cdot V_k \quad (3.4)$$

où Y est la matrice à I lignes et q colonne dans laquelle la i^{ème} ligne contient les q composantes du vecteur Y_i .

L'exposant t indique la matrice transposée.

V_k sont les inconnus à déterminer avec $V_j^t \cdot V_j = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases}$

Si $\lambda_1, \lambda_2, \dots, \lambda_m, \dots, \lambda_q$ sont les valeurs propres par ordre décroissant de $Y^t \cdot Y$ et $V_1, \dots, V_m, \dots, V_q$ les vecteurs propres associés, alors V_1, \dots, V_m est la solution de l'équation (3.4) et par suite le meilleur sous-espace de dimension m pour la représentation des I échantillons est celui formé par les m vecteurs suivants: $\sqrt{\lambda_1} V_1, \dots, \sqrt{\lambda_m} V_m$.

Le facteur t_m , équation (3.5), permet de mesurer la quantité d'information conservée.

$$t_m = \frac{\lambda_1 + \lambda_2 + \dots + \lambda_m}{\lambda_1 + \lambda_2 + \dots + \lambda_q} \quad 0 \leq t_m \leq 1 \quad (3.5)$$

Si $t_m = 1$, il n'y a aucune perte d'information par suite de la réduction de données en passant de E^q à F^m .

3.1.2 Recherche de la solution :

Soit la distance $D(i, j)$ entre les échantillons. Partant de la relation (3.6), on obtient facilement l'équation (3.7), donnant les éléments $\gamma_i \cdot \gamma_j$ de la matrice $S = Y^t \cdot Y$, rien qu'en fonction des distances mutuelles.

$$X_i \cdot X_j = -\frac{1}{2} \left\{ D(i, j)^2 - X_i^2 - X_j^2 \right\} \quad (3.6)$$

$$Y_i \cdot Y_j = -\frac{1}{2} D(i, j)^2 + \frac{1}{2I} \sum_k D(i, k)^2 + \frac{1}{2I} \sum_k D(k, j)^2 - \frac{1}{2I^2} \sum_k \sum_l D(k, l) \quad (3.7)$$

Puisque S est semi-définie, on peut alors l'écrire sous la forme :

$$S = Q^t \cdot A^2 \cdot Q \quad (3.8)$$

où A est une matrice diagonale à valeurs propres positives et Q une matrice orthogonale. Il suffit alors de prendre comme solution pour Y :

$$Y = A \cdot Q \quad (3.9)$$

3.1.3 Conclusion :

L'analyse factorielle détermine les paramètres ou facteurs permettant de condenser les informations. Elle permet, en particulier de réduire à un plan la dimension de l'espace des variables étudiées, ceci est très important pour les méthodes de programmation dynamique où les problèmes d'encombrement mémoire sont cruciaux.

L'utilisation de l'analyse factorielle pour visualiser les prononciations des mots de divers locuteurs met en évidence l'existence de classes de prononciations pour un même mot.

3.2 Classification :

Le terme de classification est employé pour désigner le partage d'un ensemble d'individus en classe de telle sorte que tout individu appartienne à une classe et une seule, c'est ce qu'on appelle partition mais le terme de classification sert aussi à désigner, des systèmes emboîtés de classe, on parle alors de classification hiérarchique /21/. Le but de cette classification est la recherche de représentants de chaque mot dans les classes trouvées ce qui impose le choix des méthodes de partitionnement.

On distingue deux types d'algorithmes de classification par partitionnement :

- Les algorithmes parallèles (ou par échange) où les classes sont cherchées et produites simultanément.
- Les algorithmes séquentiels où les classes sont cherchées et

produites l'une après l'autre sans que leurs nombre ne soit fixé à priori.

Plusieurs expérience /21/ ont montrées la supériorité des algorithmes d'échanges basés sur une fonction critère(A.E.C).

3.2.1 Notions mathématiques :

désigne la distance dynamique entre les matrices correspondant aux éléments X et X .

- Soit n_k le nombre d'éléments dans la classe $C_k = \{ \dots, X_i, \dots, X_j, \dots \}$ on définit la moyenne des distances de X_i à tous les X_j par :

$$L(X_i, C_k) = \frac{1}{n_k - 1} \sum_{X_j \in C_k} D(X_i, X_j) \quad (3.10)$$

- Le minimum des distances moyennes d'un élément à tous les autres au sein d'une classe C_k , représente la fonction d'homogénéité $H_{(C_k)}$:

$$H_{(C_k)} = \min_{X_i \in C_k} L(X_i, C_k) \quad (3.11)$$

La fonction d'homogénéité sera d'autant plus petite que la dispersion des éléments de la classe est petite est vice-versa.

- Le représentant $R_{(C_k)}$ est l'élément qui minimise la distance d'un élément à tous les autres.

$$R_{(C_k)} = X_i^* / \min_{X_i \in C_k} L(X_i, C_k) = L(X_i^*, C_k) \quad (3.12)$$

- La fonction-critère F est une grandeur qui mesure la qualité d'une partition. On la définit comme étant la somme des fonctions d'homogénéité $H_{(C_k)}$ pondérées par $n_k \cdot (n_k - 1)$:

$$F = \sum_k H_{(C_k)} \cdot n_k \cdot (n_k - 1) \quad (3.13)$$

Quand la fonction critère diminue la qualité augmente.

3.2.2 L'algorithme d'échange basé sur une fonction critère (A.E.C) :

Etape 1 : choisir une partition initiale

- " 2 : chercher $X_i \in C$ pour lequel il existe une classe C_1 telle que le transfert de X_i de sa classe C_k vers la classe C_1 doit minimiser (diminuer) la fonction critère F : $\Delta F < 0$
- " 3 : arrêter si aucun élément n'a été trouvé :

$$\forall X_i \in C, l=1, \dots, K : \Delta F(X_i, C_k \rightarrow C_l) \geq 0$$

Etape 4 : transférer X_i dans la classe C_l qui donne la plus grande décroissance :

$$\Delta F(X_i, C_k \rightarrow C_l) = \min_l \Delta F(X_i, C_k \rightarrow C_l)$$

" 5 ; aller à l'étape 2 .

Partant d'une partition k quelconque, l'algorithme ci-dessus cherche à améliorer cette partition en transférant les éléments d'une classe à l'autre tout en minimisant la fonction critère F .

Il convient donc, de transférer X_i si et seulement si :

$$\Delta F(X_i, C_k \rightarrow C_l) < 0$$

Le nombre de classe n'est pas affecté puisque les classes ne doivent pas être vides. On s'arrête quand on ne peut plus échanger d'éléments.

$$\Delta F(X_i, C_k \rightarrow C_l) \geq 0$$

3.2.3 Conclusion :

L'algorithme A.E.C a été testé et comparé à d'autres algorithmes tels que : l'algorithme " Basic Isodata ", l'algorithme séquentiel basé sur une fonction critère... /21/

Les résultats obtenus montre une supériorité nette de l'algorithme A.E.C et qu'en plus il n'est pas perturbé par les isolés alors que les autres le sont plus ou moins.

CHAPITRE : 4

CHAPITRE : 4

PROGRAMMATION DYNAMIQUE

Les méthodes de calcul de distances effectuent soit un alignement temporel linéaire ce qui est insuffisant dans le cas de fortes distorsions entre les événements vocaux de deux prononciations différentes d'un mot donné; soit un alignement temporel non linéaire basé sur le principe de la comparaison dynamique.

La phase de comparaison est réalisée en étirant et en comprimant les empreintes en mémoire suivant une méthode d'optimisation qu'on appelle programmation dynamique. Celle-ci tente pour chaque empreinte, d'associer chaque intervalle du mot prononcé avec un intervalle de l'empreinte de façon à minimiser l'écart entre les deux.

La programmation dynamique est basée sur le principe d'optimisation de R. Bellman /4 / énoncé dans son livre comme suit :
" une politique est optimale si, à une période donnée, quelque que soient les décisions précédentes, les décisions qui restent à prendre constituent une politique optimale en regard du résultat des décisions précédentes."

Ce qui veut dire tout simplement : " Toute sous-stratégie d'une stratégie optimale est elle même optimale."

4. Distance cepstrale :

On définit la distance cepstrale comme suit :

$$d_0(f_1, f_2) = \sum_{h=-\infty}^{+\infty} (c_h(1) - c_h(2)) \quad (4.1)$$

L'intérêt de cette métrique tient au fait qu'on peut se limiter à une somme finie ($h = -m, \dots, +m$) avec une excellente approximation de l'ordre de 10^3 dès que $m > 2p$, si p est l'ordre du modèle.

Pour simplifier on prendra par la suite l'écart absolu entre les coefficients cepstraux des deux signaux à comparer comme distan-

ce équivalente à d_0 :
$$d = \sum_{h=1}^N \left| c_h^{(1)} - c_h^{(2)} \right| \quad (4.2)$$

où N : nombre de coefficients cepstraux.

4.2 Fonction de coïncidence :

On appelle fonction de coïncidence (ou de déformation) F , le chemin d'alignement optimal qui relie l'ensemble des points de l'espace de recherche.

Soit le plan constitué par les deux axes temporels I et J . Les vecteurs A et B sont représentés par leurs indices $(a_1, \dots, a_i, \dots, a_I)$ et $(b_1, \dots, b_j, \dots, b_J)$ le long des axes I et J (fig. 4.1)

$$F = c_{(1)}, c_{(2)}, \dots, c_{(k)} \quad \text{avec } c_h (i_{(h)}, j_{(h)}) \quad (4.3)$$

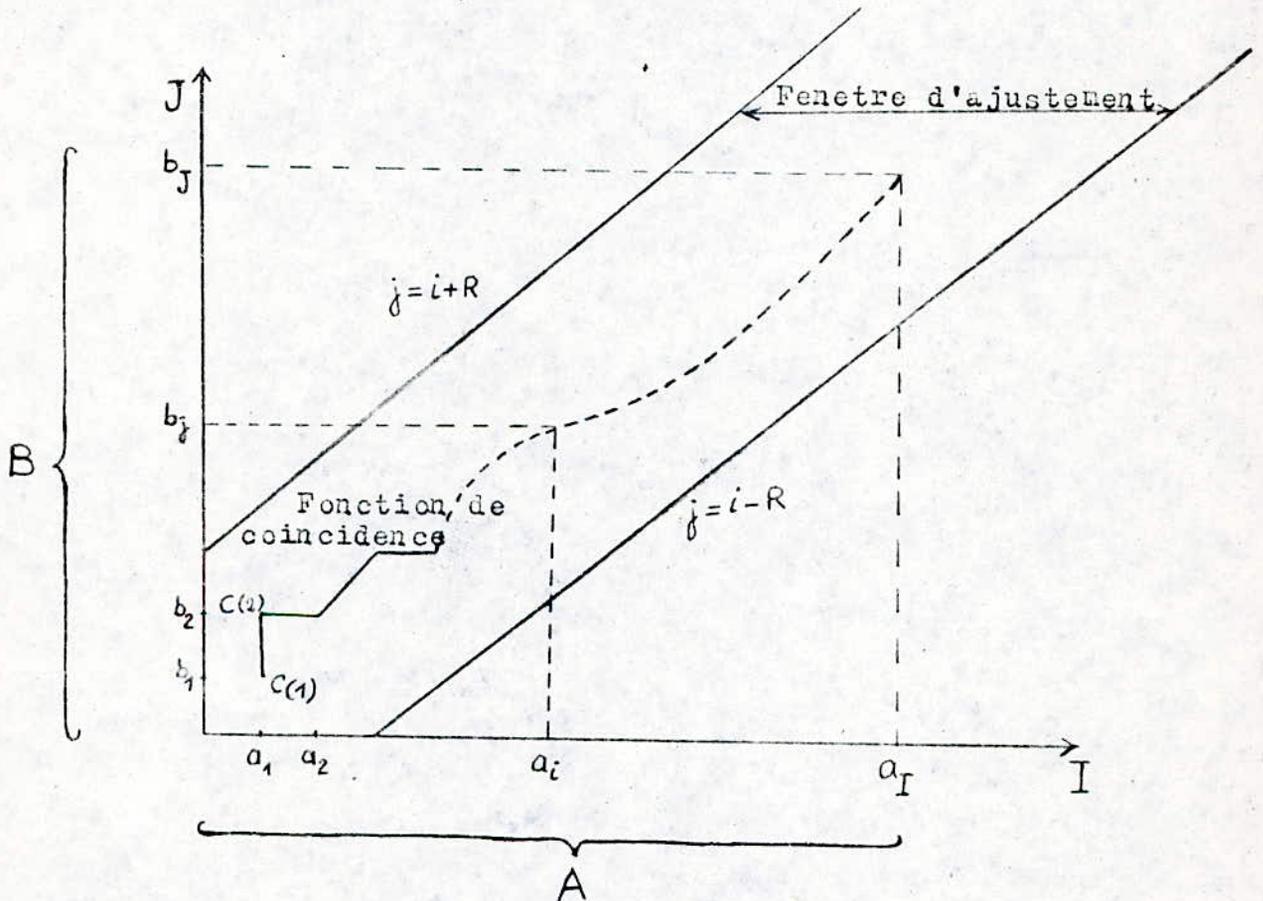


Fig. 4.1 Alignement temporel dynamique

4.3 Fonction de pondération :

Une pondération $W(i)$ s'avère nécessaire pour compenser l'effet du nombre de points le long de la fonction de coïncidence F .

Pour pouvoir appliquer la technique de programmation dynamique,

il suffit de rendre $\sum_{k=1}^K W(k) = N$ indépendant de la fonction F.

Il existe deux choix typiques :

1) Forme symétrique :

$$W(k) = (i(k) - i(k-1)) + (j(k) - j(k-1)) \quad (4.4)$$

ce qui donne $N = I + J$

où I et J sont les longueurs respectives des échantillons A et B de parole.

2) Forme asymétrique :

$$W(k) = (i(k) - i(k-1)) \quad \text{d'où } N = I \quad (4.5)$$

$$\text{ou } W(k) = (j(k) - j(k-1)) \quad \text{d'où } N = J$$

La forme symétrique agit sur les deux axes et donne des meilleurs taux de reconnaissance que ceux de la forme asymétrique.

4.4 Contraintes :

La fonction de coïncidence doit respecter les caractéristiques essentielles de l'évolution du signal parole. Pour cela Sakoe-Chiba ont proposés les conditions suivantes :

- Conditions de monotonie et de continuité :

avec $C(k) = (i(k), j(k))$ on a pour $C(k-1)$ les possibilités, voir figure 4.2, suivantes :

$$C(k-1) = \begin{cases} (i(k), j(k)-1) \\ \text{ou } (i(k)-1, j(k)-1) \\ \text{ou } (i(k)-1, j(k)) \end{cases} \quad (4.6)$$

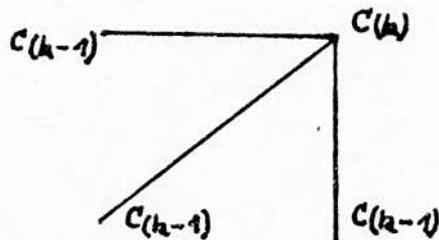


Fig. 4.2

- Limitation :

$$\begin{array}{l} i(1) = 1 \\ j(1) = 1 \end{array} \quad \text{et} \quad \begin{array}{l} i(K) = I \\ j(K) = J \end{array} \quad (4.7)$$

- Conditions de pente :

Si un point $C(k)$ se déplace dans la direction de l'axe i (ou j) m fois consécutives, il ne peut pas faire un pas supplémentaire dans la même direction avant d'en faire n pas dans la direction de la diagonale (fig. 4.3). Cette condition de pente est donnée par le rapport :

$$P = n/m$$

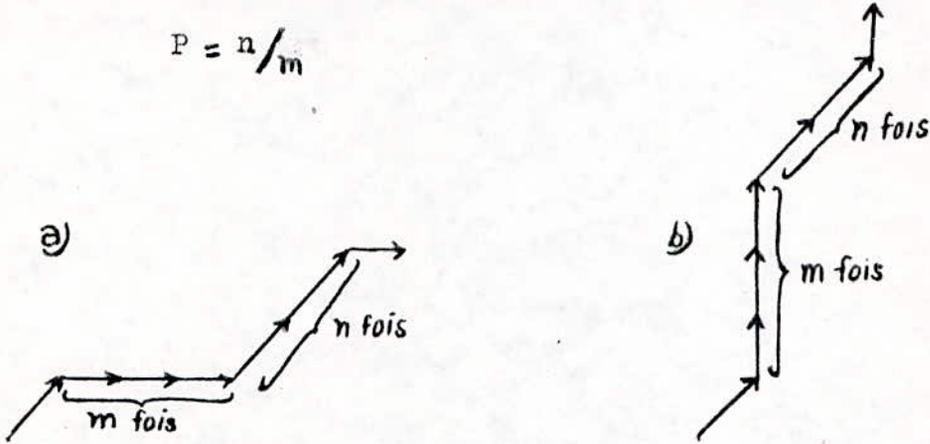


fig. 4.3 a) Minimum de pente
b) Maximum de pente

- Ajustement :

$$|i(k) - j(k)| \leq R \quad ; \quad R \geq 0 \quad (4.8)$$

R définit une fenêtre d'ajustement temporelle tenant compte du fait que les fluctuations de durée et de rythme entre deux mots restent toujours limitées.

Les conditions ci-dessus impliquent des contraintes locales et globales.

* Les contraintes locales :

Elles ont pour rôle la limitation des chemins admissibles qui arrivent en un point donné. Sakoe-Chiba ont proposés les contraintes locales symétriques, voir la figure 4.4 (les chiffres 1 et 2 représentent les pondérations qui gardent constante la longueur effective du mot test.).

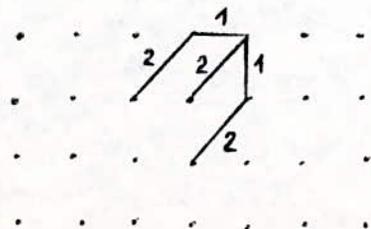


Fig. 4.4 Contraintes locales selon Sakoe et Chiba.

* Les contraintes globales :

Elles réduisent l'espace de recherche en rejetant les formes de référence improbables (5.6.1). Le choix d'un rapport maximal de deux ou minimal de un-demi entre les durées réelles des mots à comparer paraît raisonnable.

4.5 Programmation dynamique :

Soit $d(C_{(h)}) = d(i_{(h)}, j_{(h)}) = d(A, B)$ la distance entre le couple de mots A et B, et soit la distance $D^k(C)$ définie par :

$$D^k(C) = \sum_{h=1}^k d(C_{(h)}) \cdot W_{(h)} \quad (4.9)$$

k étant le nombre de couples mis en correspondances.

$D^k(C)$ atteint son minimum quand l'alignement temporel est réalisé de façon optimale. Ainsi, on définit la distance entre les mots A et B par la relation suivante :

$$D(A, B) = \min_F \left(\sum_{h=1}^k d(C_{(h)}) \cdot W_{(h)} \right) / \sum_{h=1}^k W_{(h)} = \frac{1}{N} \min_F \left\{ \sum_{h=1}^k d(C_{(h)}) \cdot W_{(h)} \right\} \quad (4.10)$$

On peut alors appliquer le théorème d'optimalité à la fonction $D(A, B)$ comme suit :

$$\begin{aligned} * \text{ Min}_{F(1), \dots, F(k)} \sum_{h=1}^k d(C_{(h)}) \cdot W_{(h)} &= \\ = \text{ Min}_{F(1), \dots, F(k-1)} \sum_{h=1}^{k-1} [d(C_{(h)}) \cdot W_{(h)}] + \min_{F(k)} d(C_{(k)}) \cdot W_{(k)} & \\ ** \text{ Min}_{F(1), \dots, F(k-1)} \sum_{h=1}^{k-1} d(C_{(h)}) \cdot W_{(h)} &= \\ = \text{ Min}_{F(1), \dots, F(k-2)} \sum_{h=1}^{k-2} d(C_{(h)}) \cdot W_{(h)} + \min_{F(k-1)} d(C_{(k-1)}) \cdot W_{(k-1)} & \end{aligned} \quad (4.11)$$

Et ainsi de suite ...

En utilisant les contraintes de la figure 4.4 et une fonction de pondération symétrique, le calcul de la distance est déterminée récursivement selon l'algorithme ci-dessous :

- Condition initiale :

$$\epsilon_1(C_{(1)}) = d(C_{(1)}) \cdot W_{(1)} \quad (4.2)$$

- L'équation de programmation dynamique est :

$$\epsilon_h(C_{(h)}) = \min_{C_{(h-1)}} \left[\epsilon_{h-1}(C_{(h-1)}) + d(C_{(h)}) \cdot W_{(h)} \right] \quad (4.13)$$

- La distance normalisée est alors :

$$D(A,B) = \frac{1}{N} \sum_k g_k(C(k))$$

Quand la distance $D(A,B)$ dépasse une certaine valeur cela veut dire que les deux mots sont différents et qu'il n'est pas nécessaire de continuer la comparaison.

Les résultats sont très intéressants mais le temps de calcul est assez grand. Certaines modifications /22/, /25/ permette d'alléger énormément ce temps de calcul

CHAPITRE : 5

CHAPITRE : 5

RECONNAISSANCE DES MOTS CONNECTES

Soit une forme de test inconnue $T(m)$ $m = 1, 2, \dots, M$ où pour chaque valeur m , $T(m)$ dénote un vecteur de caractéristiques et M est la longueur de T en trames (ou échantillons). Supposons que T est enregistré avec une séquence de L formes de référence $R_{q(1)}^{(n)}, \dots, R_{q(L)}^{(n)}$, où chaque $R_{q(k)}$ ($k = 1, 2, \dots, L$) fait partie d'un groupe de V formes de référence R_v .

On définira une super forme de référence $R^S_{q(1)\dots q(L)}$ (qu'on notera R^S) comme étant la concaténation des L formes de référence.

$$R^S = R_{q(1)} \oplus R_{q(2)} \oplus R_{q(3)} \oplus \dots \oplus R_{q(L)} \quad (5.1)$$

L'objectif recherché est d'appliquer la DTW pour trouver la meilleur super forme \bar{R}^S qui s'adapte, de façon optimale, à une forme de test donnée.

5.1 L'alignement DTW entre T et R^S :

En désignant par W le chemin d'alignement temporel optimum, on a alors $n = W(m)$ qui représente un cadrage fonctionnel entre la trame test m et la trame n de R^S .

Quand $q(1)$ est connu pour tout l , R^S peut être traitée comme une seule forme de référence et la détermination de W peut alors se faire en utilisant une seule application d'un algorithme de DTW avec contrainte sur les bornes (fig. 5.1).

L'algorithme DTW doit minimiser la distance globale D définie par l'équation (5.2) à l'aide de la distance locale $d(m, n)$ entre les trames m et n avec les contraintes (5.3) sur les bornes.

$$D = \min_{W(m)} \left[\sum_{m=1}^M d(m, W(m)) \right] \quad (5.2)$$

$$W(1) = 1 \quad \text{et} \quad W(M) = L \quad (5.3)$$

où $\phi(1)$ est la longueur en trame de l formes de référence concaténées

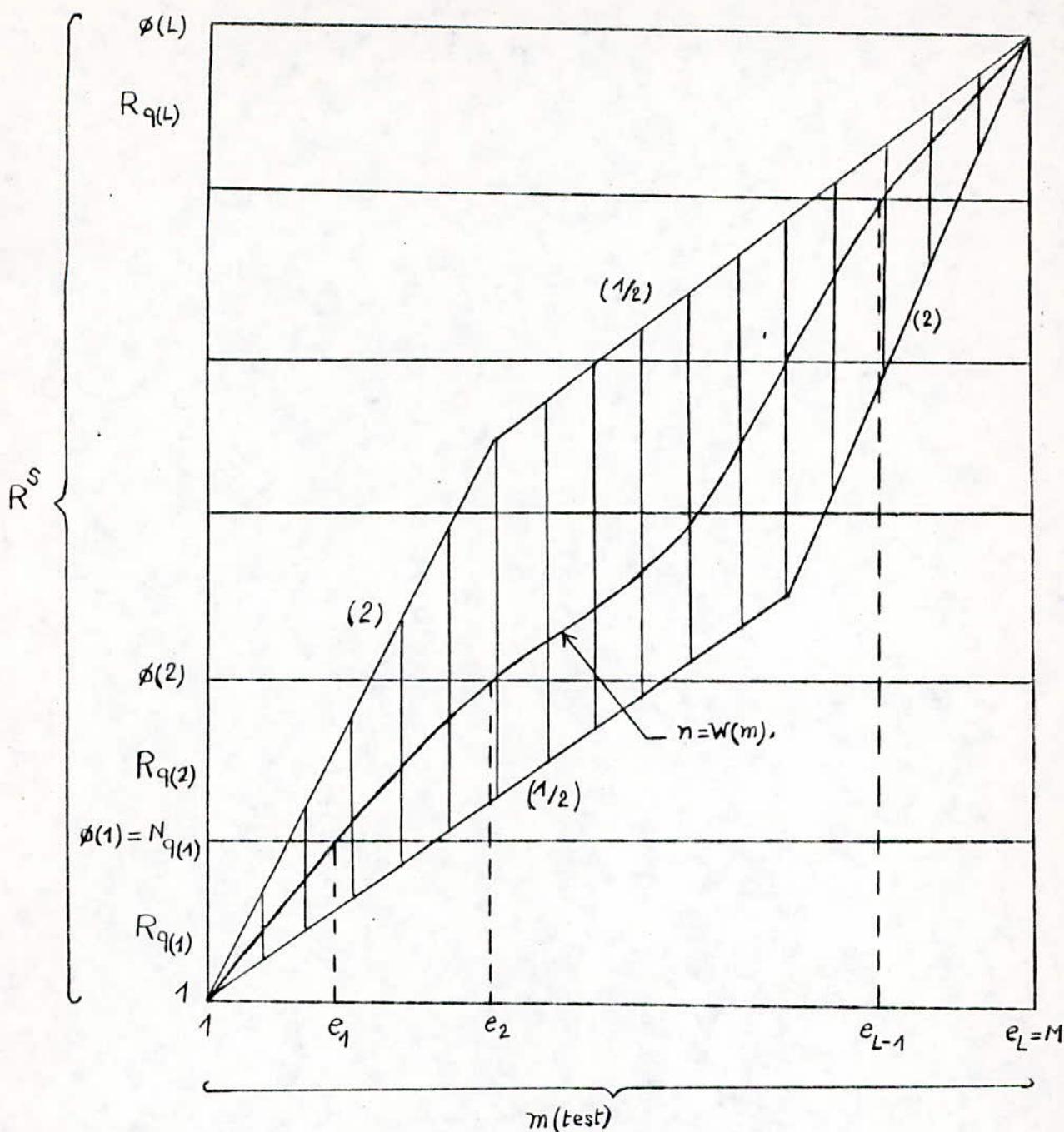


Figure 5.1 Alignement D.T.W typique entre T et R^S avec les contraintes de pente.

5.2 Préléminaires :

Au lieu de considérer la super forme comme une seule forme de référence, on applique l'algorithme DTW pour chaque référence; l'algorithme ainsi obtenu est appelé level building dynamic time warp (LB-DTW). Pour la 1^{ème} référence de l'ensemble des distances accumulées le long de la ligne ($n = \phi(1)$) est alors sauvgardé pour l'initia-

lisation du niveau suivant. Ainsi nous sommes amenés à définir d'autres distances :

$$* d_{\ell}(m, n) = d(m, n + \phi(1-1)) \quad (5.4)$$

Distance locale à la trame m du test et la trame n de la 1^{ème} de la super forme de référence.

$$* D_{\ell}(m, n) = D_A(m, n + \phi(1-1)) \quad (5.5)$$

Distance accumulée à la trame m du test et la trame n de la 1^{ème} référence de la super forme de référence.

$$* \bar{D}_{\ell}(m) = D_{\ell}(m, N_{q(\ell)}) = D_A(m, \phi(1)) \quad (5.6)$$

$$\text{avec } \bar{D}_0(m) = \begin{cases} 0 & \text{pour } m = 0 \\ \infty & \text{pour } m \neq 0 \end{cases}$$

Distance accumulée à la trame m du test et la dernière trame de la 1^{ème} référence de la super forme de référence.

$$* \bar{D}_{\ell}^B(m) = \min_{q(1) \dots q(\ell)} (D_{q(1) \dots q(\ell)}(m)) \quad (5.7)$$

$$\text{avec } \bar{D}_0^B(m) = \begin{cases} 0 & \text{pour } m = 0 \\ \infty & \text{pour } m \neq 0 \end{cases}$$

Distance accumulée à la m^{ème} trame du test et à la fin de la 1^{ème} forme de référence de la meilleure super forme de référence partielle (à la trame m). Autrement dit la valeur minimal de $\bar{D}_{\ell}(m)$ pour toutes les super forme de référence de la longueur l possible.

$D_{q(1)q(2)\dots q(\ell)}(m)$ désigne la distance entre la forme de test et la super forme de référence $R_{q(1)} + R_{q(2)} + \dots + R_{q(\ell)}$ à la trame m du test.

5.3 Comparaison de la forme test à plusieurs formes de référence ;

Dans ce cas on doit minimiser la distance D^* , définie par (5.8) et ceci à chaque niveau l.

$$D^* = \min_{L_{min} \leq L \leq L_{max}} \left[\min_{q(1) \dots q(L)} (D_{q(1) \dots q(L)}(M)) \right] \quad (5.8)$$

Théorème :

S'il existe une super forme de référence $R_{s_1 s_2 \dots s_L}^S$ telle que :

$$D_{r_1 r_2 \dots r_\ell} (e_\ell) \leq D_{s_1 s_2 \dots s_\ell} (e_\ell) \quad \forall R_{s_1 s_2 \dots s_\ell}^S \quad \text{alors}$$

$$D_{r_1 r_2 \dots r_\ell r_{\ell+1}} (e_{\ell+1}) \leq D_{s_1 s_2 \dots s_\ell r_{\ell+1}} (e_{\ell+1}) \quad \forall R_{s_1 s_2 \dots s_\ell}^S \quad \text{et } R_{r_{\ell+1}}$$

Ce théorème permet de trouver la meilleure super forme de référence \bar{R}^S de longueur L. Comme les frontières e_1, e_2, \dots, e_L ne sont pas connues à chaque niveau, on utilise donc la distance $\bar{D}_\ell^B(m)$, (5.7).

La figure (5.2) résume tout ce qui a été dit précédemment. on admet que L est connu (L=4), que W(m) est astreint à être dans le parallélogramme et que v=2, c-à-d qu'il existe seulement deux formes de référence à la m^{ème} trame $R_{W_\ell(m)}$, et le meilleur chemin à partir du niveau précédent. Par exemple, au premier niveau, la forme de référence peut finir à n'importe laquelle des 6 différentes trames le long du test. La trame où elle prend fin est considérée comme la meilleure forme de référence (B dans le cas de la figure 5.2).

A chaque niveau, pour toutes les trames de fins possibles, la meilleure référence est retenue. Jusqu'à ce qu'au niveau 4 une unique meilleur référence est obtenu.

Pour former la meilleur super forme de référence, on concatène les références obtenues en faisant un chainage arrière "backtracking" à partir de la fin du test ($R^S = R_B \oplus R_A \oplus R_A \oplus R_B$) (fig. 5.2). Les frontières e_1, e_2, e_3 et $e_4 = M$ le long du test sont alors déterminées.

5.4 Le backtracking :

Il est admis que le chemin optimal vert le point (m,n) au 1^{ème} niveau est venu du point $(e_{\ell-1}, \phi(1-1))$ à la fin du (1-1)^{ème} niveau (fig. 5.3). Ainsi on définit un réseau $F_\ell(m,n)$, comme étant la valeur $e_{\ell-1}$ à partir de laquelle le meilleur chemin vers (m,n) au niveau 1 utilisant la référence q(1), est obtenu.

Il est donc permis d'appliquer le backtracking avec cependant les définitions suivantes :

- On désigne par F (m,n) le poiteur en un point (m,n) du niveau 1, de la valeur initiale $e_{\ell-1}$ du niveau 1-1 à partir de laquelle le

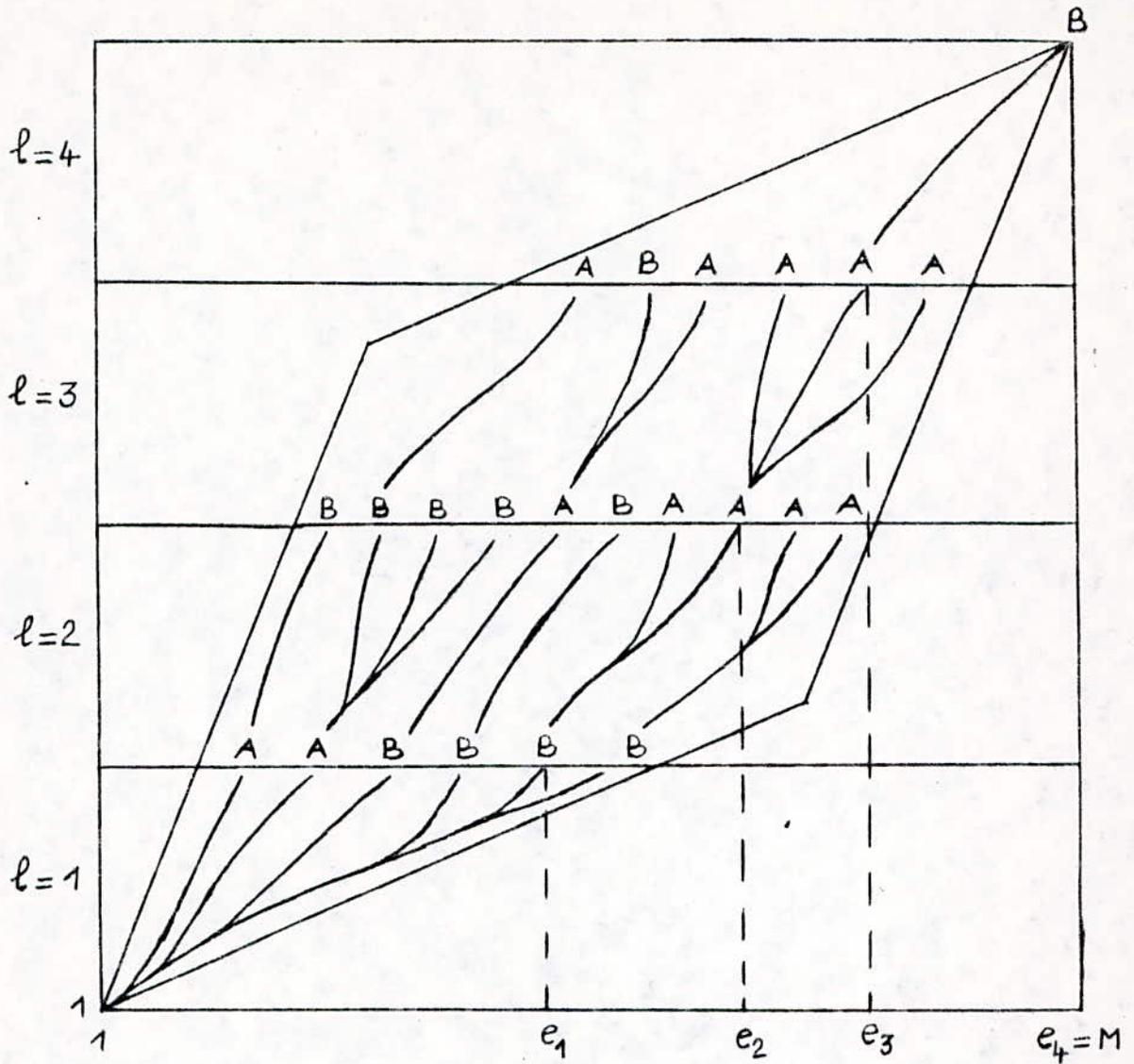


Figure 5.2 Illustration du backtracking permettant de retrouver le meilleur chemin, à partir de la fin, sur 4 niveau DTW pour 2 formes A et B.

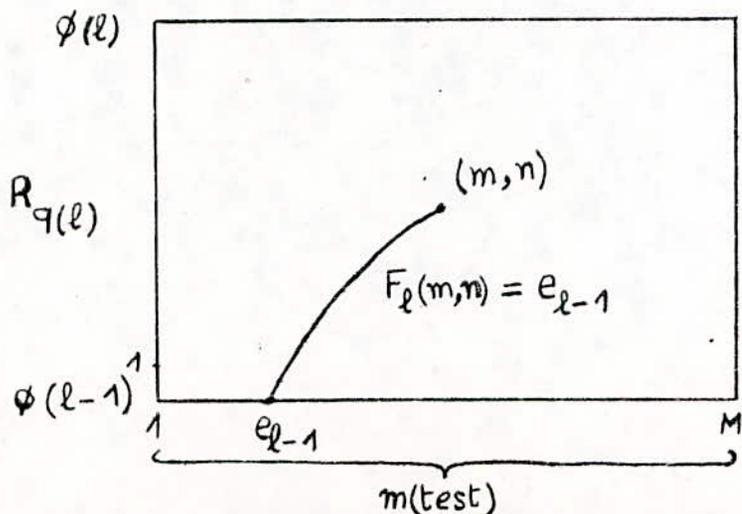


Figure 5.3 Backtracking (pointeur arrière)

meilleur chemin vers (m, n) est venu.

- De même $F_\ell(m)$ reflète la position le long de la forme du test, à la trame m du niveau l , à partir de laquelle le meilleur chemin vers le point (m, N_V) est venu.

- La valeur de $\bar{F}_\ell(m)$ associée au mot de référence V qui donne $\bar{D}^B(m)$ sera notée $\bar{F}_\ell^B(m)$

- on définit aussi $\bar{F}_\ell^V(m)$ comme étant la valeur de $\bar{F}_\ell(m)$ pour la référence R_V

- $\text{Argmin } f(x)$ est la valeur de x qui minimise $f(x)$

- Les fonctions limites inférieure et supérieure sont indiquées respectivement par $L(m)$ et $U(m)$, alors que les contraintes inférieure et supérieure de la fonction de déformation W sont dénotées respectivement par $\bar{L}(n)$ et $\bar{U}(n)$.

5.5 L'algorithme LB-DTW :

1°) Initialiser $\bar{D}_0^B(m) = 0$ pour $m = 0$

$\bar{D}_0^B(m) = \infty$ pour $m = 1, 2, \dots, M$

et $\bar{D}_\ell^B(0) = \infty$ pour $l = 1, 2, \dots, L$

2°) Pour $l = 1, 2, \dots, L$ faire les étapes de 3°) à 7°)

3°) Pour $v = 1, 2, \dots, V$ faire les étapes de 4°) à 6°)

4°) Pour $m = 0, 1, 2, \dots, M$ initialiser $D_\ell(m, 0) = \bar{D}_{\ell-1}^B(m)$ et $F_\ell(m, 0) = m$

5°) Pour $m = 1, 2, \dots, M$ et $n \in (L_\ell(m), \dots, U_\ell(m))$ calculer en utilisant $R = R_V$:

$$\hat{n} = \underset{\bar{L}(n) \leq n' \leq \bar{U}(n)}{\text{argmin}} (D_\ell(m-1, n'))$$

$$D_\ell(m, n) = d_\ell(m, n) + D_\ell(m-1, \hat{n})$$

$$F_\ell(m, n) = F_\ell(m-1, \hat{n})$$

6°) Pour $m = 1, 2, \dots, M$ calculer :

$$\bar{D}_\ell^V(m) = D_\ell(m, N_V)$$

$$\bar{F}_\ell^V(m) = F_\ell(m, N_V)$$

7°) Pour $m = 1, 2, \dots, M$ calculer :

$$\bar{D}_l^B(m) = \min_{1 \leq v \leq V} (\bar{D}_l^V(m))$$

$$W_l(m) = \operatorname{argmin}_{1 \leq v \leq V} (\bar{D}_l^V(m))$$

$$\bar{F}_l^B(m) = \bar{F}_l^{W_l(m)}(m)$$

$$8^\circ) \quad L = \operatorname{argmin}_{L_{\min} \leq l \leq L_{\max}} (\bar{D}_l^B(M))$$

$$D = \bar{D}_L^B(M)$$

$$9^\circ) \quad e_L = M$$

$$10^\circ) \quad e_l = \bar{F}_{l+1}^B(e_{l+1}), \quad l = L-1, L-2, \dots, 1.$$

$$11^\circ) \quad \bar{R}^S = R_{W_1(e_1)} \oplus R_{W_2(e_2)} \oplus R_{W_3(e_3)} \oplus \dots \oplus R_{W_L(e_L)}$$

L'algorithme IB-DTW donne le nombre optimal de formes de références, L , la distance minimale D de la chaîne, les références dans la chaîne $R_{W_1(e_1)}, \dots, R_{W_L(e_L)}$ et le groupe de points frontières e_l , qui adaptent la dernière trame de référence dans la chaîne.

5.6 Certaines modifications de l'algorithme IB-DTW ;

Pour pouvoir faire une reconnaissance en temps réel certaines modifications s'avèrent nécessaire.

5.6. Réduction du champ de recherche de la fonction de déformation ;

On choisira les contraintes d'ITAKURA car elles réalisent une économie de traitement et assurent la stabilité de W quand m augmente.

- Contraintes d'ITAKURA ;

$$\begin{cases} 0 \leq W(m) - W(m-1) \leq 2 \\ 1 \leq W(m) - W(m+2) \end{cases} \quad (5.9)$$

- Les contraintes locales inférieure et supérieure sur la fonction de déformation seront ;

$$\bar{L}(n) = \max(0, n-2) \quad (5.10)$$

$$\bar{U}(n) = \begin{cases} n & \text{si } W(m-1) \neq W(m-2) \\ n-1 & \text{si } W(m-1) = W(m-2) \end{cases}$$

- Les contraintes aux limites ;

$$L(m) = \max\left(\frac{m+2}{2}, 2(m-M) + \phi(1)\right) \quad (5.11)$$

$$U(m) = \min\left(2(m-1) + 1, \frac{1}{2}(m-M) + \phi(L_{\max})\right)$$

La région obtenue n'est plus un losange (région hachurée voir fig. 5.4). Ainsi pour une chaîne de longueur $L = L_{\max}$ et une fonction de déformation sans contraintes, la zone totale moyenne est :

$$A_T = L_{\max} \bar{N} \cdot M$$

où \bar{N} est la longueur moyenne d'une forme de référence. Alors que dans la région hachurée de la figure 5.4 la zone totale moyenne réduite est approximativement :

$$A_R = L_{\max} \bar{N} \cdot M / 3$$

En admet que $\bar{N} \approx M / L_{\max}$ c-à-d que la longueur test est L_{\max} fois la longueur de référence moyenne.

On remarque donc qu'une réduction dans les calculs, d'environ $1/3$, est obtenue en utilisant d'une manière appropriée les contraintes supérieure et inférieure.

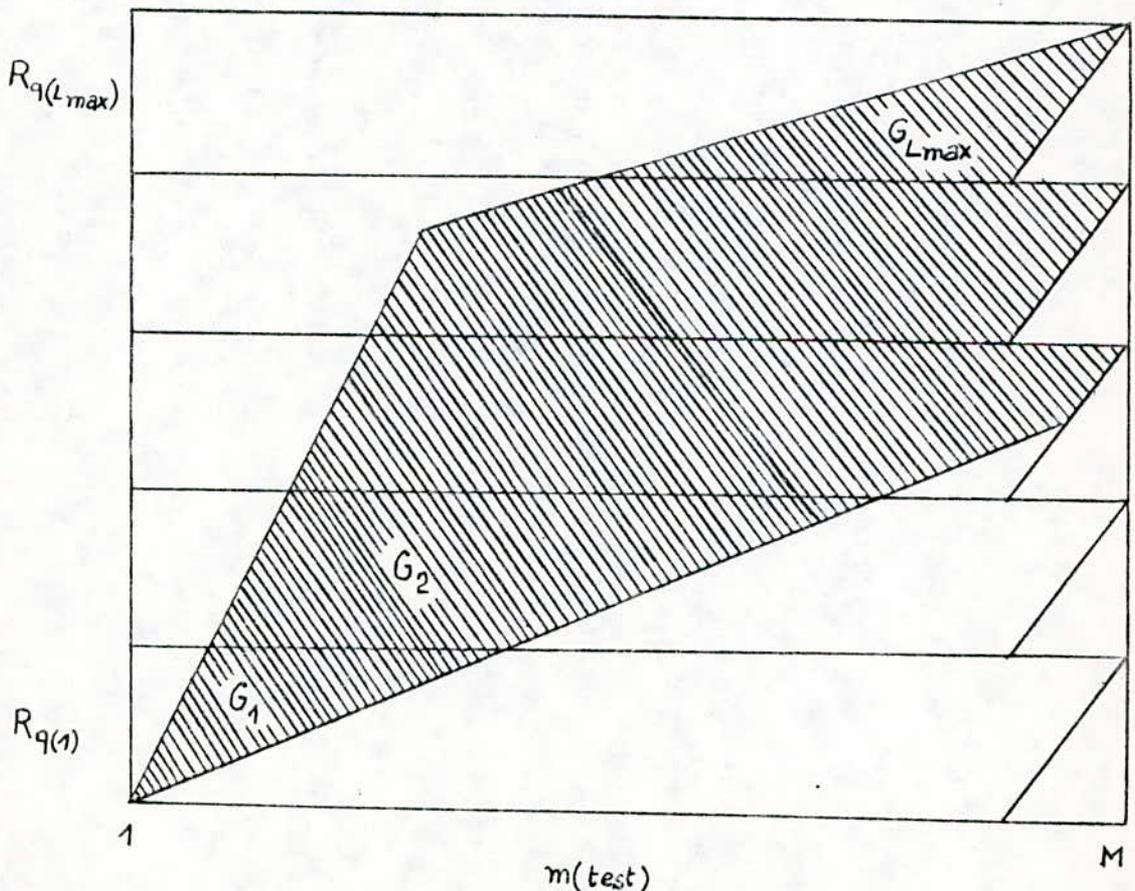


Figure 5.4 Espace de recherche réduit.

5.6.2 Technique de réduction d'intervalle ;

La figure (5.5) montre comment on peut réduire l'intervalle à chaque niveau. La partie (a) montre la région originale (bornes discontinues) et la région réduite (bornes pleines). La partie (b) illustre la méthode décrite ci-dessous pour la réduction de l'intervalle.

Au début du 1^{ème} niveau, la distance moyenne minimale à la fin du (l-1)^{ème} niveau est donnée par ϕ_{l-1} :

$$\phi_{l-1} = \min_{1 \leq m \leq M} \left(\frac{\bar{D}_{l-1}^B(m)}{m} \right) \quad (5.12)$$

Deux valeurs S_l^1 et S_l^2 peuvent alors être trouvées comme suit :

S_l^1 = le plus grand m tel que :

$$\left(\frac{\bar{D}_{l-1}^B(m)}{m} \right) > \phi_{l-1} \cdot M_T \quad \text{pour tout } m < S_l^1$$

S_l^2 = le plus petit m tel que :

$$\left(\frac{\bar{D}_{l-1}^B(m)}{m} \right) > \phi_{l-1} \cdot M_T \quad \text{pour tout } m > S_l^2$$

où $M_T \approx 1,2$ est un paramètre controlant la grandeur du l'intervalle de début.

L'exemple donné par la figure (5.5.b) explique pourquoi on doit chercher S_l^1 et S_l^2 à partir des extrémités de l'intervalle à réduire puisque la courbe excède le seuil ($M_T \cdot \phi_{l-1}$) entre S_l^1 et S_l^2 .

On doit donc changer l'étape 4°) de manière à retrécir l'intervalle. Les fonctions de bornes modifiées supérieure et inférieure $U_l(m)$ et $L_l(m)$ est données par :

$$L_l(m) = \max \left(1, \frac{m - S_l^2}{2} \right), 2(m - M) + N_V \quad (5.13.a)$$

$$U_l(m) = \min \left(N_V, 2(m - S_l^1), \frac{m - M}{2} (L_{max} - 1) N_{max} + N_V \right) \quad (5.13.b)$$

N_{max} représente la longueur de la forme de référence la plus longue. Le dernier terme dans (5.13.b) reflète une ligne de pente 1/2 qui vient du point $m = M$ (comme l'indique la figure 5.4).

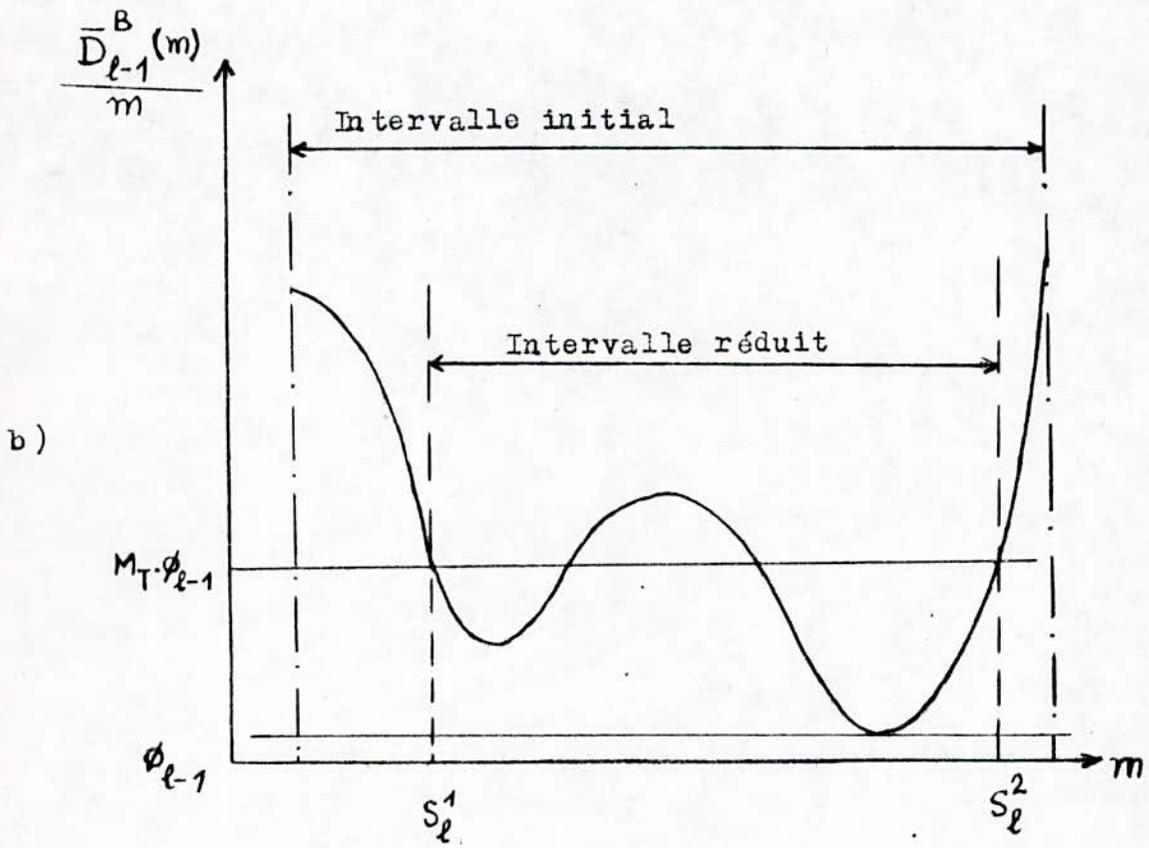
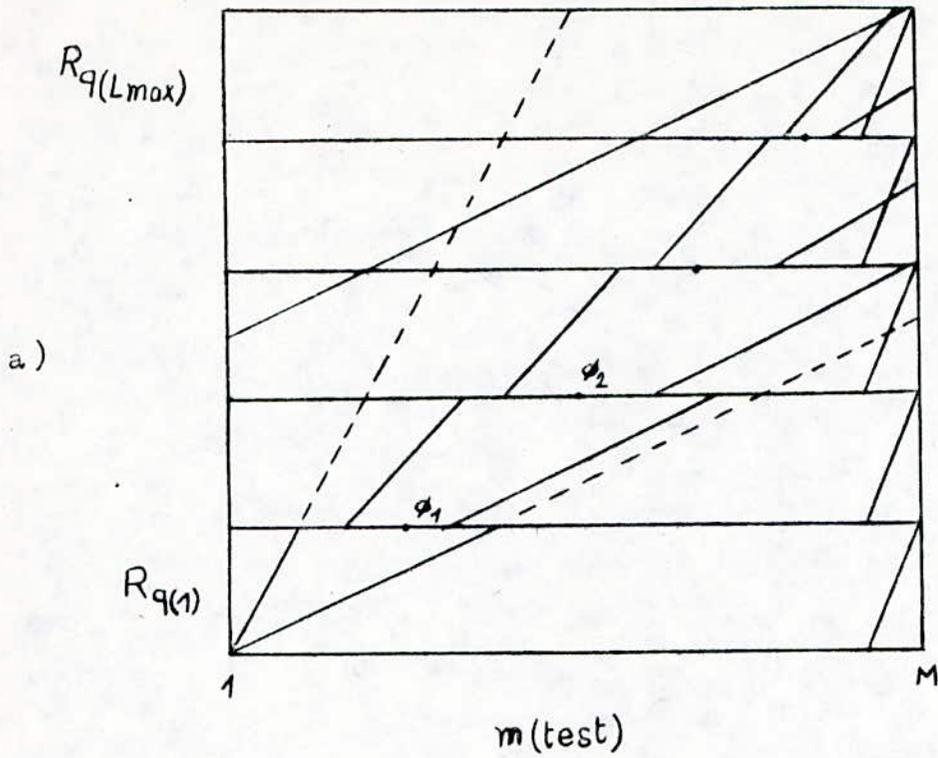


Figure (5.5) Illustration de la technique de réduction d'intervalle.

5.6.4 Introduction des flous :

RABINER et SCHMIDT ont proposés certaines modifications, qui consiste à introduire deux plages de tolérance δ_{R_1} et δ_{R_2} constantes au début et à la fin de la trame N_v de la référence (fig. 5.6) dans le but d'accélérer de plus en plus l'algorithme de calcul.

On réalise ces propositions en modifiant les étapes 5°) et 6°) comme suit :

5°) Pour $m=1, 2, \dots, M$ et $n=L_\ell(m), \dots, U_\ell(m)$ calculer en utilisant $R=R_v$

$$\hat{n} = \begin{cases} n' = 0 \text{ ou } \underset{\substack{L(n) \leq n' \leq U(n)}}{\operatorname{argmin}} (D_\ell(m-1, n')) & \text{pour } 1 \leq n \leq 1 + \delta_{R_1} \\ \underset{\substack{L(n) \leq n' \leq U(n)}}{\operatorname{argmin}} (D_\ell(m-1, n')) & \text{pour } n > 1 + \delta_{R_2} \end{cases}$$

$$D_\ell(m, n) = d_\ell(m, n) + D_\ell(m-1, \hat{n})$$

$$F_\ell(m, n) = F_\ell(m-1, \hat{n})$$

6°) Pour $m=1, 2, \dots, M$ calculer

$$\bar{n} = \underset{N_v - \delta_{R_2} \leq n \leq N_v}{\operatorname{argmin}} (D_\ell^v(m, n))$$

$$\bar{D}_\ell^v(m) = D_\ell(m, \bar{n})$$

$$\bar{F}_\ell^v(m) = F_\ell(m, \bar{n})$$

où \bar{n} est utilisé pour indiquer le meilleur point final référentiel.

5.7 L'algorithme LB-DTW modifié ;

1°) Initialiser $\bar{D}_0^B(m) = 0$ pour $m=0$
 $\bar{D}_0^B(m) = \infty$ pour $m=1, 2, \dots, M$
 $\bar{D}_\ell^B(0) = \infty$ pour $l=1, 2, \dots, L_{\max}$

2°) Pour $l=1, 2, \dots, L_{\max}$ faire les étapes de 3°) à 7°)

3°) Pour $v=1, 2, \dots, V$ faire les étapes de 4°) à 6°)

4°) Pour $m=0, 1, 2, \dots, M$ initialiser

$$D_\ell(m, 0) = \begin{cases} \bar{D}_{\ell-1}^B(m) & \text{si } S_\ell^1 \leq m \leq S_\ell^2 \\ \infty & \text{ailleurs} \end{cases}$$

$$F_\ell(m, 0) = m \quad \text{si } S_\ell^1 \leq m \leq S_\ell^2$$

où $S_1^1 = S_1^2 = 1$

5°) Pour $m = S_\ell^1, S_\ell^1 + 1, \dots, M$ et pour $n = L_\ell(m), \dots, U_\ell(m)$
calculer en utilisant $R = R_V$

$$\hat{n} = \begin{cases} n' = 0 ; \quad \underset{\bar{L}(n) \leq n' \leq \bar{U}(n)}{\operatorname{argmin}} (D_\ell(m-1, n')) & \text{pour } 1 \leq n \leq 1 + \delta_{R_1} \\ \underset{\bar{L}(n) \leq n' \leq \bar{U}(n)}{\operatorname{argmin}} (D_\ell(m-1, n')) & \text{pour } n > 1 + \delta_{R_2} \end{cases}$$

$$D_\ell(m, n) = d_\ell(m, n) + D_\ell(m-1, \hat{n})$$

$$F_\ell(m, n) = F_\ell(m-1, \hat{n})$$

6°) Pour $m = S_\ell^1, S_\ell^1 + 1, \dots, M$ calculer :

$$\bar{n} = \underset{N_V - \delta_{R_2} \leq n \leq N_V}{\operatorname{argmin}} (D_\ell^V(m, n))$$

$$\bar{D}_\ell^V(m) = D_\ell^V(m, \bar{n})$$

$$\bar{F}_\ell^V(m) = F_\ell^V(m, \bar{n})$$

7°) Pour $m = S_\ell^1, S_\ell^1 + 1, \dots, M$ calculer :

$$\bar{D}_\ell^B(m) = \min_{1 \leq v \leq V} (\bar{D}_\ell^V(m))$$

$$W_\ell(m) = \underset{1 \leq v \leq V}{\operatorname{argmin}} (\bar{D}_\ell^V(m))$$

$$\bar{F}_\ell^B(m) = \bar{F}_\ell^{W_\ell(m)}(m)$$

8°) $(L, E) = \underset{L_{\min} \leq \ell \leq L_{\max}}{\operatorname{argmin}} \underset{M - \delta_{\text{end}} \leq e \leq M}{\operatorname{argmin}} (D(e) / e)$

9°) $e_L = E$ (E est utilisé pour enregistrer le point de la fin)

10°) $e_\ell = \bar{F}_{\ell+1}^B(e_{\ell+1})$, $\ell = L-1, L-2, \dots, 1$

11°) $\bar{R}^S = R_{W_1(e_1)} \oplus R_{W_2(e_2)} \oplus \dots \oplus R_{W_L(e_L)}$

5.8 Conclusion :

Ces deux algorithmes avec l'algorithme SAKOE (two-Level) ont été comparés /18/ en prenant des valeurs numériques. Ce qui nous a permis d'affirmer que le Two-Level de SAKOE exige 3 fois plus de mémoire et 15 fois plus d'opérations que l'algorithme LB-DTW.

D'autre part le calcul a été réduit de moitié en modifiant l'algorithme LB-DTW. Il est donc conseillé d'utiliser l'algorithme LB-DTW modifié pour toute reconnaissance de mots connectés.

PROGRAMMATION

DEUXIEME PARTIE :

PROGRAMMATION

En se basant sur l'algorithme LB-DTW modifié, on a élaboré un programme permettant la reconnaissance des mots connectés. Ce programme fait apparaître une étude syntaxique pour une meilleure compréhension de la parole continue.

Le programme est écrit en FORTRAN qui est le langage le plus adapté à notre problème. De plus c'est un langage scientifique très puissant et universel.

I Commentaires :

Pour permettre aux lecteurs, de comprendre aisément cette deuxième partie, nous donnerons ci-après un ensemble des notations utilisées qu'on fait suivre d'un bref commentaire :

- CLB : tableau des distances temporaires, qui recopie les distances ou les remplace par ∞ quand il n'y a pas de successeur.
- CORES : numéro du mot dans la classe syntaxique.
- D(N) : Distance optimale cumulée entre le segment (1, ..., INDTES-1) du test et la concaténation des mots trouvés aux niveaux précédents et du segment (1, ..., N) de la référence.
- IDEL : flou final décalé.
- IFOPT : frontière du mot précédent au point courant (I, INDTES).
- IFF(N) : frontière du mot précédent pour le chemin optimal arrivant au point (N, INDTES - 1).
- IG(N) : indicateur mémoire de la dernière contrainte locale employée pour aboutir au point (N, INDTES - 1)

- IFLOU : tableau des flous
- ISLP1 : borne inférieure du vecteur résultat au niveau précédent (voir schéma d'illustration ci-dessous).
- INDPAS : déplacement dans le test.
- INDREF : index de position dans la référence.
- NETATM : nombre maximum d'état.
- NMAXRC : longueur maximale du test.
- NUMOT : numéro du mot comparé.
- DLB : tableau des distances cumulées.
- DOPT : distance optimale cumulée au point courant (I , INDTES).

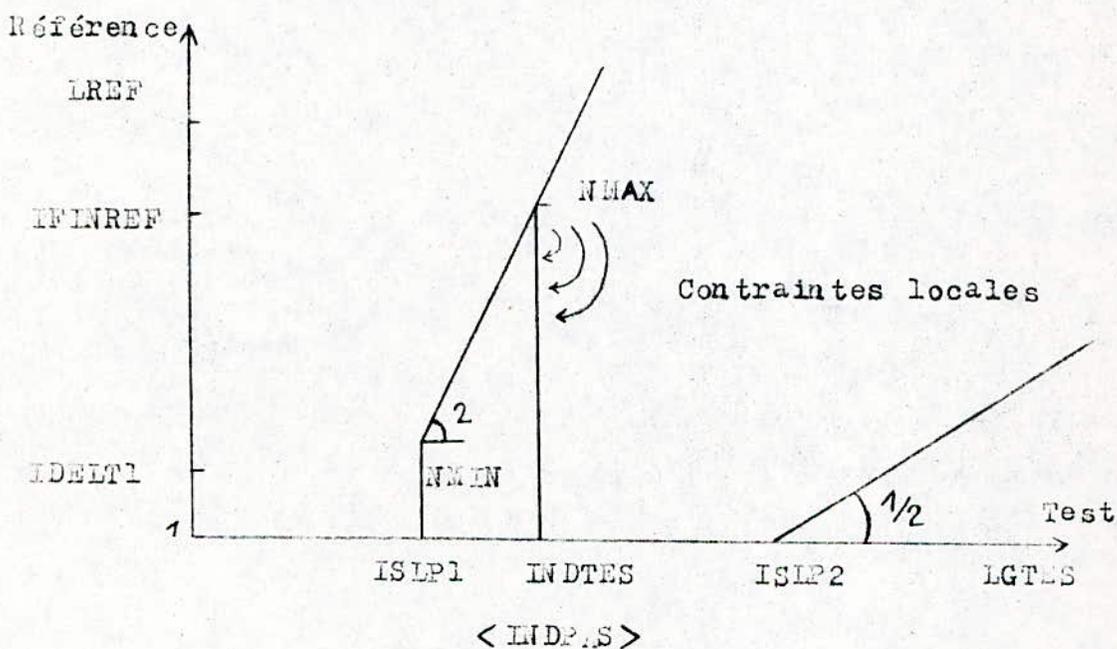
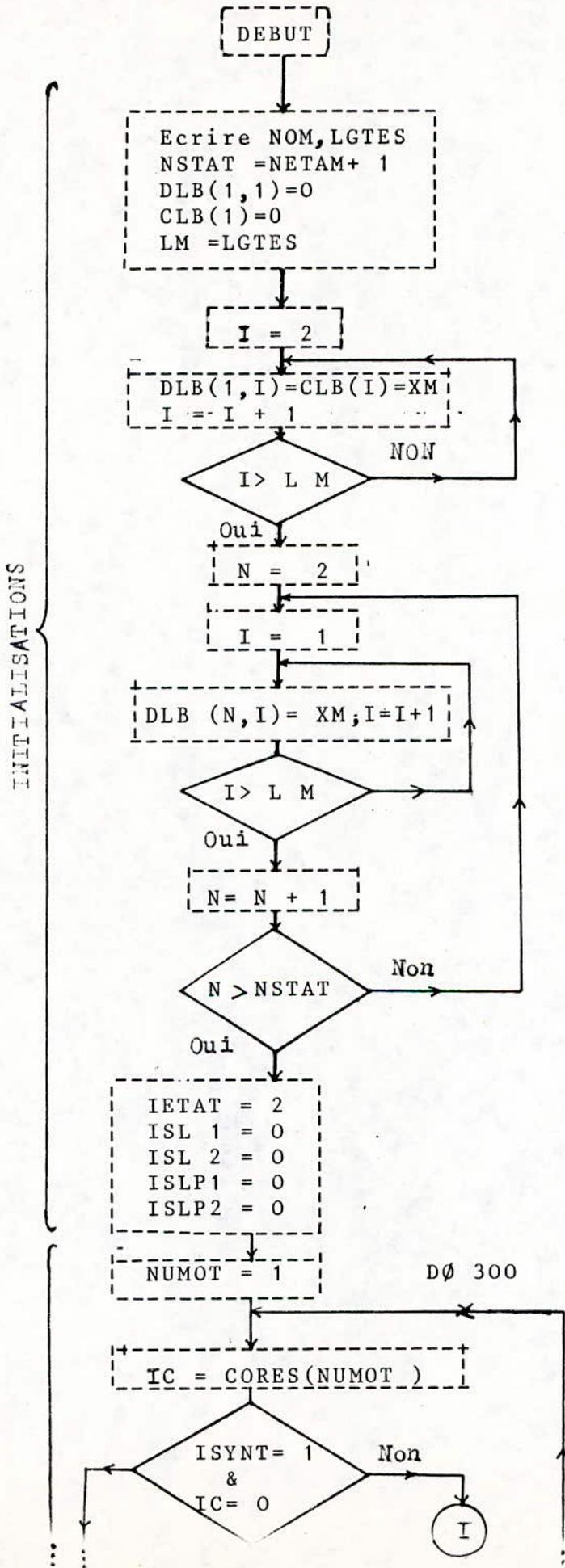


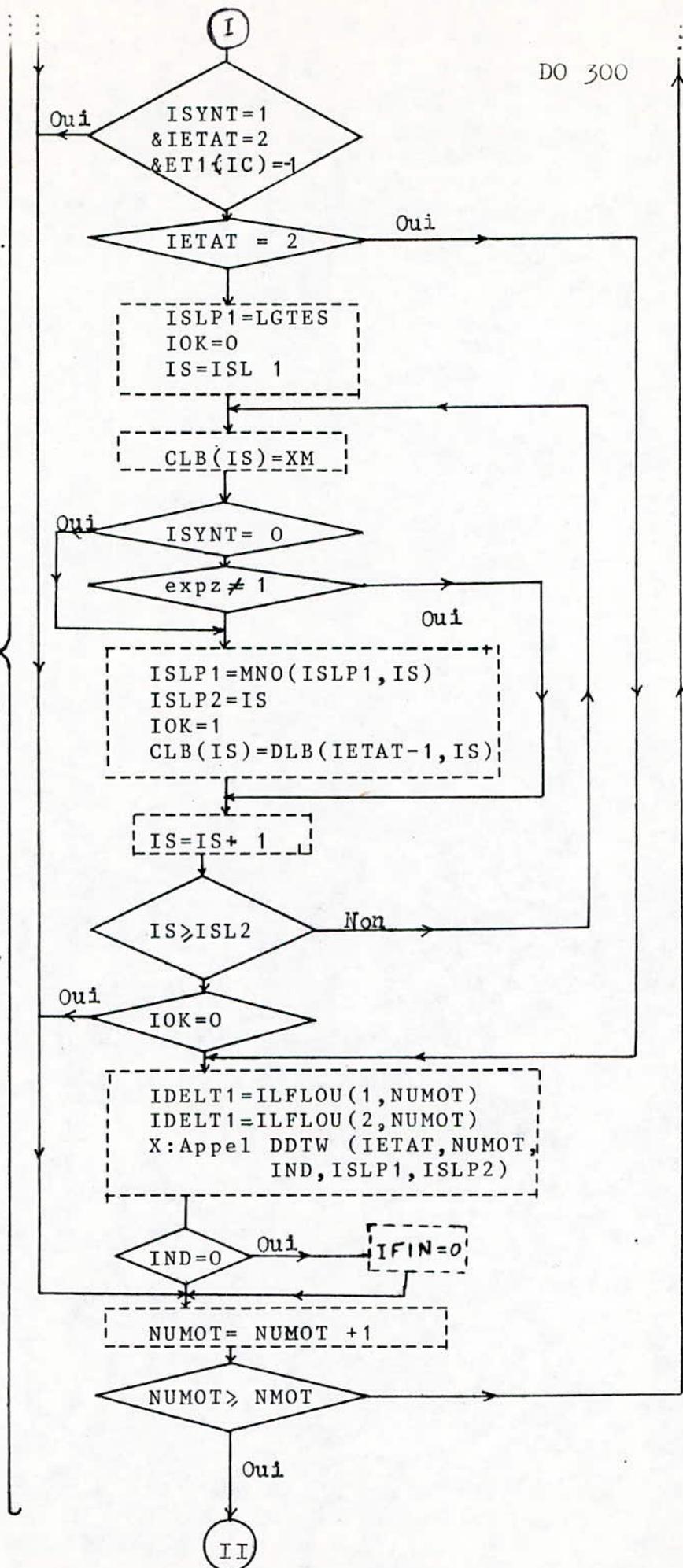
Schéma d'illustration de quelques paramètres.

En fin notons que le segment (ISLP1 , ISLP2) du vecteur résultant au niveau précédent sert de conditions initiales pour le niveau courant.

A) . Organigramme de reconnaissance type " LB -DT W " :

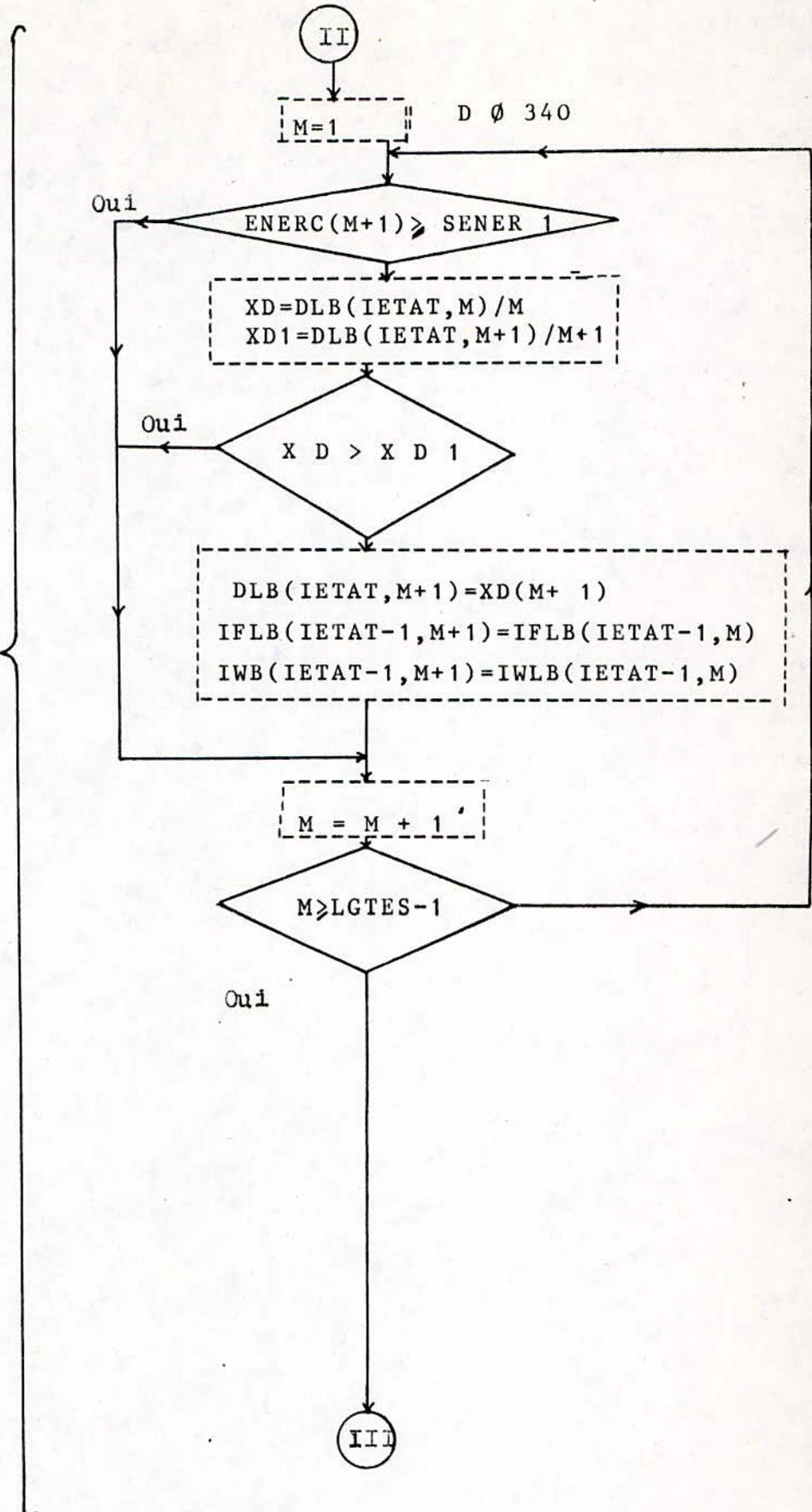


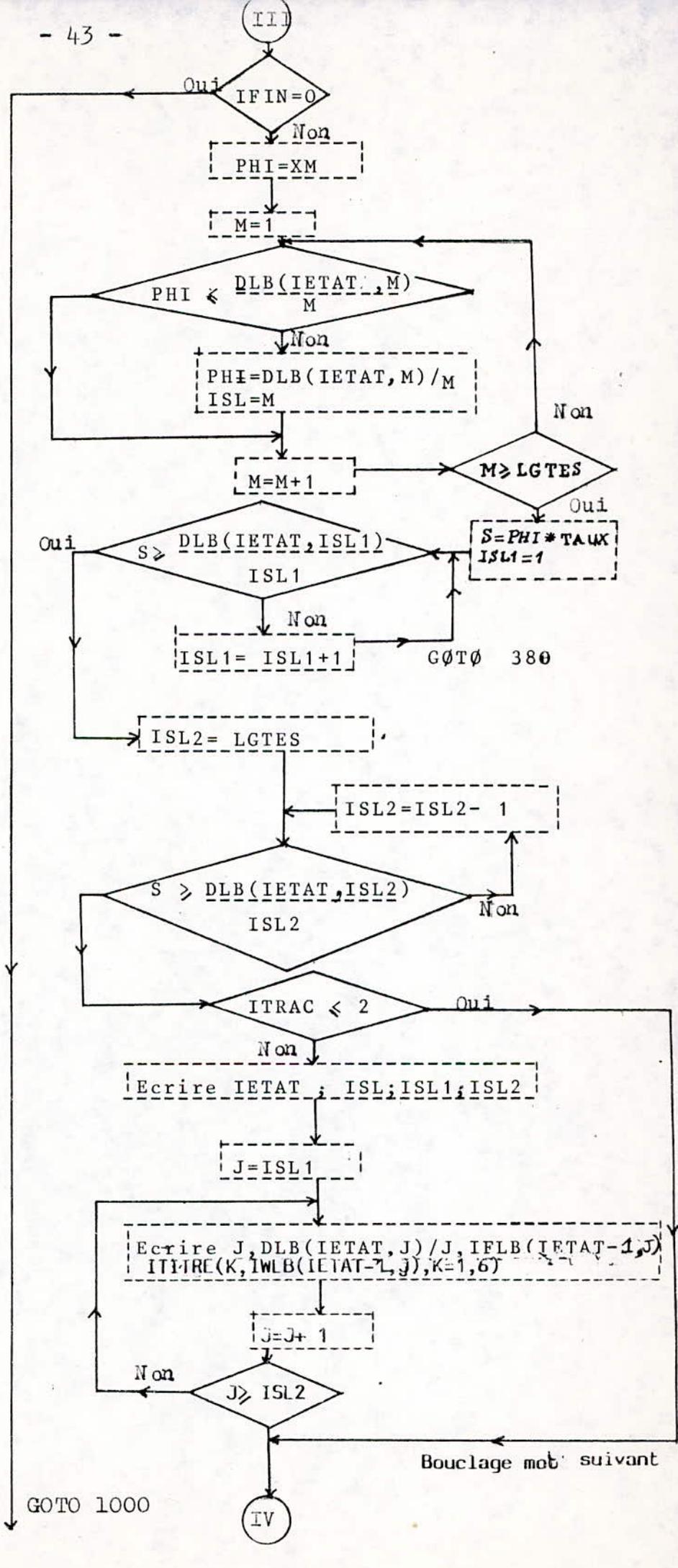
Boucle sur
les mots du
dictionnaire



expz = (IETAT-2 , IC+1 , CORES(IWLB(IETAT-2, IS)))

Suppression des
Silences
intermédiaires



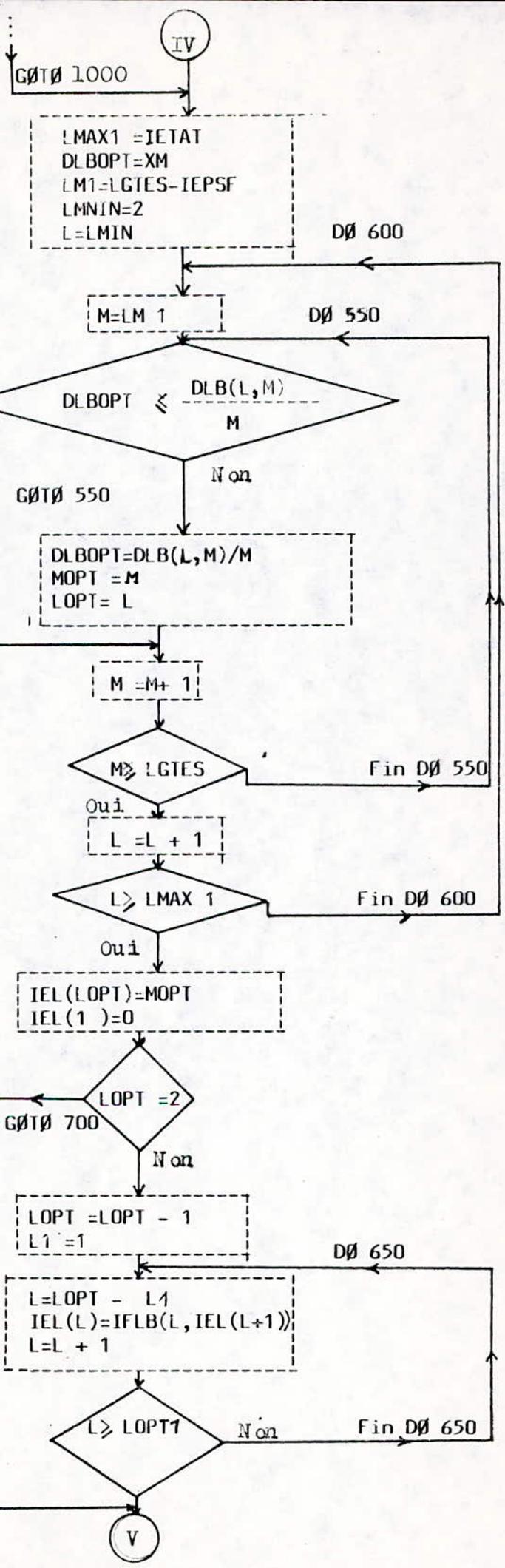


Heuristique

GOTO 1000

IV

Bouclage mot suivant

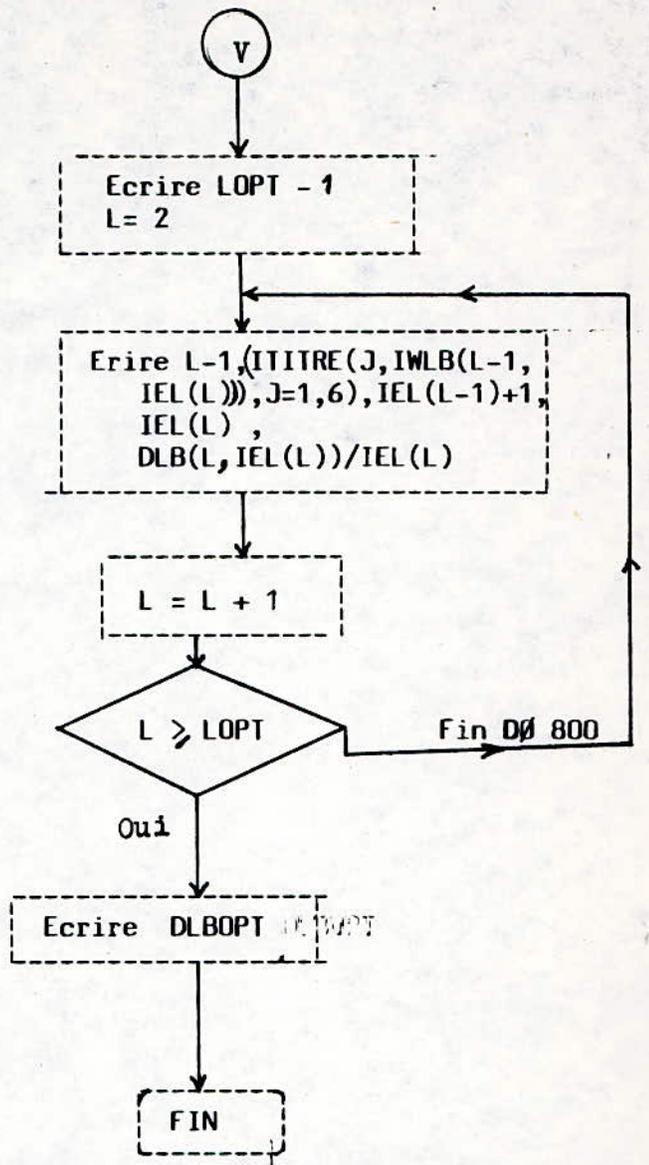


Fin de phrase

C
H
A
I
N
A
G
E

A
R
R
I
E
R
E

Impression
des
Résultats



GOTO 300

(A)

INDREF = NTOT - N, N1 = INDREF - 1, N2 = INDREF - 2
 X = DIST1 (TABRF (1, N2, NUMOT), TABRC (INDTES))
 DOPT = D(N1), IFOPT = IFF(N2), IGOPT = 1

$D(N2) \geq DOPT$ (Oui)

DOPT = D(N2); IFOPT = IFF(N2); IGOPT = 1

$IG(INDREF) \neq 1$ (Oui)

$D(INDREF) \geq DOPT$ (Oui)

DOPT = D(INDREF); IFOPT = IFF(INDREF); IGOPT = 0

$N2 > IDEL1$ (Oui)

$M1 > ISLP2$ (Oui)

$CLB(M1) > DOPT$ (Oui)

DOPT = CLB(M1); IFOPT = M1; IGOPT = 1 (Non)

$D(INDREF) = DOPT + X$; $IFF(INDREF) = IFOPT$; $G(INDREF) = IGOPT$

$N2 < IFINRF$ (Oui)

$D(INDREF) \geq DLB(IETAT, INDTES)$ (Oui)

$DLB(IETAT, INDTES) = D(INDREF)$; $IFLB(IETAT - 1, INDTES) = IFOPT$
 $IWLB(IETAT - 1, INDTES) = NUMOT$

$N = N + 1$

$N \geq NMAX$ (Non)

$INDPAS = IFINT1$ (Oui)

$INDPAS = IFINT2$ (Oui)

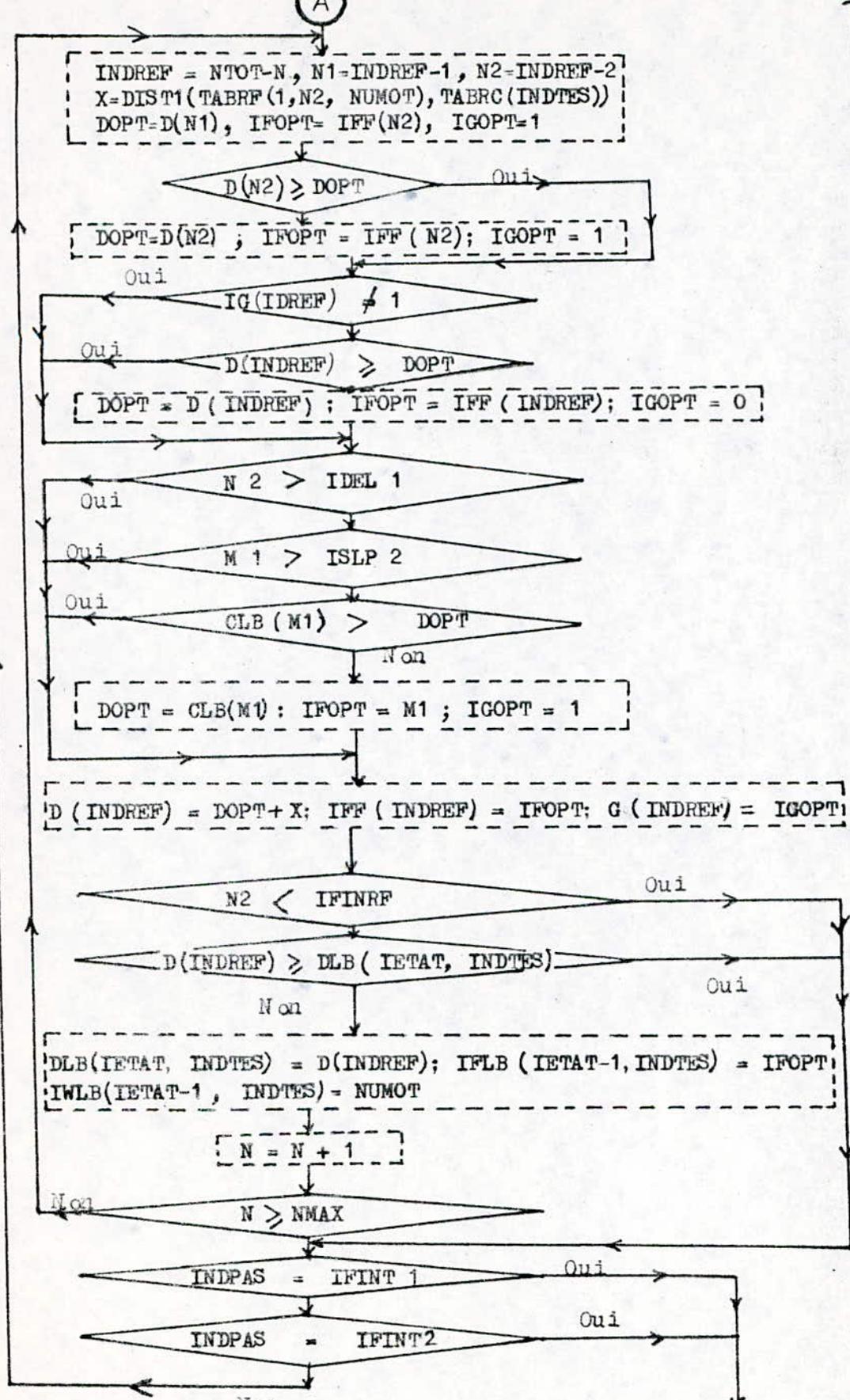
(Non)

RETURN

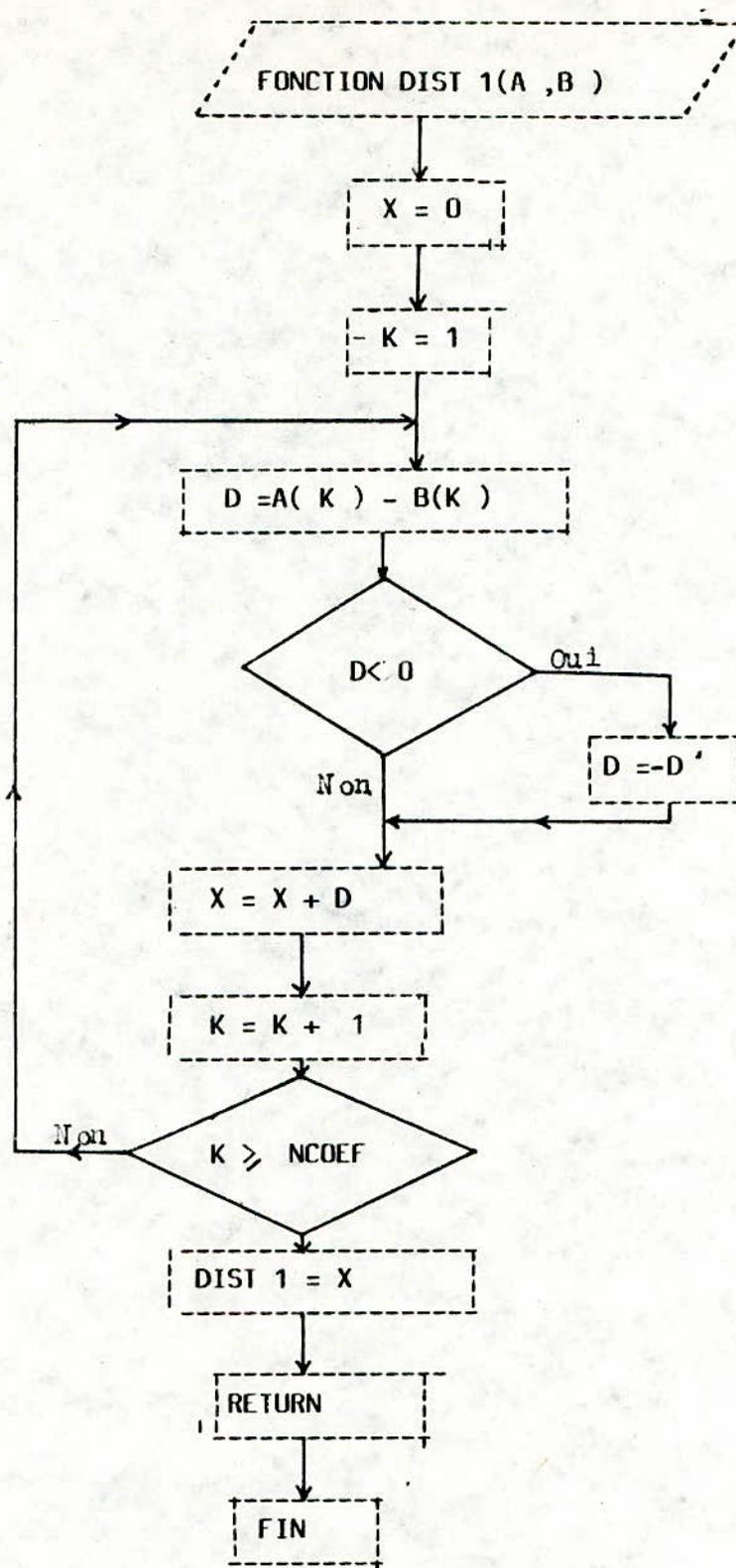
FIN

Points
suivants

Test
fin
boucle



C) . Fonction calculant la distance entre deux échantillons de parole A et B



III. Programme :

Le programme que nous livrons ici est un programme écrit en FORTRAN sur le Vax 11.750 du centre de calcul de l'E.N.P. . Il traduit l'algorithme LB-DTW complété par une étude syntaxique pour une meilleure compréhension de la parole continue. Ce programme fait appel à un sous programme de détection de mot et une fonction pour le calcul de la distance entre deux échantillons.

Toutes les étapes d'exécution de ce programme ont été effectuées notamment l'étape de correction et l'étape de lien. Ce programme doit être précédé par un programme d'analyse fournissant l'ensemble des paramètres tels que les coefficients cepstraux et autre qui servent de données à notre programme pour qu'il soit exécutable.

En fin nous souhaitons que les étudiants, du système annuel ou de préférence ceux du post-graduation, traiteront l'ensemble du problème tout en espérant que notre travail leur sera d'une aide très précieuse.

CONCLUSION GENERALE

L'ampleur du problème et la nécessité d'apporter des solutions spécifiques font de la reconnaissance de la parole un champ d'application important et souvent original de l'intelligence artificielle.

Dans cette thèse nous avons tenté de décrire les étapes essentielles de la reconnaissance de la parole continue à la lumière des grandes techniques récemment découvertes.

La reconnaissance que nous avons proposé s'appuie sur l'algorithme de programmation dynamique. Celle-ci permet d'atténuer les déformations temporelles. Pour ce qui est du choix de la distance d'équivalente à la distance cepstrale d_0 , nous n'avons fait que des approximations. Toutefois elle s'avère nettement plus avantageuse pour sa simplicité et les résultats obtenus et comparés à d'autres distances.

L'alignement DTW d'une chaîne de mots connectés se fait à l'aide de l'algorithme LB-DTW qui donne en sortie la super forme de référence adaptant au mieux la chaîne d'entrée. En ce qui concerne l'heuristique, son avantage est de réduire l'espace de recherche, mais il faut être prudent et choisir un compromis entre le gain en calcul et en mémoire avec la diminution du taux de reconnaissance.

Actuellement, la plus part des programmes de reconnaissance fonctionnent en temps différé et sont écrits en FORTRAN. Nous avons donné ici un programme qui retrace l'algorithme LB-DTW modifié qui devrait suivre un autre programme d'analyse pour qu'il puisse fournir les résultats cherchés.

Le sujet de ce projet de fin d'études nous a permis de consolider notre formation par la découverte d'un nouveau foyer de la recherche en traitement de signal. En effet on a vu l'application de

la FFT ,fenêtrage et codage au sigal parole,

En fin nous avons aussi approché de plus près l'algorithmique et la programmation.

Faut-il rappeler que les recherches sur la parole, qui lient indissolublement sciences humaines, sciences exactes et technologie, ont aussi une dimension culturelle? Est-il admissible que le dialogue entre l'homme et l'ordinateur doive se faire par l'intermédiaire d'une langue étrangère? Où est l'arabe, langue du Coran, des musulmans et de l'état Algérien dans ce domaine bien' préci de la recherche. Qu'il nous soit permis d'exprimer ici toute notre reconnaissance à nos professeurs B. BOUSSEKSOU, M. GUERTI en particulier et à messieurs A. HADJ SALAH et M. ANANI qui se sont assignés pour mission la revalorisation de la langue nationale par leurs contributions dans ce domaine si touchant qui est la reconnaissance de la parole.

Comment peut on faire une reconnaissance en arabe? Nous distinguerons principalement deux axes de recherche. A court terme, il est essentiel de mieux comprendre la structure fine de la langue arabe. Cela suppose des connaissances précises sur la façon dont l'oreille humaine analyse le signal et une meilleure compréhension de la relation entre les symboles sonores (phonèmes, syllabes) et les sons véritables. Il faut améliorer l'exploitation de cette information et l'incorporer aux systèmes de reconnaissance. A long terme, plusieurs domaines de recherche peuvent ouvrir la voie à des progrès substantiels. Dans le code de la parole coexistent plusieurs structures : la phonologie, la syntaxe et la sémantique, par exemple. Il faudra élaborer une théorie générale de ces codes complexes pour pouvoir comprendre et commander les interactions entre les divers niveaux. Il serait également souhaitable de mieux comprendre le mécanisme de l'apprentissage de la langue arabe chez l'homme. Il est difficile de prévoir avec quel succès ces recherches arriveront à cerner la communication orale naturelle. Ces objectifs, actuellement, paraient ambitieux, néanmoins, nous devrions les réaliser même si c'est à plus long terme.

PROGRAMME

```
C          *****
C          *          LEVEL BUILDING DTW          *
C          *****
C          DIMENSION NOM(6),TABRC(8,150),ENERC(150)
C          DIMENSION TABRF(8,90,38),ENERF(90,38)
C          DIMENSION ITITRE(6,38),LRF(38)
C
C          INTEGER TCLAS(6,20),CORES(38)
C          INTEGER ET1(20),ET2(7,21,20)
C
C          DIMENSION ILFLOU(2,38)
C          DIMENSION CLB(150),DLB(7,150),IFLB(6,150),IWLB(6,150)
C          DIMENSION IEL(6)
C
C          COMMON /SER/ IEPSF,TAUX,SENER1,SENER2,NETATM,ISYNT,IFLOU,ICAD
C          COMMON /RECO/ ITRAC,NCOEF,NMAXRF,NMAXRC
C          COMMON /REF/ NMOT,ITITRE,LRF,TABRF,ENERF
C          COMMON /TEST/ NBTES,NOM,LGTES,TABRC,ENERC
C          COMMON /SYNT/ NBCLAS,NETAT,TCLAS,CORES,ET1,ET2
C          COMMON /FLOU/ ILFLOU,IDELT1,IDELT2
C          COMMON /ALGO/ ISL,ISL1,ISL2,CLB,DLB,IFLB,IWLB,IEL
C
C          DATA XM/100000./
C          DATA XM1/100./
C
C          WRITE(108,1005) NOM,LGTES
C          NSTAT=NETATM+1
C          ~~~~~
C          ~          INITIALISATIONS          ~
C          ~~~~~
C          DLB(1,1)=0.
C          CLB(1)=0.
C          LM=LGTES
C          DO 10 I=2,LM
C          DLB(1,I)=XM
C          CLB(I)=XM
10          CONTINUE
C          DO 20 N=2,NSTAT
C          DO 20 I=1,LM
C          DLB(N,I)=XM
20          CONTINUE
C          IETAT=2
C          ISL1=0
C          ISL2=0
C          ISLP1=0
C          ISLP2=0
C          ~~~~~
C          ~          BOUCLE SUR LES NIVEAUX          ~
C          ~~~~~
200          CONTINUE
C          IFIN=0
C
C          DO 300 NUMOT=1,NMOT
C          IC=CORES(NUMOT)
C          IF(ISYNT.EQ.1.AND.IC.EQ.0) GOTO 300
C          IF(ISYNT.EQ.1.AND.IETAT.EQ.2.AND.ET1(IC).EQ.-1) GOTO 300
C
C          IF(IETAT.EQ.2) GOTO 280
C          ISLP1=LGTES
C          IOK=0
C          DO 250 IS=ISL1,ISL2
C          CLB(IS)=XM
C          IF(ISYNT.EQ.0) GOTO 240
C          IF(ET2(IETAT-2,IC+1,CORES(IWLB(IETAT-2,IS))).NE.1)GOTO250
```

```
240 CONTINUE
    ISLP1=MIN0(ISLP1,IS)
    ISLP2=IS
    IOK=1
    CLB(IS)=DLB(IETAT-1,IS)
250 CONTINUE
    IF(IOK.EQ.0)GOTO 300
280 CONTINUE
    IDELT1=ILFLOU(1,NUMOT)
    IDELT2=ILFLOU(2,NUMOT)
    CALL DDTW(IETAT,NUMOT,IND,ISLP1,ISLP2)
    IF(IND.EQ.0) IFIN=1
300 CONTINUE
C     ~~~~~
C     ~ SUPPRESSION DES SILENCES INTERMEDIAIRES ~
C     ~~~~~

    DO-340 M=1,LGTES-1
    IF(ENERC(M+1).GT.SENERC)GOTO340
    XD=DLB(IETAT,M)/M
    XD1=DLB(IETAT,M+1)/(M+1)
    IF(XD.GT.XD1)GOTO340
    DLB(IETAT,M+1)=XD*(M+1)
    IFLB(IETAT-1,M+1)=IFLB(IETAT-1,M)
    IWLB(IETAT-1,M+1)=IWLB(IETAT-1,M)
340 CONTINUE
C     ~~~~~
C     ~ HEURISTIQUE ~
C     ~~~~~

    IF(IFIN.EQ.0)GOTO 1000
    PHI=XM
    DO 350 M=1,LGTES
    IF(DLB(IETAT,M)/M.GE.PHI)GOTO 350
    PHI=DLB(IETAT,M)/M
    ISL=M
350 CONTINUE
    S=PHI*TAUX
C
    ISL1=1
380 CONTINUE
    IF(DLB(IETAT,ISL1)/ISL1.LE.S)GOTO 390
    ISL1=ISL1+1
    GOTO 380
390 CONTINUE
    ISL2=LGTES
400 CONTINUE
    IF(DLB(IETAT,ISL2)/ISL2.LE.S)GOTO 410
    ISL2=ISL2-1
    GOTO 400
410 CONTINUE
C     ~~~~~
C     ~ TRACE EVENTUELLE POUR CHAQUE NIVEAUX ~
C     ~~~~~

    IF(ITRAC.LE.2)GOTO500
    WRITE(108,1001) IETAT
    WRITE(108,1002) ISL,ISL1,ISL2
    DO 499 J=ISL1,ISL2
    WRITE (108,1003) J,DLB(IETAT,J)/J,IFLB(IETAT-1,J),(ITITRE(
1 K,IWLB(IETAT-1,J)),K=1,6)
499 CONTINUE
500 CONTINUE
```

```
C
C
C      ~~~~~
C      ~   BOUCLAGE MOTS SUIVANT   ~
C      ~~~~~
```

```
IF(IETAT.EQ.NSTAT)GOTO1000
IETAT=IETAT+1
GOTO 200
```

```
C
C
C      ~~~~~
C      ~   FIN DE PHRASE   ~
C      ~~~~~
```

```
1000  CONTINUE
      LMAX1=IETAT
      DLBOPT=XM
      LM1=LGTES-IEPSF
      LMIN=2
      DO 600 L=LMIN,LMAX1
      DO 550 M=LM1, LGTES
      IF(DLB(L,M)/M.GE.DLBOPT)GOTO 550
      DLBOPT=DLB(L,M)/M
      MOPT=M
      LOPT=L
550   CONTINUE
600   CONTINUE
```

```
C
      IEL(LOPT)=MOPT
      IEL(1)=0
      IF(LOPT.EQ.2)GOTO 700
      LOPT1=LOPT-1
      DO 650 L1=1,LOPT1
      L=LOPT-L1
      IEL(L)=IFLB(L, IEL(L+1))
650   CONTINUE
700   CONTINUE
```

```
C
C
C      ~~~~~
C      ~   IMPRESSION DES RESULTATS   ~
C      ~~~~~
```

```
      WRITE(108,1006) LOPT-1
      DO 800 L=2,LOPT
      WRITE(108,1007) L-1,(ITITRE(J,IWLB(L-1,IEL(L))),J
1     =1,6),IEL(L-1)+1,IEL(L),DLB(L,IEL(L))/IEL(L)
800   CONTINUE
      WRITE(108,1008) DLBOPT
```

```
C
1001  FORMAT('ETAT: ',I4)
1002  FORMAT('MIN:',I3,' HEURIST.',I5)
1003  FORMAT(2X,I4,5X,F7.2,2X,I5,2X,6A2)
1005  FORMAT('/', 'PHRASE: ',6A2, 'DE LONGUEUR: ',I4)
C
1006  FORMAT('SEGMENTATION: ',I4,'MOT(S) ')
1007  FORMAT('MOT',I3,2X,6A2,'D: ',I4,'F: ',I4,'DIST:',F7.2)
1008  FORMAT('DISTANCE MOYENNE',F8.2)
      STOP
      END
```

```
*****  
*          SOUS PROGRAMME DE DETECTION DE MOTS DANS      *  
*          LA PAROLE CONTINUE                             *  
*****
```

SUBROUTINE DDTW (IETAT,NUMOT,IND,ISLP1,ISLP2)

DIMENSION NOM(6),TABRC(8,150),ENERC(150)
DIMENSION TABRF(8,90,38),ENERF(90,38)
DIMENSION ITITRE(6,38),LRF(38)

INTEGER TCLAS(6,20),CORES(38)
INTEGER ET1(20),ET2(7,21,20)

DIMENSION IFLOU(2,38)

DIMENSION CLB(150),DLB(7,150),IFLB(6,150),IWL(6,150)
DIMENSION IEL(6)

COMMON /SER/ IEPST,TAUX,SENER1,SENER2,NETATM,ISYNT,IFLOU,ICAD
COMMON /RECO/ ITRAC,NCOEF,NMAXRF,NMAXRC
COMMON /REF/ NMOT,ITITRE,LRF,TABRF,ENERF
COMMON /TEST/ NBTES,NOM,LGTES,TABRC,ENERC
COMMON /SYNT/ NBCLAS,NETAT,TCLAS,CORES,ET1,ET2
COMMON /FLOU/ IFLOU,IDEFT1,IDEFT2
COMMON /ALGO/ ISL,ISL1,ISL2,CLB,DLB,IFLB,IWL,IEL

DIMENSION D(75),IG(75),IFF(75)
DATA XM/100000./

LREF=LRF(NUMOT)
NPT=ISLP2-ISLP1+1
IND=0
LM1=LGTES-ISLP1
LM2=(LREF-IDEFT1-IDEFT2-1)/2+1
IF(LM1.GE.LM2)GOTO 500
IND=-1
RETURN

CONTINUE

```
*****  
*          PREMIER POINT DU NIVEAU                       *  
*****
```

INDPAS=1
INDTES=INDPAS+ISLP1
NMIN=3
NMAX=3+IDEFT1
D(1)=XM
D(2)=XM
D(NMAX+1)=XM
D(NMAX+2)=XM
NTOT=NMIN+NMAX
DO 20 N=NMIN,NMAX
INDREF=NTOT-N
DOPT=DIST1(TABRF(1,INDREF-2,NUMOT),TABRC(1,INDTES))
D(INDREF)=DOPT
IF(IETAT.NE.2) D(INDREF)=D(INDREF)+CLB(INDTES-1)
IG(INDREF)=1
IFF(INDREF)=ISLP1
CONTINUE

B I B L I O G R A P H I E

- / 1 /. AUGUSTINE B. , GRAY J-R. , MARKEL J-D. , Distance mesures for speeck processing. IEEE Trans ASSP N° 5 Oct.76
- / 2 /. AUVRAY J. , Electronique des signaux échantillonnées et numériques . DUNOD,79
- / 3 /. BASTIN C. , BENZEKRI J-P. , BORGARIT C. et CAZES P. , Pratique de l'analyse des données, DUNOD,80
- / 4 /. BELLMAN R-E. , DREYFUS S-E. , La programmation dynamique et ses application. DUNOD,65
- / 5 /. BENZECRI J-P. et F. , Analyse des correspondances et classification . DUNOD,84
- / 6 /. BOUSSEKSOU B. , Reconnaissance automatique de la parole par la méthode globale.Application aux particularités linguistiques de l'arabe standard.Thèse de magist.83
- / 7 /. COULLON F. , Théorie et traitement des signaux. DUNOD,84
- / 8 /. DUPRAZ J. , Théorie de la communication signaux,bruit et modulation. EYROLLES,73
- / 9 /. FELDBAUM M. , Principes théoriques des systèmes asservis optimaux. MIR. MOUSCOU,73
- / 10 /. GRENIER Y. , MODELISATION ET RECONNAISSANCE DE LA PAROLE. ENST,82
- / 11 /. GUIBERT J. ,La parole: compréhension et synthèse par les ordinateurs. PUF,79
- / 12 /. HATON J-P. , Intelligence artificielle en compréhension automatique de la parole. TSI Rairo,volume 4,N° 3, MAI -JUIN 85 DUNOD.
- / 13 /. KAUFFMANN A. , CRUON R. , La programmation dynamique et ses applications. DUNOD,85
- / 14 /. KUNT M. , Traitement numérique des signaux. DUNOD,81
- / 15 /. LEIPP E. , Le pouvoir séparateur temporel : R ewue d'acoustique N° 50 , 79
- / 16 /. LEVINSON S. , LIBERMAN M. , La reconnaissance de la parole par ordinateur. INSERM,81
- / 17 /. MAX J. et Colaborateurs, Méthodes et techniques du signal "principes généraux et méthodes classiques" T.1 "appareillages,exemples d'applications,méthodes nouvelles" T.2 MASSON,81
- / 18 /. MAYERS C-S. , RABINER L-R. , a level building dynamic time warping algorithm for connected word recognition. IEEE Trans ASSP, April 81,VOL.29,N° 2,PP 284-297

- / 19 /. MAYERS C-S. , RABINER L-R. , Connected digit recognition using a level building DTW algorithm. IEEE Trans SSP , June 81, vol. 29, N°3, PP 351-363
- / 20 /. MENACER A. , Reconnaissance de la parole en mode multi-locuteur par des méthodes globales(mots isolés).thèse de magistère 85.
- / 21 /. MOKEDDEM A. , Analyse de la parole reconnaissance multi-locuteur des mots isolés.Thèse de magistère 85.
- / 22 /. QURRE M. , MERCIER G. , GRESSER J-Y. , Reconnaissance automatique de la parole. CNET mars 74.
- / 23 /. RABINER L-R. , LEVINSON S-E. , Isolated and connected word recognition. Theory and selected applications. IEEE Trans comm , may 81 , vol. 29 , N°5 , PP 693-644
- / 24 /. , REMY C. , La reconnaissance vocale.Derniere étape avant le dialogue. " Micro systèmes" revue N°50 , avril 85.
- / 25 /. SAKOE H. , CHIBA S. , Dynamic programing algorithm optimization for spoken word recognition. IEEE Trans ASSP , february 78 , vol. 26 , N°1 , PP 43-49
- / 26 /. SABROVITCH M. , Optimisation des réseaux EN.S.I.A.G. 79.
- / 27 /. SALAH F. , SAHLI L. , Reconnaissance de la parole par la méthode globale. Projet de fin d'études Janvier 86.
- / 28 /. SCHREIBER H. , Synthèse de la parole. Haut-parleur revue N°1721 et N°1722 , Oct. Nov. 85.
- / 29 /. SCHREIBER H. , Reconnaissance de la parole..Haut-parleur revue , N°1723 Dec. et N°1724 Janv. 86.
- / 30 /. TREMOLIERES J. , La parole artificielle et sa reconnaissance. Electronique application revue, N°46 Fév. 86.