Université de Nancy I — U.E.R. Sciences Mathématiques

Centre de Recherche en Informatique de Nancy

CONTRIBUTION À LA RECONNAISSANCE GLOBALE DE LA PAROLE: MOTS ISOLÉS ET MOTS ENCHAÎNÉS

THESE

présentée et soutenue publiquement le **9 avril 1984**

À L'UNIVERSITÉ DE NANCY I

pour l'obtention du grade de

DOCTEUR INGÉNIEUR EN INFORMATIQUE

par

Joseph DI MARTINO

devant la Commission d'Examen

Président: ..... *J.P. HATON*
Examinateurs: ..... *J.C. DERNIAME*
J. MARIANI
R. MOHR
J.M. PIERREL

Université de Nancy I — U.E.R. Sciences Mathématiques

Centre de Recherche en Informatique de Nancy

CONTRIBUTION À LA RECONNAISSANCE GLOBALE DE LA PAROLE: MOTS ISOLÉS ET MOTS ENCHAÎNÉS

THESE

présentée et soutenue publiquement le **9 avril 1984**

À L'UNIVERSITÉ DE NANCY I

pour l'obtention du grade de
DOCTEUR INGÉNIEUR EN INFORMATIQUE
par
Joseph DI MARTINO
devant la Commission d'Examen

Président: . . . . . *J.P. HATON*
Examinateurs: . . . . . *J.C. DERNIAME*
J. MARIANI
R. MOHR
J.M. PIERREL

J'exprime toute ma reconnaissance à

Monsieur Jean-Paul HATON, Professeur à Nancy I, pour m'avoir proposé un sujet de recherche aussi passionnant, pour le soutien et l'amitié qu'il m'a prodigués.

Monsieur Jean-Claude DERNIAME, Professeur à Nancy I, Directeur du Centre de Recherche en Informatique de Nancy, pour l'intérêt qu'il a manifesté tout au long de mon travail.

Monsieur Joseph MARIANI, Chargé de recherche au C.N.R.S., pour l'honneur qu'il me fait d'assister à ce jury.

Monsieur Roger MOHR, Professeur à l'Institut Polytechnique de Lorraine, et
Monsieur Jean-Marie PIERREL, Professeur à Nancy I, d'avoir accepté de juger ce travail.

Je tiens aussi à remercier Martine TESOLIN pour la réalisation matérielle de cette thèse qu'elle a accomplie avec grand soin ainsi que pour son sourire et sa gaieté.

S O M M A I R E

CHAPITRE I

INTRODUCTION

Le travail que nous avons réalisé est relatif à la reconnaissance globale de la parole dont l'objet est d'identifier une forme vocale inconnue en la comparant dans son ensemble à différentes formes de référence constituant le vocabulaire de l'application. Avant d'en venir aux différents algorithmes de comparaison relatifs à la reconnaissance de mots isolés où à la reconnaissance de mots enchaînés il est nécessaire d'expliciter certaines notions de base concernant la parole pour la bonne compréhension de cette thèse. Ceci est le rôle des chapitres II, III et IV.

Dans le chapitre II nous analyserons le système phonatoire. Nous mettrons en évidence les différentes parties de ce dernier en nous appuyant sur une modélisation proposée par Flanagan [FLAN, 1972] qui permet d'expliciter de façon simple les mécanismes de production de la parole.

Le chapitre III sera consacré à la phonétique. Nous présenterons les différents phonèmes du Français ainsi que leur classification. Ce chapitre a pour but essentiellement d'introduire et d'expliciter les notations phonétiques qui seront employées tout au long de ce mémoire.

Au chapitre IV nous allons nous intéresser à la première étape de l'étude de la parole : l'analyse acoustique. Celle-ci a pour but d'extraire de l'onde vocale des informations pertinentes pour la reconnaissance. Cette phase de traitement de la parole est nommée paramétrisation de l'onde vocale. De nombreuses techniques réalisant cette opération, issues pour la plupart de la théorie du traitement numérique du signal, seront présentées.

Le chapitre V sera entièrement consacré à la reconnaissance de mots isolés. Nous analyserons quelles sont les principales difficultés que l'on rencontre dans ce genre de problème. Nous présenterons ensuite en détail la programmation dynamique qui permet d'apporter

une solution fort élégante au problème du recalage temporel. Nous ménerons à ce sujet une étude comparative des contraintes locales sur les chemins et recalage. Nous verrons aussi comment le problème de la normalisation temporelle peut être abordé à l'aide d'algorithmes sous-optimaux sensiblement différents des algorithmes à optimum global fondés sur la programmation dynamique. Après cela nous présenterons deux algorithmes de reconnaissance de mots isolés à stratégie optimale en insistant sur C.R.S.L.S., l'algorithme que nous proposons, qui a été déduit à partir d'une généralisation des relations de programmation dynamique permettant aux chemins de recalage de ne pas débuter ou se terminer en des points fixes du plan de comparaison. Nous montrerons les avantages de C.R.S.L.S. par rapport aux algorithmes traditionnels de programmation dynamique. Enfin nous donnerons les résultats expérimentaux que nous avons obtenus avec C.R.S.L.S. et d'autres algorithmes de comparaison dynamique.

Au chapitre VI nous aborderons le problème de la reconnaissance de mots enchaînés. Nous verrons que les difficultés rencontrées sont autrement plus complexes que celles qui ont été observées dans le problème de la reconnaissance de mots isolés. Nous donnerons le principe général des algorithmes de reconnaissance de mots enchaînés et nous formaliserons la solution du problème. Nous mettrons en évidence le fait que celle-ci nécessite une double programmation dynamique, l'une pour le niveau mot et l'autre pour le niveau phrase. Nous détaillerons ensuite les principaux algorithmes relatifs au niveau phrase existant à l'heure actuelle. Nous montrerons, après cela, comment les relations généralisées de programmation dynamique que nous avons introduites pour la reconnaissance de mots isolés peuvent être appliquées à la reconnaissance de mots enchaînés. Ceci dit nous exhiberons le principal défaut des algorithmes de reconnaissance de mots enchaînés proposés jusqu'à présent qui est dû au fait qu'ils ignorent le phénomène de la coarticulation. Ce qui a pour effet de limiter leur performance. Nous présenterons alors des solutions pour compenser les distorsions dues à ce phénomène coarti-

culatoire. Pour clore ce chapitre nous donnerons les résultats expérimentaux que nous avons obtenus avec le programme de reconnaissance de mots enchaînés que nous avons développé.

Enfin, au chapitre VII, pour conclure l'ensemble de ce travail nous récapitulerons les résultats intéressants de cette thèse qui, selon nous, ont contribué à la reconnaissance globale de la parole.

CHAPITRE II :
LA PAROLE ET LE SYSTEME PHONATOIRE

II.1.- INTRODUCTION.

Comprendre le mécanisme de production de la parole est un aspect de l'étude de la parole qui a une grande importance. En effet, comme nous le verrons au chapitre IV, c'est grâce à un modèle du conduit vocal, certes imparfait, mais qui a le mérite d'expliquer de façon convenable la production de la parole, que nous serons à-même d'expliciter différents algorithmes de paramétrisation de l'onde vocale. Nous allons voir aussi que l'étude du système de phonation va nous permettre d'identifier les grandes classes de sons élémentaires.

II.2.- MODELISATION FONCTIONNELLE DU SYSTEME DE PRODUCTION DE LA PAROLE.

La figure 1 est une représentation schématique du système phonatoire humain. La cage thoracique en se contractant chasse l'air contenu dans les poumons à travers la trachée. Cet air, avant de passer dans le larynx et la cavité pharyngale, va devoir franchir un obstacle : les cordes vocales. Si celles-ci sont tendues elles se mettent alors à vibrer, laissant passer l'air dans le larynx par saccades ou par impulsions. Si par contre elles sont relachées, l'air passe librement à travers la glotte - orifice créé par les cordes vocales -. Dans le cas où il y a vibration tout se passe comme si le conduit vocal était excité par une source quasi-impulsionnelle. L'air au sortir de la cavité pharyngale aura la possibilité alors de passer dans la cavité buccale et/ou la cavité nasale suivant la position du vélum.

Les sons émis sont habituellement différenciés en deux grandes classes. Une première classe est constituée des sons voisés et une deuxième des sons non voisés.

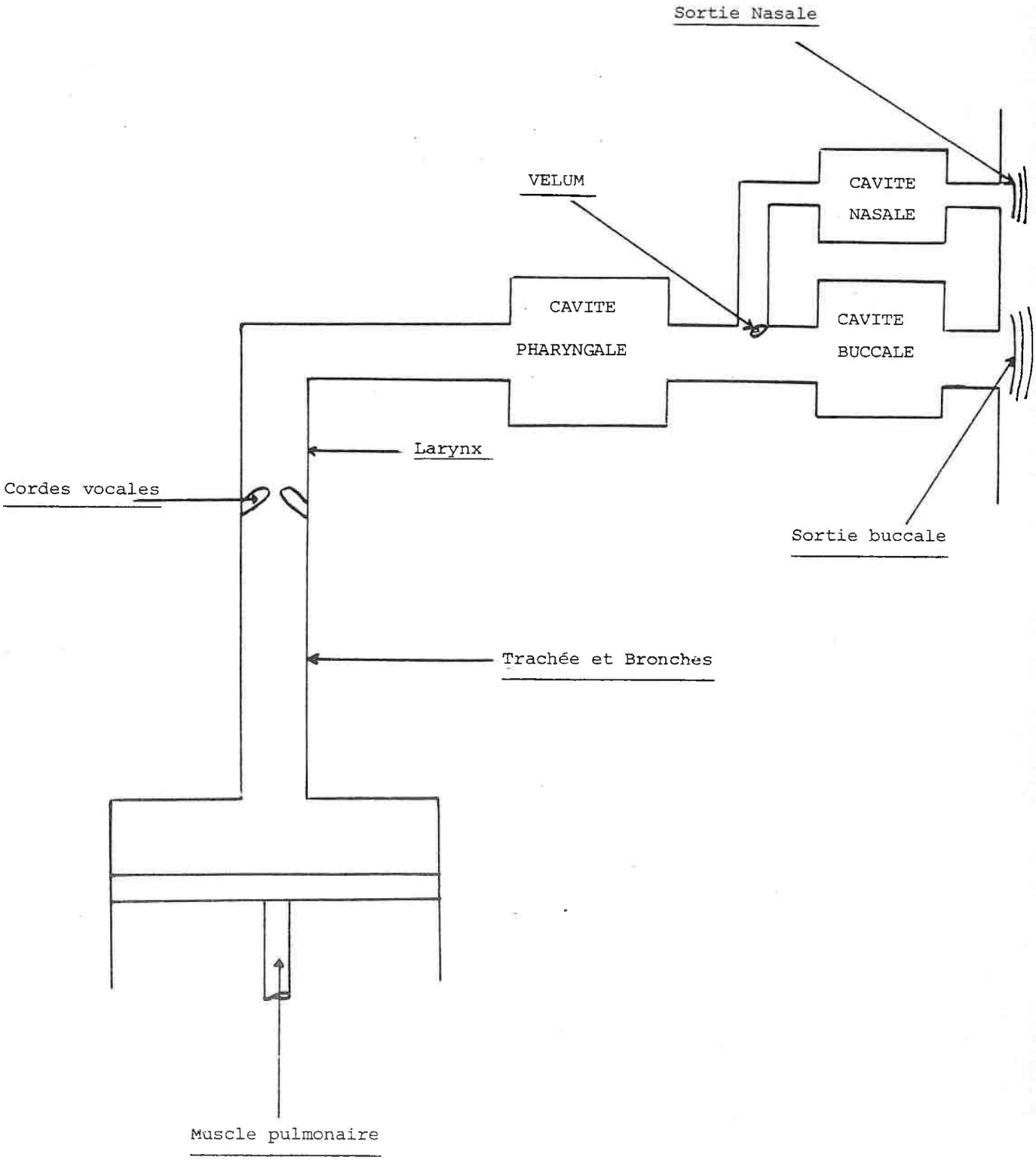


Figure 1 : Modélisation du conduit vocal d'après FLANAGAN [FLAN, 1972].

Dans le cas où le son est tenu, avec vibration des cordes vocales et sans constriction du conduit vocal, les sons émis sont les voyelles qui peuvent être nasales ou non suivant qu'une partie de l'onde vocale aura traversé ou non la cavité nasale.

Dans chacune des deux classes que nous venons de citer on distingue deux autres catégories de sons : les sons fricatifs et les sons occlusifs. On a une fricative lorsque l'air s'échappe du conduit vocal au travers d'une constriction de ce dernier créée par les dents, les lèvres ou la langue et le palais. Par contre on a une occlusive lorsque l'air sort du système phonatoire de façon brutale après avoir vu sa pression croître derrière un obstacle : lèvres, dents, etc. Ainsi /s/ et /ʃ/ sont des exemples de fricatives sourdes et /ʒ/, /z/ des fricatives voisées ou sonores alors que /p/, /t/, /k/ sont des occlusives - on dit aussi plosives - sourdes et /b/, /d/, /g/ des occlusives sonores.

On peut donc considérer de part les développements que nous venons de faire qu'il y a trois modes d'excitation du conduit vocal. Dans le cas de sons voisés du type voyelle, la source d'excitation est localisée au niveau de la glotte. Pour les fricatives sourdes la source se trouve au point de resserement du conduit vocal alors que pour les occlusives sourdes elle se situe au point de fermeture de ce dernier - étant entendu que dans le cas de sons complexes comme les occlusives ou les fricatives voisées deux sources d'excitation sont mises en jeu -.

II.3.- LES FORMANTS.

Le conduit vocal que nous venons de décrire possède des fréquences de résonance dénommées formants qui dépendent de façon importante des articulateurs : les lèvres, la langue, les joues, le vélum etc...

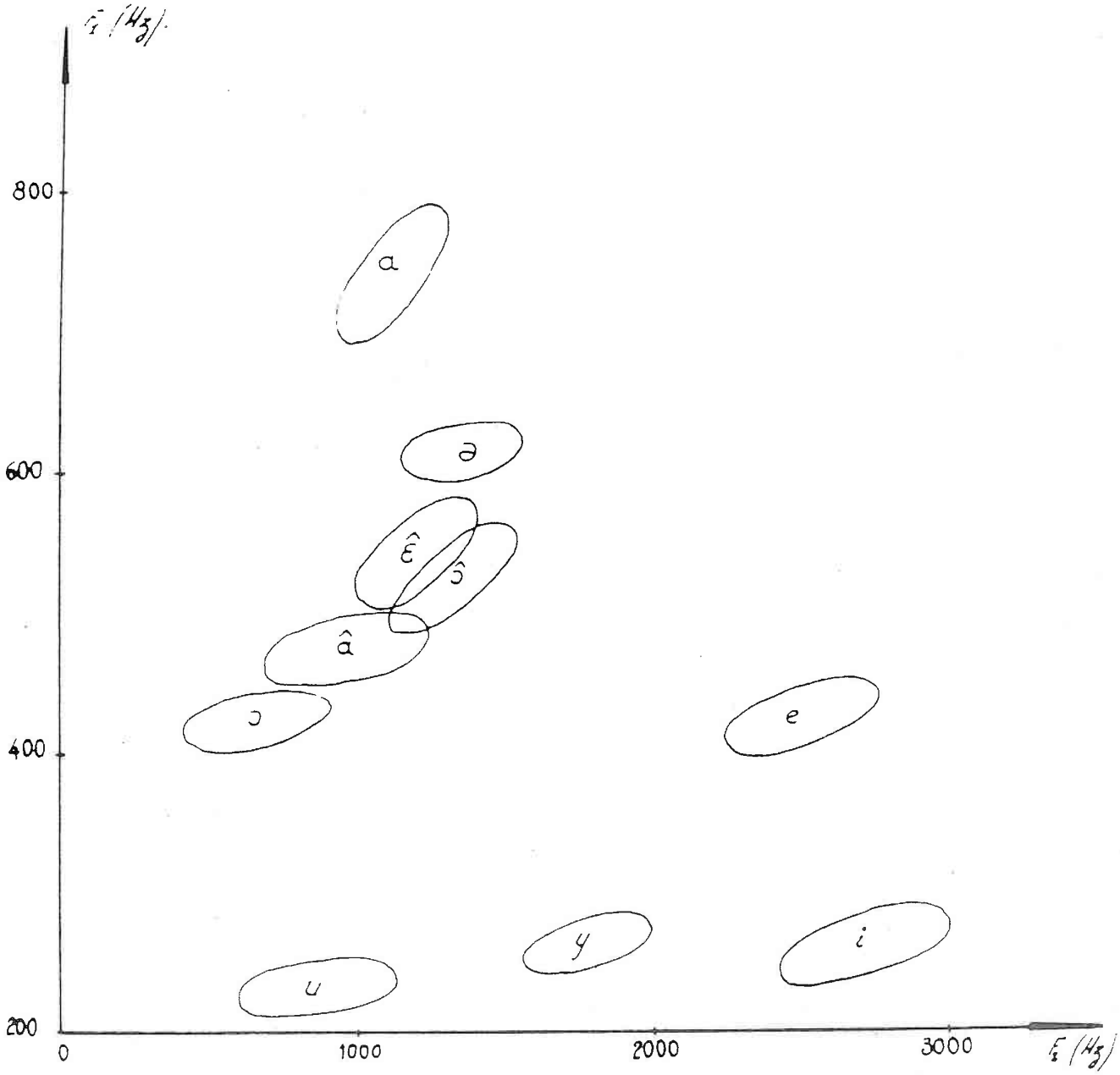


Figure 2 : Représentation des voyelles Françaises dans le plan (1er Formant, 2ème Formant). D'après HATON [HATO, 74]

Les formants permettent d'identifier de façon convenable les voyelles pour un locuteur donné comme le montre la figure 2. Cette propriété importante des formants a été utilisée récemment par Jean-Yves Pérot [PERO, 1984] dans une étude sur la segmentation de la parole.

II.4.- MODELISATION NUMERIQUE DU SYSTEME DE PRODUCTION DE LA PAROLE.

A partir du modèle fonctionnel du système phonatoire que nous avons décrit précédemment il est possible de définir un modèle numérique. La figure 3 visualise un tel modèle.

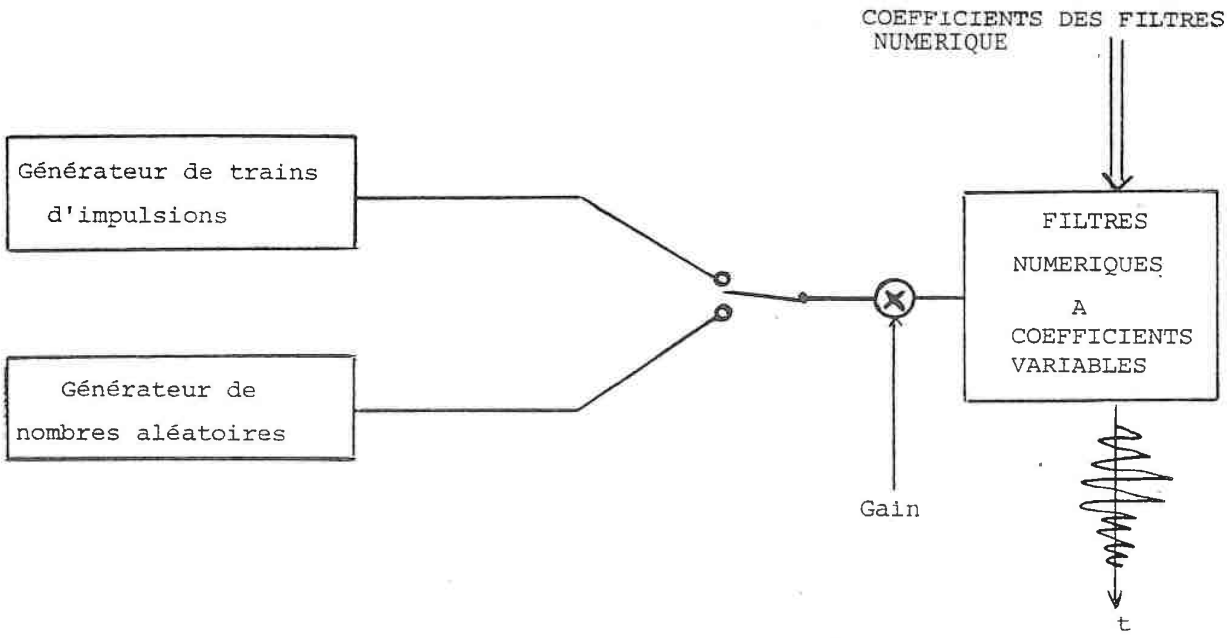


Figure 3 : Modèle numérique du système phonatoire d'après SCHAFER [Scha, 1972].

L'idée de base qui a été émise pour établir celui-ci est l'indépendance des sources d'excitation et du conduit vocal. C'est cette hypothèse qui permet de définir la notion de fonction de transfert du conduit vocal. Celle-ci est valable dans la majorité des cas. Mais malheureusement certains sons transitoires comme les plosives sourdes rendent le modèle incomplet. Les différentes sources du conduit vocal ont été modélisées par des générateurs de trains d'impulsions. Pour les sons voisés un générateur délivre des trains d'impulsions périodiques alors que pour les sons non voisés un générateur fournit des trains d'impulsions aléatoires. Le train d'impulsions périodiques est réglable à l'aide d'un paramètre, le pitch, qui détermine la période qui sépare deux impulsions. Ces différents générateurs sont appliqués à un filtre numérique à coefficients variant dans le temps pour simuler la déformation spatiale du conduit vocal lors de l'élocution. Un dernier paramètre agissant sur l'amplitude des signaux d'excitation permet de contrôler le gain du système.

Le modèle du système de phonation que nous venons de décrire permet de comprendre et de simuler les mécanismes de production de la parole. Comme nous le verrons au chapitre IV, c'est à partir de ce modèle que de puissants algorithmes de paramétrisation de l'onde vocale pourront être déduits.

CHAPITRE III :
QUELQUES ELEMENTS DE PHONETIQUE.

III.1.- INTRODUCTION.

On a vu au chapitre précédent que le système phonatoire pouvait émettre des sons variés comme les voyelles, les fricatives et les occlusives. Les phonéticiens ont recensé et classifié toutes ces entités élémentaires dénommées phonèmes.

Ceux-ci constituent en fait une abstraction de la réalité car un phonème n'a pas de représentation unique. Un phonème verra par exemple sa réalisation varier énormément en fonction du contexte dans le discours continu, en fonction de l'état général du locuteur et évidemment en fonction du locuteur lui-même. Cette pluralité de formes possibles que peut prendre un phonème rend la tâche de décodage acoustico-phonétique extrêmement ardue. A l'heure actuelle il n'existe aucun système de décodage multilocuteur avec plus de 70 % de taux de reconnaissance. Toutefois des progrès intéressants sont à noter de nos jours dans ce domaine : Mohamed Lazrek [LAZR, 1983] a présenté récemment un décodeur acoustico-phonétique monolocuteur ayant un taux de reconnaissance de l'ordre de 70 %.

III.2.- LES PHONEMES DU FRANCAIS.

Le tableau 4 que nous avons emprunté à [LAZR, 1983] donne la liste des phonèmes du Français, les mots-clefs dans lesquels ils apparaissent et les classes phonétiques auxquelles ils appartiennent :

PHONEMES	MOTS-CLEFS	CLASSES
a	plat	
ɑ	mât	
i	il	
y	nu	
ɔ	bol	voyelles
o	eau	
ə	le	
ɛ	lait	
e	blé	
ø	peu	
œ	heure	
u	ou	
ɑ̃	an	
ɔ̃	on	voyelles nasales
ɛ̃	lin	
œ̃	brun	
v	vie	fricatives sonores
z	zéro	
ʒ	je	
f	feu	fricatives sourdes
s	son	
ʃ	chat	
b	bon	occlusives sonores
d	dans	
g	gare	

PHONEMES	MOTS-CLEFS	CLASSES
p	pas	occlusives sourdes
t	tas	
k	coût	
m	Ma	consonnes nasales
n	nous	
ɲ	agneau	
ŋ	camping	
r	rue	liquides
l	lent	
w	voir	Semi-voyelles
j	bailler	
y	huit	

Tableau 4 : Les phonèmes du Français.

On voit apparaître dans le tableau 4 des catégories de sons que nous n'avons pas mentionnées précédemment comme les consonnes nasales, les liquides ou les voyelles. La description articulatoire de ces sons est complexe et dépasse en fait le but de notre propos dans ce chapitre, qui est de donner les éléments de base sur la phonétique pour la bonne compréhension de la suite de ce travail.

CHAPITRE IV :
L'ANALYSE ACOUSTIQUE DU SIGNAL VOCAL.

IV.1.- INTRODUCTION

L'analyse acoustique du signal vocal a pour but d'extraire de ce dernier des informations utiles pour la reconnaissance. On dénomme couramment cette phase paramétrisation de la parole. Diverses techniques ont été proposées. On peut les classer en deux grandes catégories : les techniques temporelles et les techniques fréquentielles. Les techniques de paramétrisation temporelle se proposent d'extraire des informations sur le signal issu directement du microphone. Le nombre de passages par zéro évalué dans une fenêtre temporelle du signal vocal ou de sa dérivée première, ou bien l'emplacement et l'amplitude codés des extrêmas sont les paramètres les plus souvent considérés dans ce type de techniques de paramétrisation. Ces données qui peuvent être utiles pour identifier certains phonèmes sont en fait extrêmement variables en fonction du locuteur et même pour un locuteur donné. Aussi nous ne pensons pas que ces techniques seront celles qui seront retenues dans les systèmes de reconnaissance futurs. En revanche les techniques fréquentielles - et en particulier celles qui s'appuient sur un modèle du système de production de la parole - dont les performances ne sont plus à discuter - nous semblent être les plus intéressantes. Ce sont ces méthodes de paramétrisation qui utilisent généralement de façon importante les résultats de la théorie du traitement numérique du signal que nous nous proposons de décrire dans ce chapitre.

IV.2.- LES METHODES D'ANALYSE SPECTRALE A COURT-TERME PAR TRANSFORMEE DE FOURIER DISCRETE OU PAR BANC DE FILTRES.

IV.2.1.- Introduction.

Les techniques d'analyse spectrale à court-terme de la parole se proposent d'évaluer l'énergie du signal vocal dans des bandes de fréquences bien choisies c'est-à-dire qui correspondent aux bandes de fréquences auxquelles est sensible l'oreille humaine. Les méthodes les plus couramment utilisées consistent à évaluer le spectre à court-terme du signal vocal à l'aide de la transformée de Fourier discrète ou bien à l'aide d'un banc de filtres. D'autres méthodes, plus complexes, comme celles mises en oeuvre dans le vocoder, tout en utilisant la technique du banc de filtres, tiennent compte de plus du modèle de production de la parole.

IV.2.2.- Le spectre à court-terme du signal vocal.

Soit $x(nT)$ le signal vocal échantillonné où T est la période d'échantillonnage. Le spectre de ce dernier est défini par la relation :

$$X(e^{jwT}) = \sum_{n=-\infty}^{+\infty} x(nT)\, e^{-jwnT} \qquad (4.1)$$

qui est la transformée de Fourier de $x(nT)$. Une telle approche pour évaluer le spectre de $x(nT)$ ne s'applique pas bien pour la parole car la relation (4.1) masque l'évolution dans le temps du conduit vocal.

Afin de tenir compte des déformations spatiales du système vocal une notion intéressante a été introduite :
le spectre à court-terme qui est défini par la relation suivante :

$$(4.2) \qquad X(w, nT) = \sum_{r=-\infty}^{n} x(rT)\, h(nT - rT) e^{-jwrT}$$

où h est la réponse impulsionnelle d'un filtre passe-bas de fréquence de coupure w_c, approximant au mieux le filtre passe-bas idéal visualisé par la fibure 5.

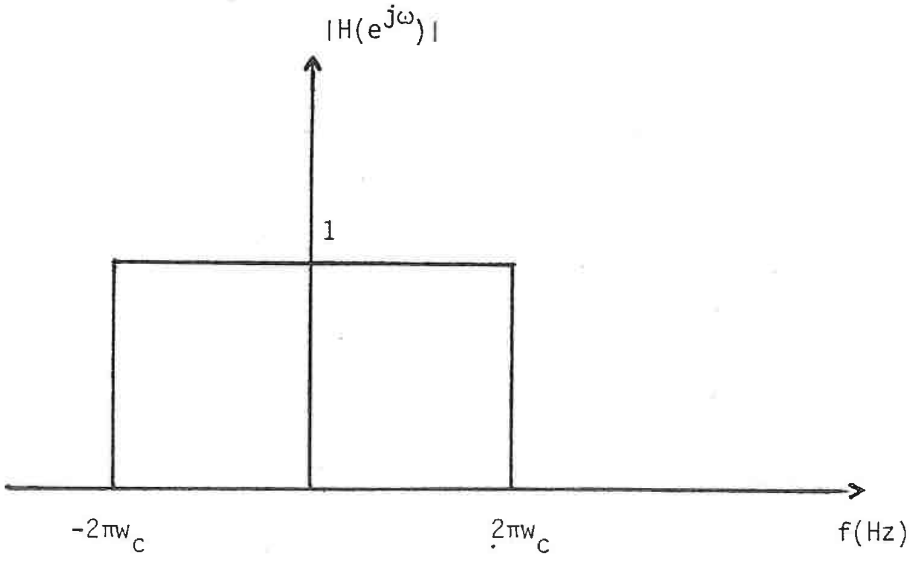


Figure 5 : Le filtre passe-bas idéal.

On peut écrire la relation (4.2) de façon équivalente à l'aide de l'opérateur de convolution $*$ ainsi :

$$(4.3) \qquad X(w, nT) = \left[x(nT).e^{-jwnT} \right] * [h(nT)]$$

La relation (4.3) montre que le spectre à court-terme évalue l'énergie du signal $x(nT)$ depuis l'origine des temps jusqu'au temps d'observation nT dans la bande de fréquence $[2\pi(w - w_c), 2\pi(w + w_c)]$.

En fait dans le cas d'intérêt pratique seule l'évaluation du spectre à court-terme pour quelques pulsations w_k est désirée. Comme l'on montré Rabiner et Gold [RABI, 1975] il est possible d'évaluer la relation (4.2) de façon rigoureuse et efficace à l'aide de techniques de transformée de Fourier rapide à condition évidemment de choisir les N pulsations d'observation régulièrement espacées :

$$(4.4) \qquad w_k = \frac{2\pi}{NT} k \ , \ k = 1, 2, \ldots, N \quad .$$

En effet supposons que h soit définie à l'aide de M points et non nulle pour $0 \leqslant n \leqslant M-1$, il vient :

$$(4.5) \quad X(w_k, nT) = \sum_{m=0}^{E(M/N)+1} \sum_{r=n-(m+1)N+1}^{n-mN} x(rT)h(nT - rT)e^{-jw_k r}$$

où $E[M/N]$ désigne la partie entière de M/N.
En effectuant le changement d'indice suivant :

$$\ell = n - mN - r$$

$X(w_k, nT)$ s'écrit :

$$(4.6) \quad X(w_k, nT) = \sum_{m=0}^{E[M/N]+1} \sum_{\ell=0}^{n-1} x(nT-\ell T-mNT).h(\ell T+mNT).e^{jw_k (\ell - n + mN)}$$

Soit :

$$(4.7)\quad X(w_k,nT) = e^{-j\frac{2\pi}{N}kn} \cdot \sum_{\ell=0}^{N-1} \left[\sum_{m=0}^{E[M/N]+1} x(nT-\ell T-mNT) \cdot h(\ell T+mNT)\, e^{j\frac{2\pi}{N}k\ell} \right]$$

En posant :

$$(4.8)\qquad g(\ell,n) = \sum_{m=0}^{E[M/N]+1} x(nT-\ell T-mNT)h(\ell T+mNT)$$

il vient :

$$(4.9)\qquad X(w_k,nT) = e^{-j\frac{2\pi}{N}kn} \sum_{\ell=0}^{N-1} g(\ell,n).e^{j\frac{2\pi}{N}k\ell}$$

La relation (4.9) exprime le fait que $X(w_k, nT)$ s'écrit comme étant le produit de

$e^{-j\frac{2\pi}{N}kn}$ par la $k^{ième}$ valeur de la transformée de Fourier discrète de la séquence $g(\ell,n)$. On voit donc que le spectre à court-terme du signal vocal peut-être évalué pour des fréquences discrètes normalisées régulièrement espacées dans l'intervalle $0 \leqslant wT \leqslant 2\pi$, par les techniques de transformée de Fourier rapide.

Une nouvelle approche pour évaluer le spectre à court-terme peut être obtenue en décomposant la relation (4.2) en partie réelle et imaginaire. En effet on a :

$$X(w,nT) = R(w,nT) - jI(w,nT) = \sum_{r=-\infty}^{n} x(rT)h(nT-rT)e^{-jwrT}$$

d'où :

$$(4.10) \qquad R(w,nT) = \sum_{r=-\infty}^{n} x(rT)h(nT-rT)\text{Cos } w_r T$$

et

$$(4.11) \qquad I(w,nT) = \sum_{r=-\infty}^{n} x(rT)h(nT-rT)\text{Sin } w_r T \quad .$$

En utilisant l'opérateur de convolution les relations (4.10) et (4.11) s'écrivent :

$$(4.12) \qquad R(w,nT) = \Big[x(nT).\text{Cos}(wnT) \Big] * \Big[h(nT) \Big]$$

et

$$(4.13) \qquad I(w,nT) = \Big[x(nT).\text{Sin}(wnT) \Big] * \Big[h(nT) \Big] \quad .$$

A partir des relations (4.12) et (4.13) on en déduit que le spectre à court-terme du signal vocal peut être déterminé pour différentes fréquences $2\pi\, w_k$ à l'aide d'un banc de filtres constitué d'autant de canaux qu'il y a de fréquences d'observation du spectre. La structure d'un canal de ce banc de filtres apparaît dans la figure 6.

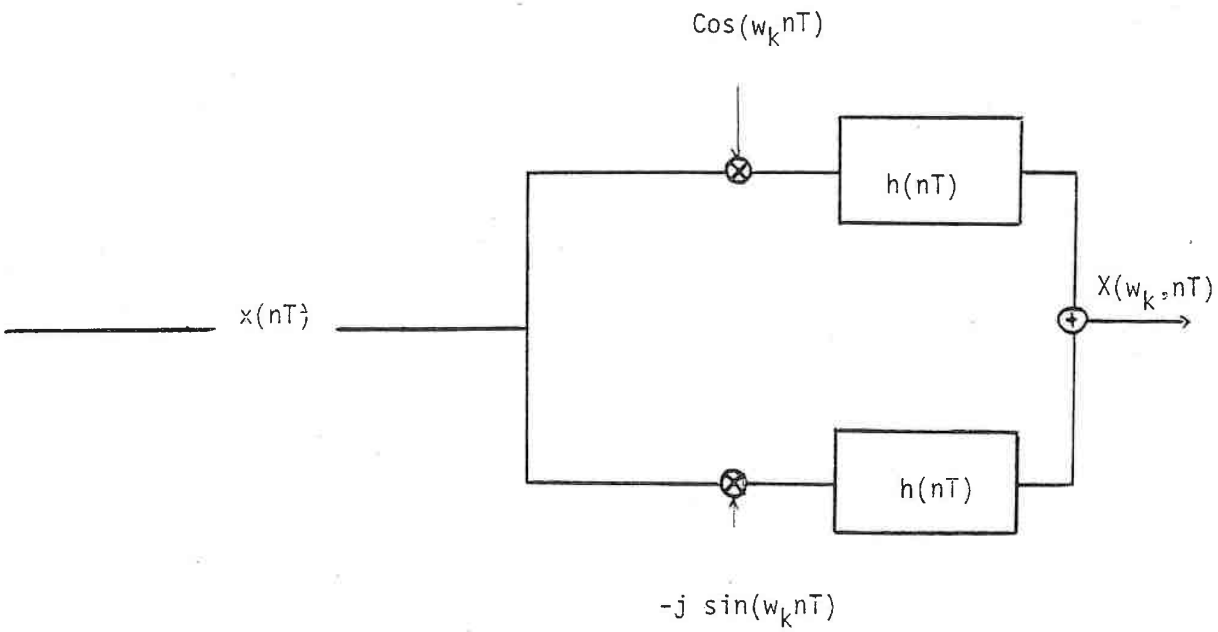


Figure 6 : Un canal du banc de filtres permettant l'évaluation du spectre à court-terme de l'onde vocale.

Les notions d'analyse spectrale a court-terme que nous venons de donner sont essentiellement théoriques car la puissance des calculateurs actuels ne permet pas d'effectuer des transformées de Fourier en un temps de l'ordre de la période d'échantillonnage du signal vocal, de même ne permet pas de réaliser un filtre à réponse impulsionnelle finie avec un nombre de coefficients importants.

IV.2.3.- Le vocoder à canaux : un outil d'analyse spectrale temps réel fondé sur le modèle de production de la parole.

IV.2.3.1.- Introduction.

Le vocoder à canaux est un système qui permet de faire l'analyse spectrale de la parole en temps réel ainsi que la synthèse à partir des données fournies par l'analyse en se fondant sur le modèle de production de l'onde vocale. L'analyse fréquentielle est une analyse à court-terme réalisée à l'aide d'un banc de filtres passe-bande à réponse impulsionnelle infinie. C'est grâce à ce type de filtres qui nécessitent peu de coefficients que l'analyse peut fonctionner en temps réel. Cette dernière a pour but de déterminer essentiellement l'enveloppe du spectre de puissance de la parole, le spectre de phase n'étant pas considéré dans un vocoder du fait que l'oreille n'est pas sensible à celle-ci. La partie synthèse quant à elle a été déduite à partir du modèle du système phonatoire que nous avons décrit au chapitre II.

IV.2.3.2.- La partie analyse du vocoder.

La figure 7 montre le schéma fonctionnel de la partie analyse du vocoder.

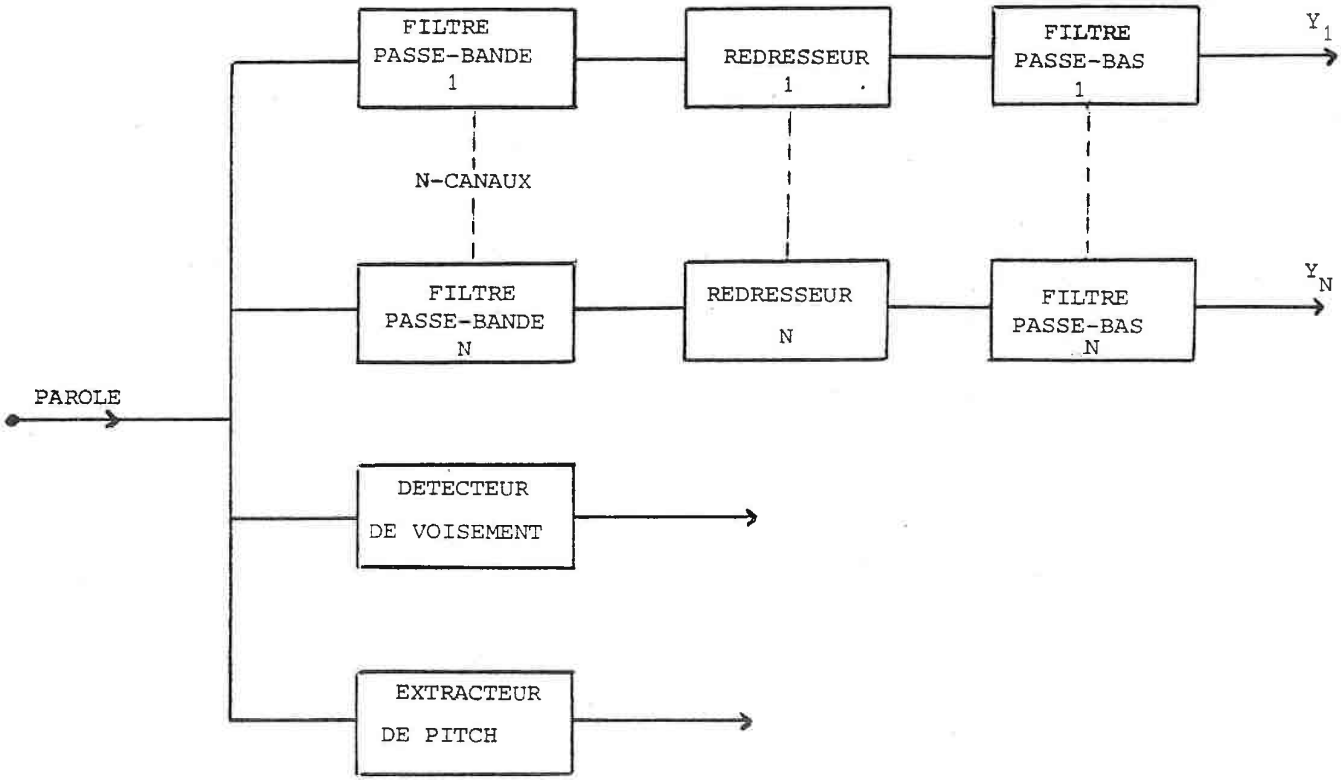


Figure_7 : Partie analyse du vocoder.

La parole passe au travers un banc de filtres passe-bande qui couvrent la plage fréquentielle d'intérêt - généralement de 0 à 5KHZ -. Les signaux issus des filtres passe-bande sont alors redressés pour être filtrés ensuite par des passe-bas qui présentent les mêmes caractéristiques fréquentielles. Ce sont les données Y_i , issues de ces filtres qui caractérisent le spectre de puissance de la parole

à un instant donné. La partie analyse effectue par ailleurs une détection de voisement qui indique si les cordes vocales vibrent, auquel cas un extracteur de pitch détermine la période du fondamental.

On pourrait penser, afin de réaliser un analyseur spectral ayant un bon pouvoir de séparation, utiliser des filtres d'ordre élevé. En fait, de tels filtres ont alors en général une réponse impulsionnelle de durée trop grande par rapport à certaines évolutions spectrales. Un analyseur avec de tels filtres aurait par conséquent une piètre définition temporelle. On voit donc qu'il faut réaliser un compromis dans le choix de l'ordre des filtres afin que le vocoder ait à la fois une bonne définition fréquentielle et temporelle. Un autre problème qui peut nuire à la bonne détection de certains événements temporels réside dans la fait que les bandes passantes des différents filtres sont de largeurs inégales afin de respecter les plages fréquentielles auxquelles est sensible l'oreille humaine : les canaux les plus bas ont une largeur de bande de 150 HZ par exemple alors que les canaux les plus hauts ont des largeurs de bande de 500 HZ. Ceci a pour effet de conférer aux filtres des temps de réponse inégaux. Un moyen simple pour résoudre ce problème consiste à égaliser les temps de réponse des différents filtres en augmentant l'ordre de ceux-ci proportionnellement à la largeur de bande.

Les filtres passe-bas que l'on trouve derrière les redresseurs ont pour rôle d'éliminer la contribution du fondamental dans les diverses bandes considérées. Pour ce faire les filtres doivent présenter une forte atténuation au-delà de 50 HZ - fréquence du fondamental le plus bas pour un être humain - et être à même de suivre l'évolution spectrable du conduit vocal. Une fréquence de coupure de 35 HZ est généralement adoptée.

IV.2.3.3.- La partie synthèse du vocoder.

La figure 8 montre le principe de fonctionnement du module effectuant la synthèse de la parole d'un vocoder.

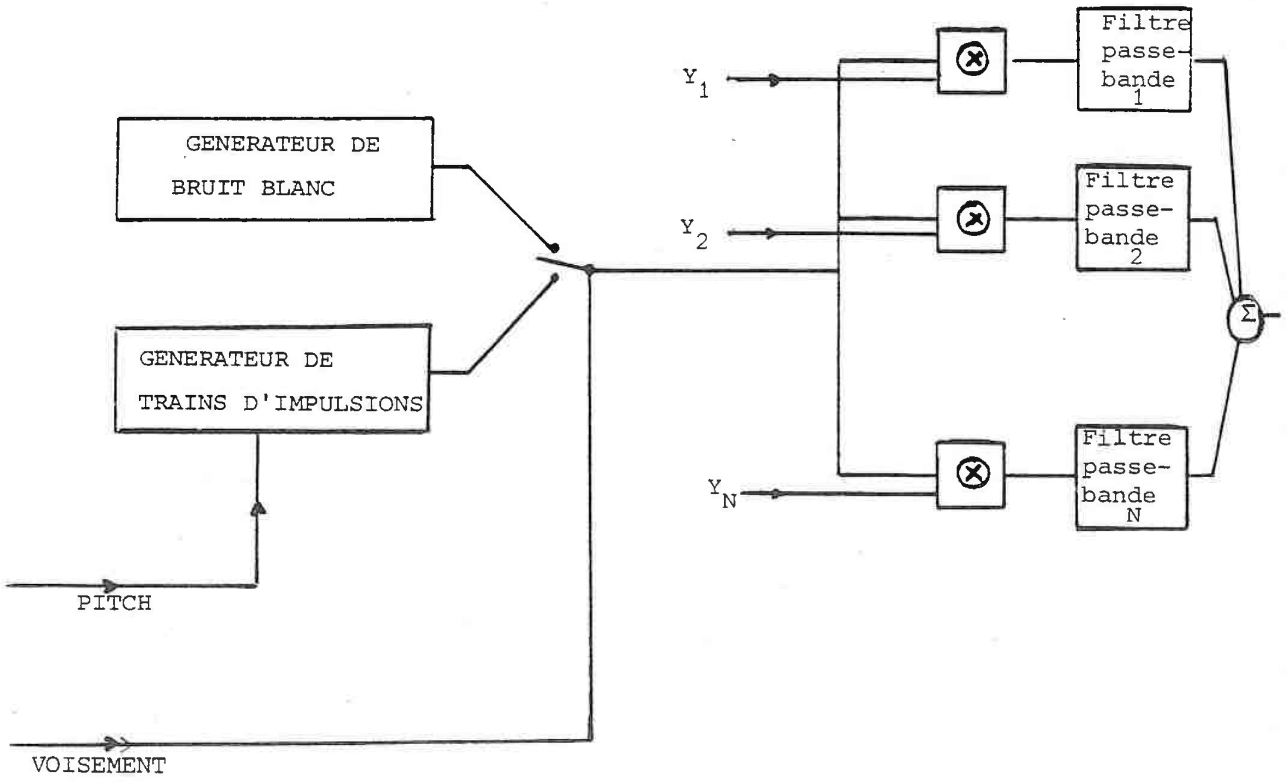


Figure 8 : Schéma fonctionnel du synthétiseur d'un vocodeur.

Le rôle du synthétiseur est de reconstruire le signal de parole à partir des données spectrales fournies par l'analyse. Pour ce faire le détecteur de voisement sélectionne une des sources d'excitation du conduit vocal - qui peut être soit un générateur de bruit blanc, soit un générateur de trains d'impulsions périodiques - simulé par un banc de filtres passe-bande identique au banc de filtres qui a été utilisé lors de la phase d'analyse. Le gain du $k^{ième}$ filtre du synthétiseur est égal à l'énergie spectrale que l'on a obtenue au sortir du $k^{ième}$ filtre passe-bas de l'analyseur. Dans le cas où la source d'excitation utilisée est le générateur de trains d'impulsions, le pitch qui a été déterminé lors de la phase d'analyse permet de régler la période du signal fournie par le générateur.

IV.2.3.4.- La compression de l'information réalisée par un vocoder.

La compression de l'information contenue dans le signal vocal s'effectue lors de l'échantillonnage des valeurs Y_i à l'aide d'une période d'échantillonnage 100 à 200 fois plus grande que celle qui a été utilisée pour discrétiser l'onde vocale. En effet dans le cas où la période pour échantillonner la parole est de 10^{-4}s et qu'il faille 12 bits pour quantifier les échantillons, le nombre de bits nécessaire pour représenter 1 seconde de parole est de $1,2\ 10^5$. En supposant que l'analyseur contienne 16 filtres, que la période d'échantillonnage des données spectrales est de $1,3\ 10^{-2}$s et que le nombre de bits pour coder celles-ci est de 8, le nombre total de bits pour représenter une seconde de parole est de $9,6\ 10^3$. On constate donc que l'on réalise un gain de 12,5.

Les différents synthétiseurs qui ont été réalisés à partir du principe exposé à la figure 8 utilisent un nombre de bits pour coder 1s de parole variant de 2400 à 9600. La qualité du signal synthétisé varie en fait proportionnellement à ce nombre. Plus ce dernier est grand et plus l'écoute est meilleure.

IV.2.4.- Description du vocoder numérique que nous avons utilisé.

IV.2.4.1.- Présentation générale.

Le vocoder que nous avons utilisé pour réaliser nos expériences est un vocoder numérique à 16 canaux. La structure d'un canal de ce vocoder est classique et est visualisée par la figure 9 .

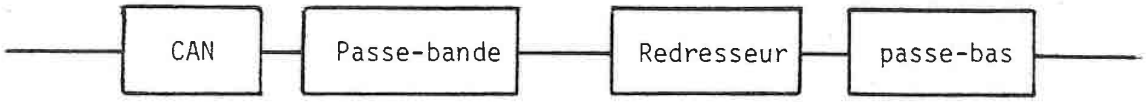


Figure 9 : Structure d'un canal du vocoder.

La fréquence d'échantillonnage du convertisseur est de 16 KHZ. D'après le théorème de Shannon la plage fréquentielle -théorique- d'analyse du vocoder est de 8 KHZ.

IV.2.4.2.- Structure des filtres passe-bande du vocoder.

Le filtre numérique réalisant le filtrage passe-bande est constitué de quatre cellules en cascade du 2ème ordre à double décalage. La figure 10 montre la structure d'une cellule.

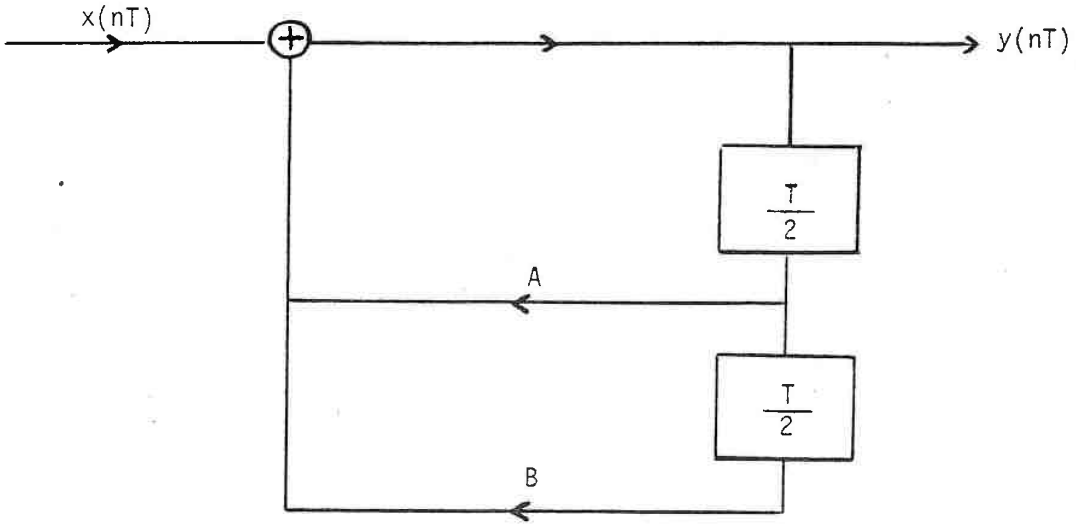


Figure 10 : Structure d'une cellule constituant le filtre passe-bande d'un canal.

Chaque cellule fonctionne en deux temps :

- pendant la première demi-période d'échantillonnage l'entrée est x(nT) avec des coefficients A_1 et B_1 et
- pendant la deuxième demi-période l'entrée est nulle et les coefficients A_2 et B_2.

Une telle structure a été adoptée afin de rendre chaque cellule moins sensible aux erreurs d'arrondi affectant les coefficients.

IV.2.4.3.- Description du filtre passe-bas de sortie d'un canal.

Comme on l'a vu précédemment lors de la description générale du vocoder, le filtre passe-bas a pour rôle d'éliminer les composantes fréquentielles dues à l'excitation. Comme ces dernières apparaissent dans le spectre au-dessus de 50 HZ une fréquence de coupure de 35 HZ est généralement adoptée.

Le filtre passe-bas du vocoder que nous avons utilisé a été réalisé à l'aide de deux cellules du $2^{ème}$ ordre en cascade avec un zéro et deux pôles complexes et présente une fréquence de coupure de 34 HZ à 3 dB.

IV.2.4.4.- La répartition des 16 canaux du vocoder.

La répartition spectrale du vocoder utilisé est donnée par la figure 11.

Les valeurs spectrales au sortir des 16 canaux du vocoder sont échantillonnées à 100 HZ et codées sur 7 bits.

Un programme de transcodage utilisant les symboles donnés par la figure 12 permet, en fonction de la valeur de sortie d'un canal, de réaliser des sonogrammes plus lisibles que les sonogrammes numériques.

La figure 13 montre un exemple de sonogramme obtenu.

N°	Fréquence contrôle	largeur de bande
1	350	200
2	550	200
3	250	200
4	950	200
5	1175	250
6	1450	300
7	1750	300
8	2050	300
9	2350	300
10	2650	300
11	2950	300
12	3300	400
13	3700	400
14	4100	400
15	4700	800
16	4900	1600

Figure 11 : Répartition spectrale des canaux du vocoder.

Plage d'énergie	code
0-15	" "
16-31	" . "
32-47	" : "
48-63	" + "
64-75	" = "
80-95	" c "
96-111	" 3 "
112	" % "

Figure 12 : Table de transcodage des sonogrammes numériques.

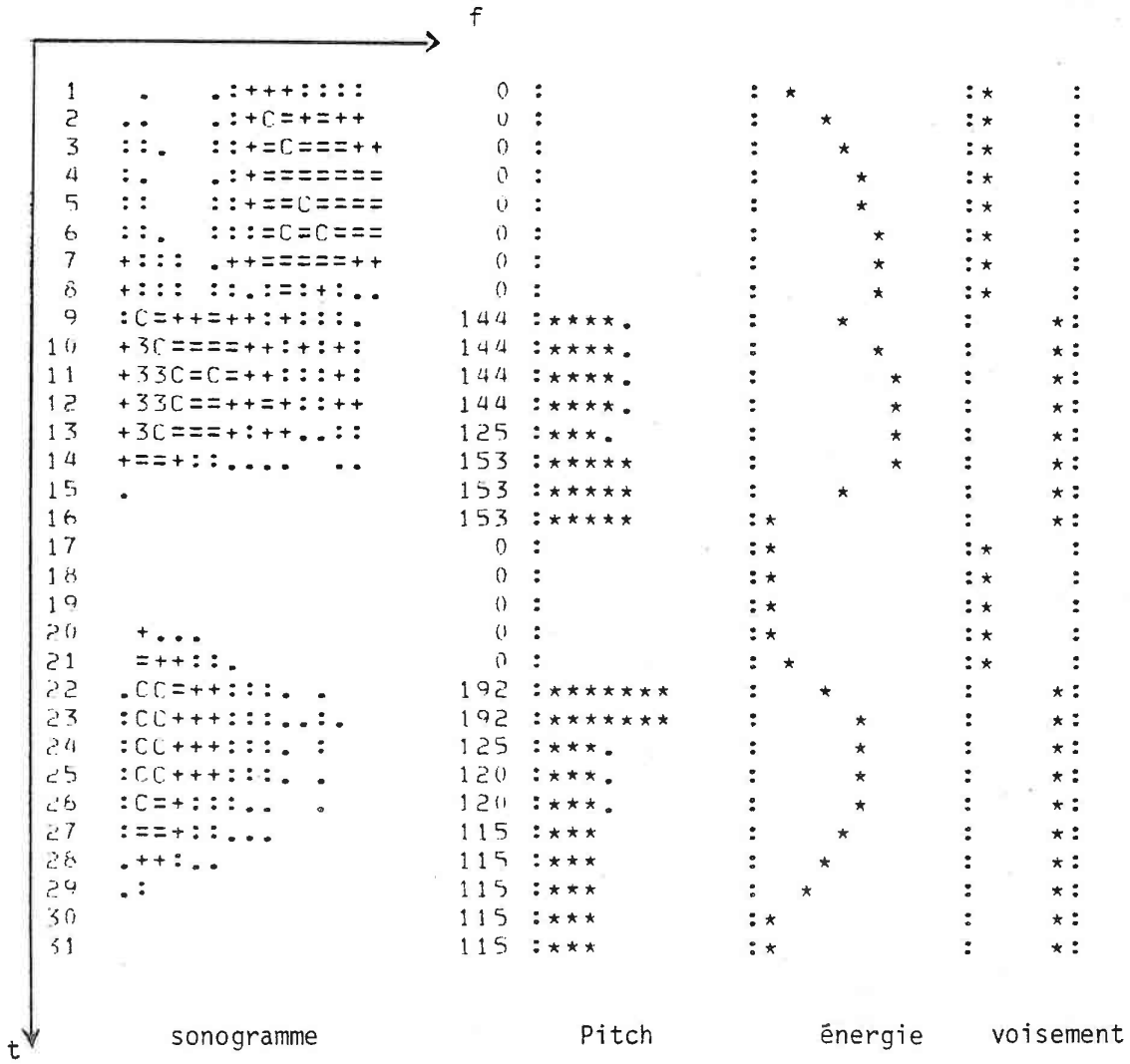


Figure 13 : Sonogramme du mot "Chapeau" obtenu à l'aide du vocoder numérique utilisé.

IV.3.- L'ANALYSE CEPSTRALE DU SIGNAL VOCAL.

IV.3.1.- Introduction.

L'analyse cepstrale du signal vocal est une méthode qui est fondée sur un modèle de production de la parole un peu plus explicite que celui considéré dans le vocoder. Toutefois cette analyse est nettement plus élégante que celle qui a été utilisée dans le vocoder du fait que l'extraction du pitch et la détermination des paramètres temporels caractérisant la fonction de transfert du conduit vocal sont réalisées par un même algorithme. Le degré de complexité de plus au niveau réalisation de l'analyseur cepstral est de beaucoup inférieur à celui caractérisant l'élaboration d'un vocoder classique.

IV.3.2.- Principe de l'analyse.

Le modèle de production de la parole permet d'établir entre le signal vocal $x(nT)$, la réponse impulsionnelle du conduit vocal $h(nT)$, caractérisant ce dernier dans un certain laps de temps, et le signal d'excitation $e(nT)$ la relation classique de convolution d'un système linéaire à savoir :

$$x(nT) = h(nT) * e(nT) \quad . \tag{4.14}$$

La méthode qui va être utilisée pour effectuer l'analyse va consister à effectuer dans la fenêtre temporelle d'analyse une déconvolution de la relation (4.14) afin de séparer les contributions du conduit vocal et de l'excitation. Le processus de déconvolution est montré sur la figure 14.

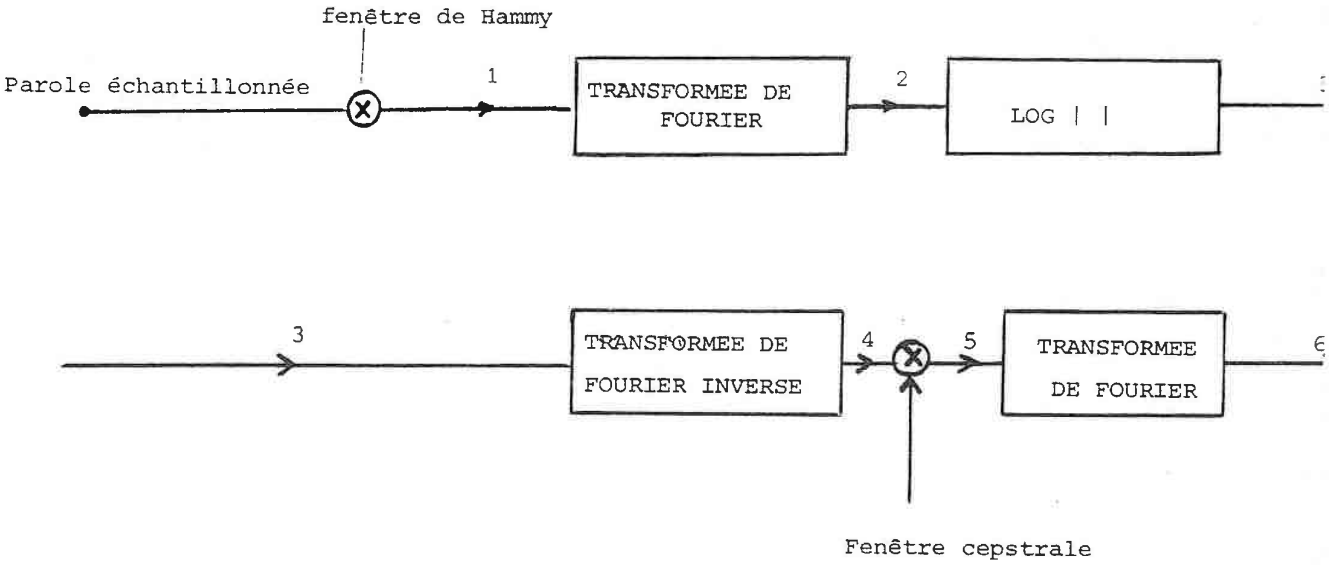


Figure 14 : Processus de déconvolution utilisé dans l'analyse cepstrale du signal vocal.

Une portion du signal de parole échantillonnée est extrait à l'aide d'une fenêtre temporelle. Les échantillons considérés sont pondérés par la fonction de Hamming afin de limiter les oscillations parasites dans le domaine fréquentiel dues au phénomène de Gibbs : c'est le rôle du premier multiplieur de la figure 14. La séquence obtenue est transmise à l'entrée d'un algorithme de transformée de Fourier discrète. Au point 2 on dispose donc de la relation suivante :

$$X(e^{jw_kT}) = H(e^{jw_kT}) . E(e^{jw_kT}) \qquad (4.15)$$

où X, H, E sont les transformées de Fourier respectivement de x, h et e . Avec $w_k = \frac{2\pi}{NT} k$, N étant le nombre de points de l'algorithme de transformée de Fourier. Au point 4 grâce à l'opérateur logarithme les modules des spectres du conduit vocal et de l'excitation sont séparés. En effet on a :

$$\text{Log}\,|X(e^{jw_kT})| = \text{Log}\,|H(e^{jw_kT})| + \text{Log}\,|E(e^{jw_kT})| \qquad (4.16)$$

En désignant $\text{Log}\,|X(e^{jw_kT})|$ par $\tilde{X}(k)$,

$\text{Log}|H(e^{jw_kT})|$ par $\tilde{H}(k)$, $\text{Log}|E(e^{jw_kT})|$ par $\tilde{E}(k)$,

il vient :

$$\tilde{X}(k) = \tilde{H}(k) + \tilde{E}(k) \qquad (4.17)$$

Ainsi on voit qu'à partir de la séquence $x(nT)$ on a réussi à créer une séquence $\tilde{X}(k)$ qui lie de façon additive - et donc très simple - les contributions du conduit vocal et de l'excitation. La transformée de Fourier inverse qui précède le point 4 a pour but de déterminer les séquences temporelles $\tilde{x}(nT)$ - qui est le cepstre du signal $x(nT)$ -, $\tilde{h}(nT)$ et $\tilde{e}(nT)$ correspondant respectivement à $\tilde{X}(k)$, $\tilde{H}(k)$ et $\tilde{E}(k)$. Du fait que l'opérateur de transformée de Fourier inverse est linéaire il vient :

$$\tilde{x}(nT) = \tilde{h}(nT) + \tilde{e}(nT) \qquad (4.18)$$

Au point 4 du processus on a donc réalisé en quelque sorte la déconvolution de la relation (4.14). Il ne reste plus, pour terminer le processus, qu'à séparer $\tilde{h}(nT)$ et $\tilde{e}(nT)$. Cette opération est

réalisée par la fenêtre cepstrale qui agit sur le deuxième multiplieur situé entre les points 4 et 5.

IV.3.3.- Conclusion.

L'analyse cepstrale du signal vocal est une méthode puissante qui permet à la fois d'extraire le pitch et le spectre du conduit vocal. Comme au niveau de la reconnaissance l'évolution spatiale du conduit vocal est une information suffisante, les signaux cepstraux permettent de paramétriser l'onde vocale de façon très judicieuse.

IV.4.- L'ANALYSE PREDICTIVE LINEAIRE DE LA PAROLE.

IV.4.1.- Introduction.

Dans l'analyse cepstrale, le spectre du signal vocal est obtenu sans qu'il soit nécessaire de définir à priori la fonction de transfert du conduit vocal. L'analyse prédictive linéaire suit une approche différente dans la mesure où le conduit vocal - modélisé par un tuyau sonore à sections variables - est représenté par la fonction de transfert d'un prédicteur linéaire qui permet à partir des p échantillons précédents l'échantillon n d'en déduire cet échantillon.

IV.4.2.- Principe de l'analyse prédictive de la parole.

La fonction de transfert du modèle du conduit vocal qui est adoptée dans cette analyse est de la forme :

$$(4.19) \qquad H(z) = \frac{X(z)}{E(z)} = \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}}$$

où $X(z)$ est la transformée en z du signal de parole vu à travers une fenêtre temporelle d'analyse de durée $\mathcal{T}$ et $E(z)$, la transformée en z de l'excitation.

Dans le modèle les données qui sont connues ou supposées connues sont $X(z)$ et $E(z)$. Ceci signifie que le pitch ou le voisement doivent être déterminés par un autre algorithme. Les inconnues du problème sont les p coefficients de prédiction a_k qui caractérisent le conduit vocal durant le laps de temps $\mathcal{C}$. Pour évaluer ceux-ci une minimisation d'un signal d'erreur par la méthode des moindres carrés est utilisée.

l'équation aux différences caractérisant le modèle est la suivante :

$$x(nT) = \sum_{k=1}^{p} a_k \, x(nT - kT) + e(nT) \qquad (4.20)$$

La figure 15 visualise le modèle du système de production de la parole adopté dans l'analyse prédictive linéaire mettant en oeuvre la relation (4.20).

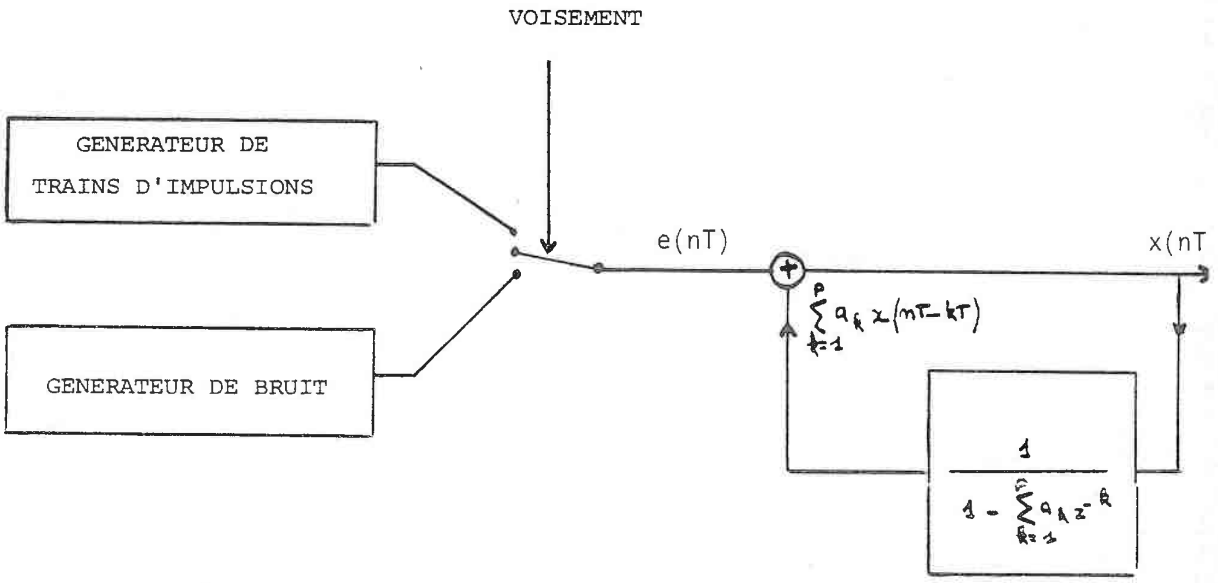


Figure 15 : Modélisation du système de production de la parole par l'analyse prédictive linéaire.

La relation (4.20) peut s'écrire :

$$x(nT) = \tilde{x}(nT) + e(nT) \tag{4.21}$$

où : $\tilde{x}(nT) = \sum_{k=1}^{p} a_k \, x(nT - kT)$ est une valeur fournie par le prédicteur linéaire et $e(nT)$ est l'erreur résiduelle entre la valeur réelle de sortie et la valeur prédite.

L'analyse prédictive se propose de déterminer les coefficients a_k en minimisant la somme des erreurs au carré pour tous les échantillons c'est-à-dire la quantité :

$$E = \sum_{n=1}^{p} \left[x(nT) - \sum_{k=1}^{p} a_k \, x(nT - kT) \right]^2 \quad . \tag{4.22}$$

Le vecteur $\bar{a}$ de composantes a_k minimisant E s'obtient en annulant toutes les dérivées partielles de E par rapport aux a_j c'est-à-dire :

$$\frac{\partial E}{\partial a_j} = 0 \quad \text{pour} \quad j = 1, 2, \ldots, p \quad . \tag{4.23}$$

En développant la relation (4.23) il vient :

$$(4.24)\quad \begin{cases} \sum_{k=1}^{p} a_k \sum_{n=1}^{p} x\left[(n-k)T\right]\; x\left[(n-j)T\right] \\ \\ = \sum_{n=1}^{p} x(nT).\; x\left[(n-j)T\right] \\ \\ \text{pour } j = 1, 2, \ldots, p \end{cases} .$$

Si l'on désigne par Φ la matrice de composantes :

$$(4.25)\qquad \varphi_{ij} = \sum_{n=1}^{p} x\left[(n-i)T\right] x\left[(n-j)T\right]$$

avec i et j variant de 1 à p et par

$$(4.26)\qquad \psi = \varphi_{0j}$$

la relation (4.27) s'écrit :

$$(4.27)\qquad \Phi.\vec{a} = \psi \qquad .$$

La matrice Φ, dénommée matrice d'autocorrélation est symétrique, définie positive. Il est donc possible d'inverser la relation (4.27).

Les coefficients a_k fournis par l'analyse prédictive linéaire de la parole constituent une représentation tout à fait convenable de la parole. Nombre de synthétiseurs d'ailleurs fonctionnent à partir du modèle que nous venons de décrire.

CHAPITRE V :

LA RECONNAISSANCE DE MOTS ISOLES

V.1.- INTRODUCTION

La reconnaissance de mots isolés est certainement le domaine de la reconnaissance de la parole qui a fait l'objet du plus grand nombre d'études. La raison à cela est due au fait que dès le début des recherches sur la parole, les chercheurs ont su faire des petits systèmes à même de reconnaître quelques dizaine de mots avec des taux de reconnaissance honorables dans un contexte monolocuteur. Les progrès encourageants qui pouvaient être observés d'année en année dans ce domaine a attiré l'attention de nombreux chercheurs et ingénieurs qui, par ailleurs, s'étaient rendus compte de l'énorme complexité de la reconnaissance du discours continu. Actuellement ce domaine est en pleine expansion tant et si bien que les algorithmes qui sont proposés une année se trouvent dépassés l'année suivante, ce qui constribue de façon très favorable au caractère passionnant du problème.

V.2.- PRINCIPE DE LA RECONNAISSANCE DE MOTS ISOLES.

Le principe de la reconnaissance de mots isolés consiste à comparer une forme inconnue à un ensemble de formes de référence qui constituent le dictionnaire ou encore le vocabulaire des mots connus. La forme de référence qui aura satisfait au mieux les critères de comparaison sans faire l'objet de dépassement de certains seuils de rejet éventuels sera décrétée être similaire à la forme inconnue qui a été présentée en entrée de l'algorithme de comparaison. Dans le cas où toutes les formes de référence sont éliminées lors de la comparaison, la forme vocale testée sera rejetée par l'algorithme comme n'étant pas connue par le système de reconnaissance. La figure 16 montre schématiquement le principe et la reconnaissance de mots isolés.

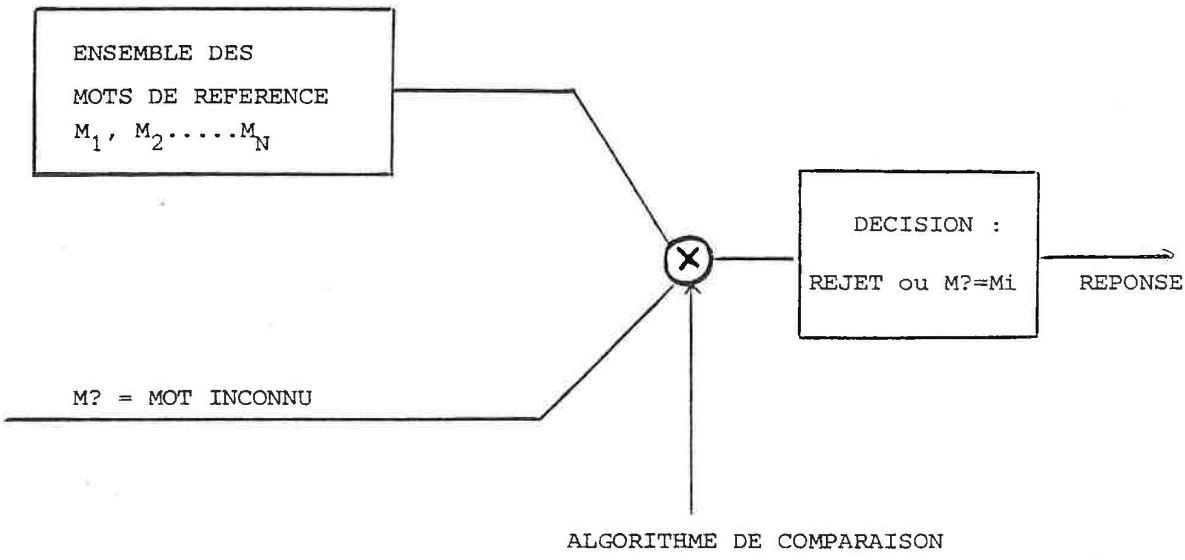


Figure 16 : Principe de la reconnaissance de mots isolés.

Les formes vocales qui sont considérées lors de la comparaison ont été au préalable codées par l'analyseur acoustique sous forme d'une suite de vecteurs, - dont les composantes sont par exemple les sorties d'un vocodeur à canaux ou bien les coefficients fournis par une analyse prédictive ou cepstrale - caractérisant l'onde vocale durant un certain laps de temps. L'algorithme de comparaison doit donc être à même de comparer des suites de vecteurs.

V.3.- LES DIFFICULTES DU PROBLEME.

Les difficultés que l'on rencontre en reconnaissance de mots isolés proviennent de la variabilité très importante de certains paramètres caractérisant l'élocution comme :

- le débit de la parole,
- la hauteur de la voie,
- le rythme.

La non-constance de la vitesse d'élocution se traduit par l'obtention de formes vocales qui ont des longueurs et des rythmes différents lors d'élocutions d'un même mot pour un même locuteur et a fortiori pour des locuteurs différents.

Au début des recherches sur la reconnaissance de la parole les chercheurs pensaient que les distorsions temporelles engendrées par le caractère variable de la vitesse d'élocution étaient essentiellement linéaires. Malheureusement, on s'est aperçu très vite qu'il n'en est rien. En effet, alors que certains phonèmes ne sont guère altérés lors d'une élocution rapide, par rapport à une élocution normale, d'autres par contre sont très fortement comprimés ou déformés. Les figures 17-a et 17-b mettent en évidence les distorsions temporelles apparaissant lors de deux élocutions du mot "chapeau" prononcé par un même locuteur. En comparant celles-ci, en effet, on observe que la fricative sourde /ʃ/ - prélèvements 1 à 8 sur 17-a et 1 à 4 sur 17-b - et la voyelle /a/ - prélèvements 8 à 14 sur 17-a et 5 à 9 sur 17-b - ont été considérablement comprimées tandis que la plosive /p/ - prélèvements 15 à 20 sur 17-a et 10 à 15 sur 17-b - n'a subi aucune déformation temporelle.

La variabilité de la hauteur de la voie qui se traduit sur le signal temporel par des différences de dynamique au niveau de l'amplitude introduit un deuxième type de distorsions.

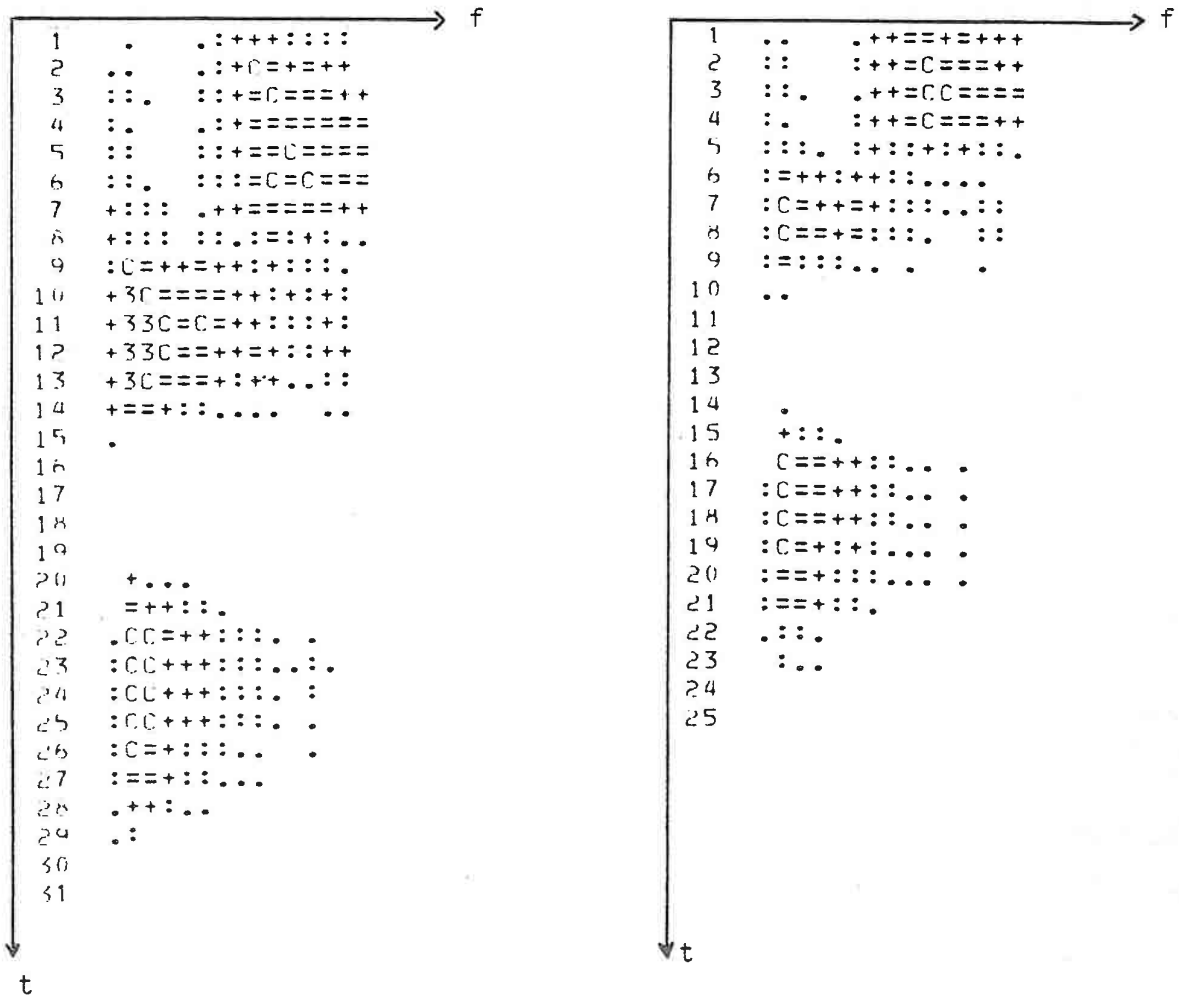


Figure 17 : Distorsions temporelles apparaissant lors de deux élocutions différentes du mot "chapeau".

Un autre genre de difficultés - non inhérent à la prononciation - intervient lors de la phase d'analyse du signal vocal. En effet il arrive que l'algorithme de détection de parole ne parvienne pas à détecter certains phonèmes de début ou de fin de forme faiblement énergétiques. La forme au sortir de celui-ci subit dans ce cas une troncature qui peut par la suite fausser la reconnaissance.

On dénomme ce genre de problème, erreur de détection parole - non parole.

Ainsi, comme on le voit, la tâche de l'algorithme de comparaison est difficile dans la mesure où il doit être à-même de tenir compte de toutes les distorsions possibles pour pouvoir effectuer une reconnaissance correcte. Toutes les difficultés que nous venons de citer ne sont pas encore résolues à l'heure actuelle. Toutefois, des solutions ont été proposées pour compenser les distorsions dues à la variabilité de la vitesse d'élocution et aux erreurs de détection parole - non parole de l'analyseur acoustique.

V.4.- LE RECALAGE TEMPOREL.

V.4.1.- Principe du recalage temporel.

Les variations de la vitesse d'élocution introduisent, on l'a vu, des distorsions essentiellement non linéaires. Afin de compenser celles-ci, le recalage temporel se propose d'effectuer une synchronisation des échelles des temps de deux formes à comparer comme le montre la figure 18.

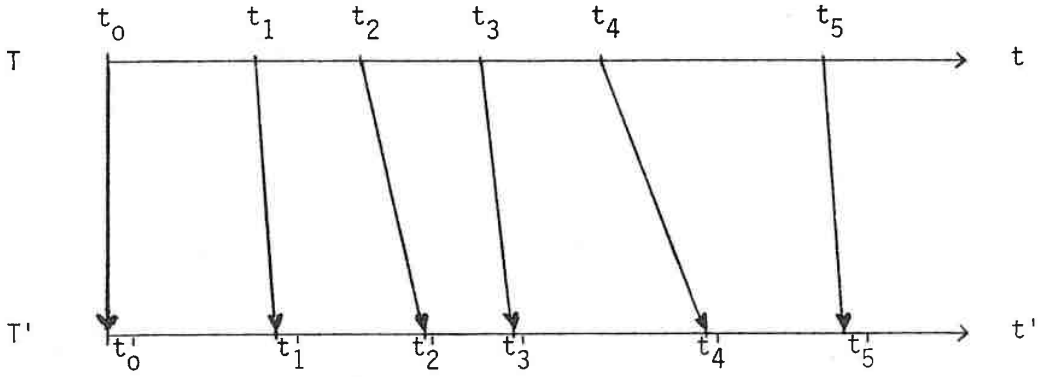


Figure 18 : Mise en correspondance des échelles des temps des formes à comparer.

V.4.2.- Les chemins de recalage.

Désignons par $t(1), t(2),\ldots, t(I)$ la suite des vecteurs fournis par l'analyseur acoustique caractérisant la forme T, I étant la longueur en nombre de prélèvements de la forme T, et de même par $r(1), r(2),\ldots, r(J)$ la suite de vecteurs caractérisant la forme R, J étant la longueur en nombre de prélèvements de la forme R. Le recalage temporel se propose d'établir entre ces suites de vecteurs une correspondance qui maximise les meilleures mises en correspondance. Il s'agit donc de trouver parmi toutes les fonctions W ainsi définies :

$$\mathbb{N}_{I+J} \xrightarrow{\quad W \quad} \mathbb{N}_I \times \mathbb{N}_J$$

$$k \longrightarrow W(k) = (i(k), j(k))$$

- où : - $\mathbb{N}_p = \{1, 2,\ldots, P\}$

et - $W(k) = (i(k), j(k))$ indique que la fonction de recalage

met en coïncidence le $i(k)^{ième}$ prélèvement de la forme T et le $j(k)^{ième}$ prélèvement de la forme R - , la fonction optimale $\hat{W}$, au sens d'une certaine métrique, qui réalise la coïncidence optimale entre les deux formes à comparer.

Si l'on porte sur un axe horizontal les différents vecteurs de la forme T - un point représentant un vecteur - et sur un axe vertical les vecteurs de la forme R , une représentation de la fonction W est un chemin dans le plan ainsi défini. La figure 19 illustre un exemple de chemin de recalage.

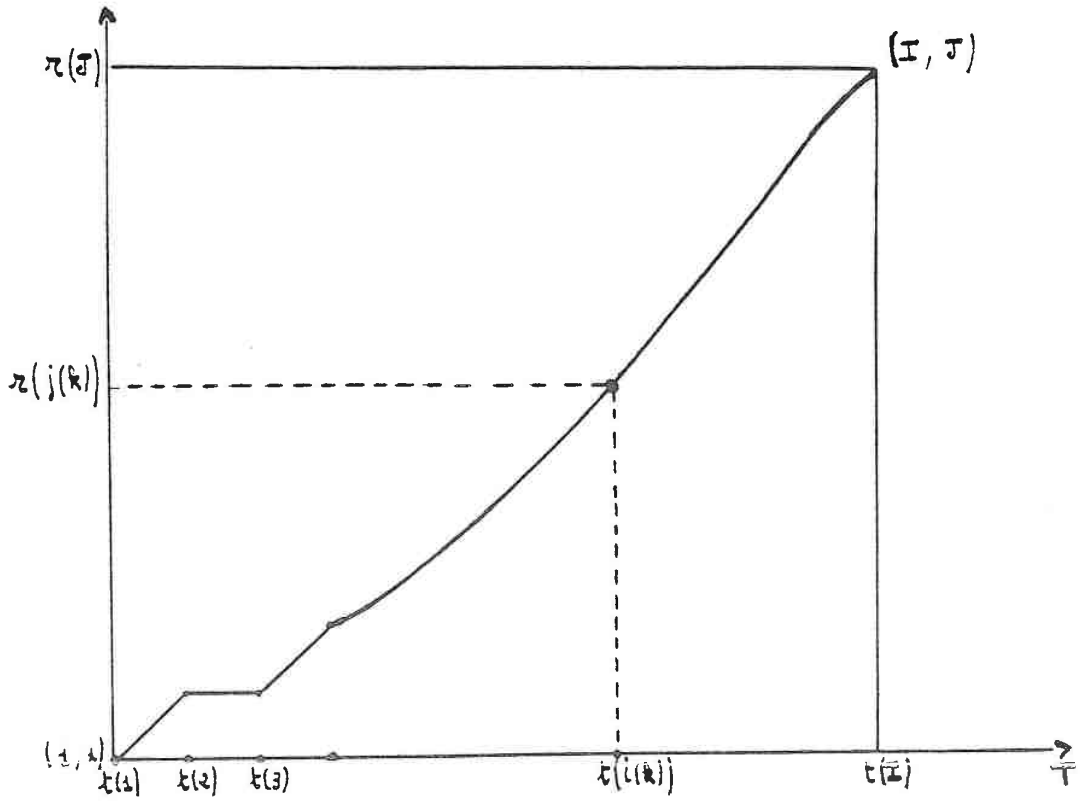


Figure 19 : Un exemple de chemin de recalage.

V.4.3.- Les contraintes imposées aux chemins de recalage.

Afin que la fonction de recalage respecte l'évolution dans le temps du signal vocal, celle-ci est soumise à des conditions de monotonie exprimées par les relations suivantes :

(5.1.a) $\quad i(k-1) \leq i(k)$

(5.1.b) $\quad j(k-1) \leq j(k)$

qui l'oblige a être monotonement croissante.

La fonction de recalage se doit aussi de ne pas effectuer des compressions ou des dilations irréalistes comme le montre la figure 20.

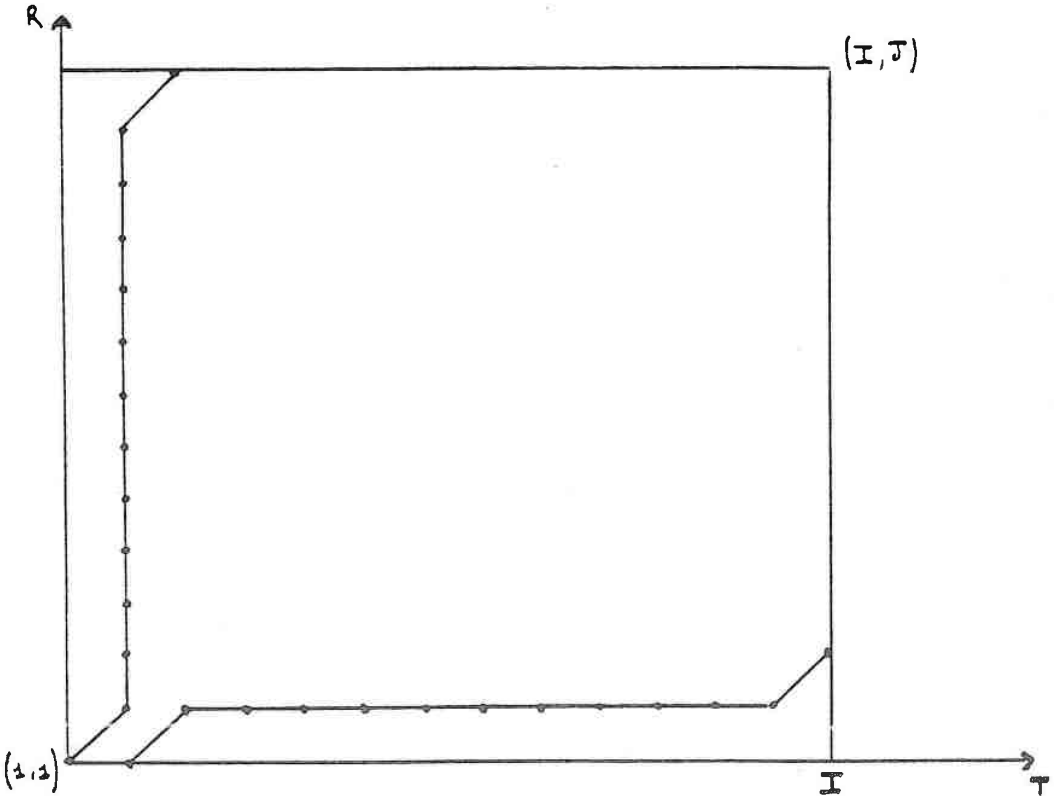
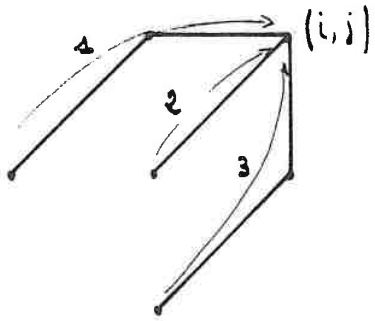
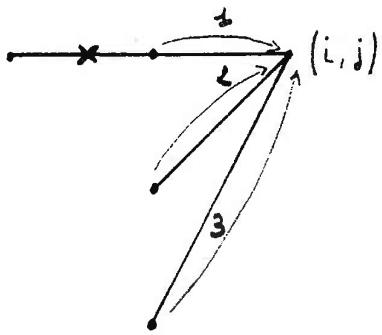


Figure 20 : Exemples de compression ou de dilatation non réalistes

Pour qu'il en soit ainsi, celle-ci est assujettie à des contraintes locales qui lui empêche d'effectuer certains déplacements locaux. On interdit souvent par exemple au chemin de recalage d'aller consécutivement deux fois dans la même direction si le sens du déplacement est horizontal ou vertical. La figure 21 montre deux exemples de contraintes qui sont très souvent utilisées.



CONTRAINTE DE SAKOE
ET CHIBA [SAKO, 1978]



CONTRAINTE D'ITAKURA
[ITAK, 1975]

Figure 21 : Exemples de contraintes locales.

Les motifs visualisés par la figure 21 indiquent qu'un chemin de recalage ne peut aboutir au point (i,j) qu'en suivant obligatoirement les chemins locaux 1, 2 ou 3. Plus précisément, si la contrainte locale est la contrainte de Sakoe et Chiba, un chemin de recalage ne peut accéder au point (i,j) qu'en passant par (i-2, j-1) et (i-1, j) ou par (i-1, j-1) ou encore par (i-1, j-2) et (i, j-1) ; et s'il s'agit de la contrainte d'Itakura, par les points (i-1, j) - à condition que le chemin ne soit pas passé précédemment par le point (i-2, j) -, (i-1, j-1) ou (i-1, j-2).

Si par exemple la contrainte à laquelle est assujettie la fonction de recalage est la contrainte locale d'Itakura alors les relations suivantes doivent être satisfaites :

(5.2.a) $\quad j(k+1) - j(k) = 0, 1, 2 \quad$ si $\quad j(k) \neq j(k-1)$

(5.2.b) $\quad j(k+1) - j(k) = 1, 2 \quad$ si $\quad j(k) = j(k-1)$

(5.2.c) $\quad i(k+1) - i(k) = 1 \quad .$

D'autre part, afin que la fonction de recalage tienne compte de l'ensemble des prélèvements de chacune des deux formes, des contraintes aux frontières lui sont imposées. Ces dernières l'assujettissent à mettre en correspondance les prélèvements terminaux de chacune des deux formes.

Ces contraintes sont exprimées par les relations (5.3) et (5.4) :

(5.3) $\quad \begin{cases} i(1) = 1 \\ j(1) = 1 \end{cases}$

(5.4) $\quad \begin{cases} i(K) = I \\ j(K) = J \end{cases}$

K désignant le nombre de coïncidences effectuées par le chemin de recalage.

V.5.- PRINCIPE DE LA PROGRAMMATION DYNAMIQUE APPLIQUEE A LA RECHERCHE DE LA FONCTION DE RECALAGE OPTIMALE.

Le recalage temporel, on l'a vu, a pour but de déterminer la fonction de recalage W qui maximise les meilleures coïncidences entre les deux formes à comparer ou encore comme nous l'avions esquissé qui minimise une certaine métrique. La fonctionnelle que l'on associe habituellement à une fonction de recalage W est donnée par la relation :

$$(5.5) \qquad D_N(W) = D_N\left(\left\{ i(k), j(k) \right\}\right) = \frac{\sum_{k=1}^{K} d\Big(i(k), j(k)\Big).P(k)}{N(P)}$$

où : - K est le nombre de points du chemin de recalage ou encore le nombre d'arcs élémentaires qui le constituent - le premier arc étant un arc fictif joignant le point (0,0) et le point (1,1) -.

- d(i,j) est la distance locale entre le $i^{ème}$ vecteur de la forme T et le $j^{ème}$ vecteur de la forme R . La distance de Hamming - somme de valeurs absolues des différences des composantes des vecteurs - - ou la distance euclidienne, où la distance d'Itakura pour les coefficients de LPC sont les distances qui sont le plus couramment utilisées.

- P(k) est une pondération qui diffère suivant la transition locale $(i(k-1), j(k-1)) \longrightarrow (i(k), j(k))$

- N(P) est un facteur de normalisation dont le rôle est de rendre $D_N(W)$ indépendant de la longueur du chemin de recalage.

La fonction de recalage W qui nous intéresse est celle qui minimise $D_N(W)$. La solution du recalage temporel est donc donnée par la relation suivante :

$$(5.6) \qquad W = \underset{W}{\text{Arg min}} \; D_N(W) \quad .$$

En fait pour effectuer une reconnaissance, W ne nous intéresse pas directement. L'information importante est le taux de dissemblance D(T,R) donné par la relation

$$(5.7) \qquad D(T,R) = D_W = D_N(W) \quad .$$

Notre problème consiste donc à évaluer :

$$(5.8) \qquad D(T,R) = \min_{K,i(k),j(k)} \left[D_N \left(\left\{ i(k), j(k) \right\} \right) \right] \quad .$$

ou encore de part la définition de $D_N(W)$:

$$(5.9) \qquad D(T,R) = \min_{K,i(k),j(k)} \frac{\sum_{k=1}^{K} d\left(i(k), j(k)\right) . P(k)}{N(P)}$$

Les fonctions typiques de pondération qui sont considérées habituellement sont les suivantes :

$$(5.10) \qquad P_a(k) = i(k) - i(k-1)$$

et

$$(5.11) \qquad P_s(k) = i(k) - i(k-1) + j(k) - j(k-1) \quad .$$

La première est dite asymétrique car elle ne fait intervenir que les indices de prélèvements de la forme projetée sur l'axe horizontal. La deuxième quant à elle, est dite symétrique car les indices de prélèvements des deux formes interviennent de façon symétrique. Du fait de l'existence de ces deux types de pondération les contraintes locales peuvent être divisées en deux groupes : les contraintes asymétriques pondérés par P_a et les contraintes symétriques pondérés par P_s . Les figures 22 et 23 mettent en évidence la contrainte de Sakoe et Chiba sous sa forme asymétrique et sous sa forme symétrique.

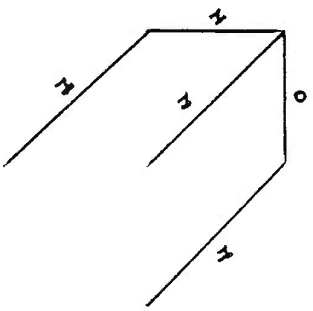


Figure 22 :
CONTRAINTE DE SAKOE ET CHIBA ASYMETRIQUE.

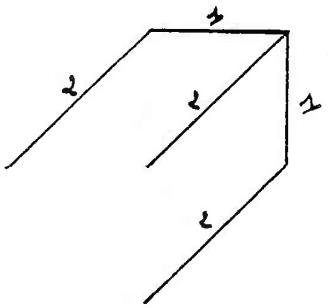


Figure 23 :
CONTRAINTE DE SAKOE ET CHIBA SYMETRIQUE.

La pondération asymétrique de la contrainte de Sakoe et Chiba illustrée par la figure 23 est essentiellement théorique. En pratique, pour ne pas avoir un arc de poids nul, la pondération des arcs de droite de la contrainte est légèrement modifiée comme l'indique la figure 24.

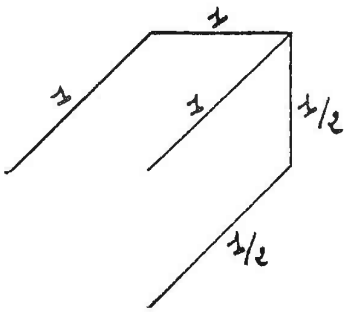


Figure 24 : Pondération asymétrique usuellement adoptée de la contrainte de Sakoe et Chiba.

Si l'on associe à tout arc élémentaire $\left\{ \left(i(k-1), j(k-1) \right) , \left(i(k), j(k) \right) \right\}$ d'un chemin de recalage W une longueur égale à $P(k)$ - pondération de l'arc de la contrainte locale qui permet de transiter de $\left(i(k-1), j(k-1) \right)$ à $\left(i(k), j(k) \right)$ - il vient que la longueur du chemin W est :

$$L(W) = \sum_{k=1}^{K} P(k) \qquad . \tag{5.12}$$

Or on a vu précédemment que $N(P)$ est un facteur de normalisation dont le but est de rendre $D_N(W)$, la métrique associée à W , indépendante de la longueur du chemin de recalage. Il est donc judicieux de prendre :

$$N(P) = \sum_{k=1}^{K} P(k) \qquad . \tag{5.13}$$

En faisant l'hypothèse que tous les chemins de recalage sont issus du même point à savoir le point $(1,1)$ ou bien ce qui revient au même du point fictif $(0,0)$ il vient que :

$$\begin{cases} i(0) = 0 \\ \text{et} \\ j(0) = 0 \end{cases} \tag{5.14}$$

Grâce aux relations (5.14) les facteurs de normalisation associés à P_a et à P_s peuvent être évalués aisément.

En effet grâce à (5.13), (5.14), (5.10) et (5.11) il vient :

$$(5.15) \qquad N(P_a) = \sum_{k=1}^{K} \Big(i(k) - i(k-1) \Big) = i(K) - i(0) = I$$

et

$$(5.16) \qquad N(P_s) = \sum_{k=1}^{K} \Big(i(k) - i(k-1) + j(k) - j(k-1) \Big)$$

$$= i(K) - i(0) + j(K) - j(0)$$

$$= I + J \qquad .$$

Des relations (5.15) et (5.16) on déduit une propriété fort importante des pondérations P_a et P_s :

si la fonction de pondération est soit P_a , soit P_s , et si les points terminaux des chemins de recalage sont des points fixes, alors pour toute fonction de recalage W , la longueur du chemin W associé à celle-ci est une constante. Autrement dit on a :

$$(5.17) \qquad \forall W \quad L(W) = \text{constante} \qquad .$$

Si la pondération est P_a , la constante est égale à I .

Si la pondération est P_s , la constante est égale à $I+J$.

Il est à noter ici que si les contraintes aux frontières sur les chemins de recalage, explicitées par les relations (5.3) et (5.4), n'étaient pas imposées alors la relation (5.17) ne serait pas vérifiée et par conséquent le facteur de normalisation serait dépendant de la longueur du chemin de recalage.

Compte-tenu de la relation (5.17) on peut simplifier la relation (5.9) :

$$(5.18)\quad D(T,R) = \frac{1}{N(P)} \underset{K,i(k),j(k)}{\text{Min}} \left[\sum_{k=1}^{K} d\left(i(k),j(k)\right) .P(k) \right]$$

Si l'on pose :

$$(5.19)\quad D_p\left[\left\{ \left(i(k),j(k) \right) \right\} \right] = \underset{K,i(k),j(k)}{\text{Min}} \left[\sum_{k=1}^{K} d\left(i(k),j(k)\right) .P(k) \right]$$

le taux de dissemblace s'écrit :

$$(5.20)\quad D(T,R) = \frac{1}{N(P)} \cdot D_p\left[\left\{ \left(i(k),j(k) \right) \right\} \right] .$$

L'équation (5.20) peut être résolue par la programmation dynamique à l'aide du principe d'optimalité local introduit par Bellman [Bell, 57]

Soit $C_{(1,1)}^{(i,j)}$ le chemin optimal joignant le points (1,1) et (i,j) , alors pour tout point $(i', j') \in C_{(1,1)}^{(i,j)}$, le chemin de recalage $C_{(1,1)}^{(i',j')}$ est optimal.

Désignons la distance cumulée optimale au point (i,j) associée à $C_{(1,1)}^{(i,j)}$ par $D_{pp}(i,j)$, soit :

$$(5.21)\quad D_{pp}(i,j) = \underset{K',i(k),j(k)}{\text{Min}} \sum_{k=1}^{K'} d\left(i(k),j(k)\right) . P(k)$$

- avec $i(1) = 1$, $j(1) = 1$ et $i(K') = i$, $j(k') = j$ -

D'après le principe d'optimalité local il vient :

$$(5.22) \qquad D_{pp}(i,j) = \min_{(i',j')} \left[D_{pp}(i',j') + d_p((i',j'),(i,j)) \right]$$

où : - (i',j') appartient à un voisinage de (i,j) défini par la contrainte locale utilisée.

- $d_p((i',j'), (i,j))$ est la distance locale pondérée entre les points (i',j') et (i,j) .

Si la contrainte considérée est par exemple la contrainte locale de Sakoe et Chiba, le voisinage du point (i,j) est constitué par les points (i',j') appartenant à l'ensemble des points $\{(i-2, j-1), (i-1, j-1) , (i-1, j-2)\}$ et, les distances pondérées entre les points (i', j') et le point (i,j) s'écrivent :

$$(5.22.a) \qquad d_p((i-2, j-1), (i,j)) = 2 * d(i-1, j) + d(i,j)$$

$$(5.22.b) \qquad d_p((i-1, j-1), (i,j)) = 2 * d(i,j)$$

$$(5.22.c) \qquad d_p((i-1, j-2), (i,j)) = 2 * d(i, j-1) + d(i,j)$$

L'équation (5.21) s'exprime donc dans le cas de la contrainte de Sakoe et Chiba par les équations récursives suivantes :

$$(5.23) \quad D_{pp}(i,j) = \text{Min} \begin{cases} D_{pp}(i-2,j-1) + 2 \cdot d(i-1,j) + d(i,j) \\ D_{pp}(i-1,j-1) + 2 \cdot d(i,j) \\ D_{pp}(i-2,j-1) + 2 \cdot d(i,j-1) + d(i,j) \end{cases}$$

qui constituent les relations traditionnelles de programmation dynamique. Grâce à celles-ci, il est possible d'évaluer $D_{pp}(i,j)$ en tout point du plan (i,j) et par conséquent la solution de l'équation (5.20) donnant le taux de dissemblance entre les formes T et R est fournie par l'équation suivante :

$$(5.24) \qquad D(T,R) = \frac{1}{N(P)} \cdot D_{pp}(I,J) \quad .$$

Grâce aux relations (5.23) et (5.24) il est aisé de déduire un algorithme permettant d'évaluer $D(T,R)$:

Algorithme 1 :

1) Initialisation :

$$D_{pp}(1,1) = d(1,1).P(1)$$

2) Programmation dynamique

évaluer $D_{pp}(i,j)$ pour $1 \leqslant i \leqslant I$ et $1 \leqslant j \leqslant J$

3) Détermination du taux de dissemblance :

$$D(T,R) = \frac{1}{N(P)} \cdot D_{pp}(I,J) \quad .$$

V.6.- LIMITATION DU DOMAINE DE RECHERCHE DU CHEMIN DE RECALAGE OPTIMAL.

L'algorithme de programmation dynamique général que nous venons d'expliciter met en évidence le fait qu'il faut en tout point (i,j) du plan de comparaison déterminer une distance cumulée partielle optimale $D_{pp}(i,j)$. Celle-ci d'après la relation (5.23) nécessite l'évaluation d'une distance locale $d(i,j)$ entre le $i^{ème}$ prélèvement de la forme T et le $j^{ème}$ prélèvement de la forme R . Cette dernière opération qui est effectuée sur des données multidimentionnelles est très couteuse en temps de calcul. Ainsi est-il intéressant de limiter la zone de recherche du chemin de recalage optimal et par conséquent le nombre de points où il faut évaluer une distance locale. Myers [MYER, 1980] a montré qu'en fonction de la contrainte locale à laquelle est assujettie la fonction de recalage il est possible de définir une zone en dehors de laquelle il est inutile de rechercher

le chemin optimal. En désignant par EMAX et EMIN respectivement la pente du chemin local de pente maximale et la pente du chemin local de pente minimale de la contrainte utilisée - par exemple dans le cas de la contrainte de Sakoe et Chiba EMAX = 2 et EMIN = 1/2 - le domaine de recherche du chemin de recalage est défini par les relations suivantes :

$$(5.25.a) \qquad 1 + EMIN\,[i(k) - 1] \leqslant j(k) \leqslant 1 + EMAX\,[i(k) - 1]$$

$$(5.25.b) \qquad J + EMAX\,[i(k) - I] \leqslant j(k) \leqslant J + EMIN\,[(i(k) - I] \quad .$$

La figure 25 visualise la zone dans le plan de comparaison en dehors de laquelle il est inutile de rechercher le chemin optimal de recalage dans le cas de la contrainte de Sakoe et Chiba.

Sakoe et Chiba, toujours dans l'optique de limiter le nombre de distances locales à évaluer ont proposé la contrainte suivante [SAKO, 1978] :

$$(5.26) \qquad |i(k) - j(k)| \leqslant R$$

qui se traduit physiquement par le fait que le $i(k)^{ème}$ prélèvement de la forme T ne peut être en retard ou en avance de plus de R prélèvements par rapport au $j(k)^{ème}$ prélèvement de la forme R . La figure 26 visualise le domaine de recherche de Sakoe et Chiba.

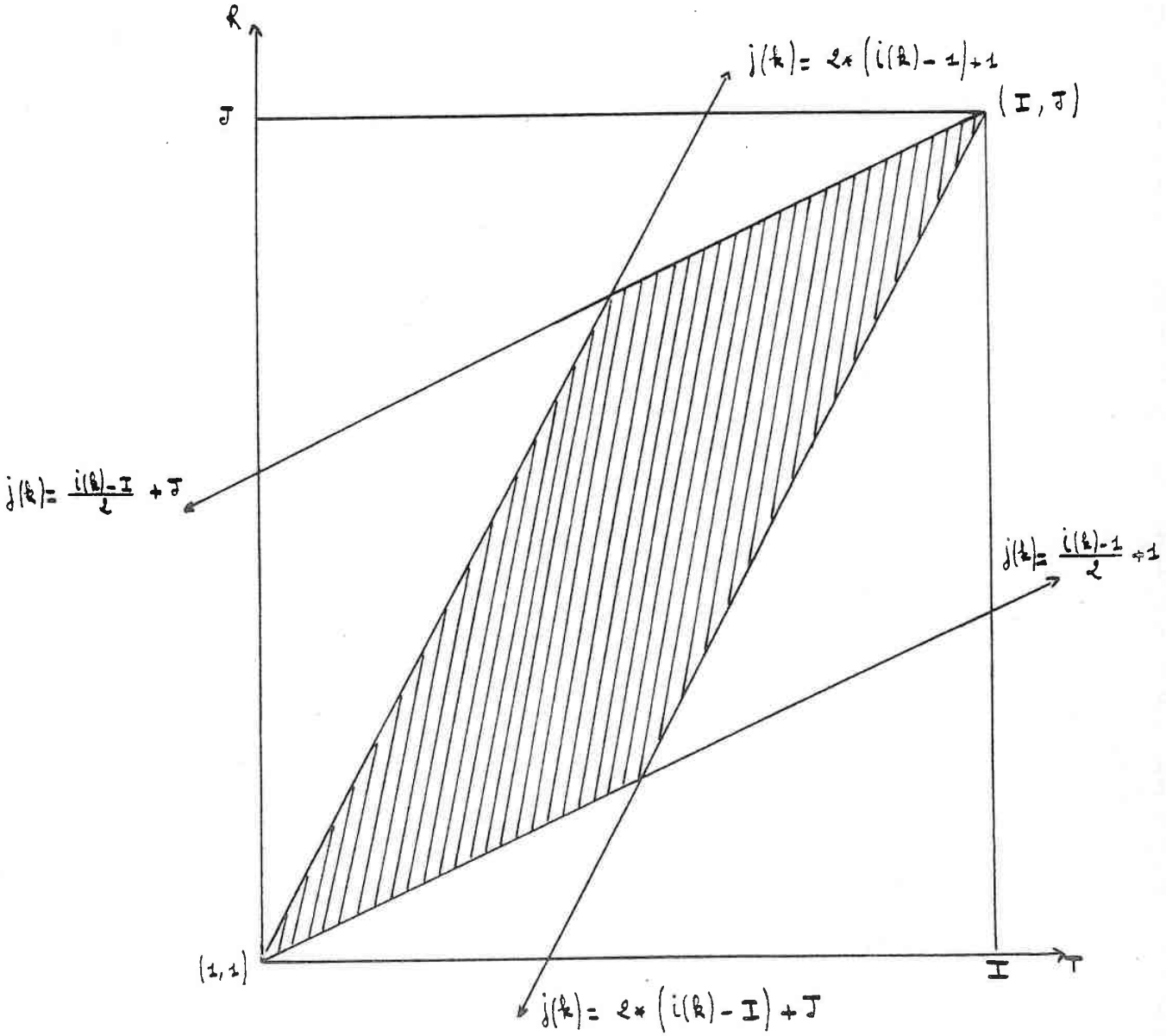


Figure 25 : La zone hachurée du plan de comparaison définie le domaine de recherche du chemin de recalage optimal proposé par Myers [MYER, 1980]

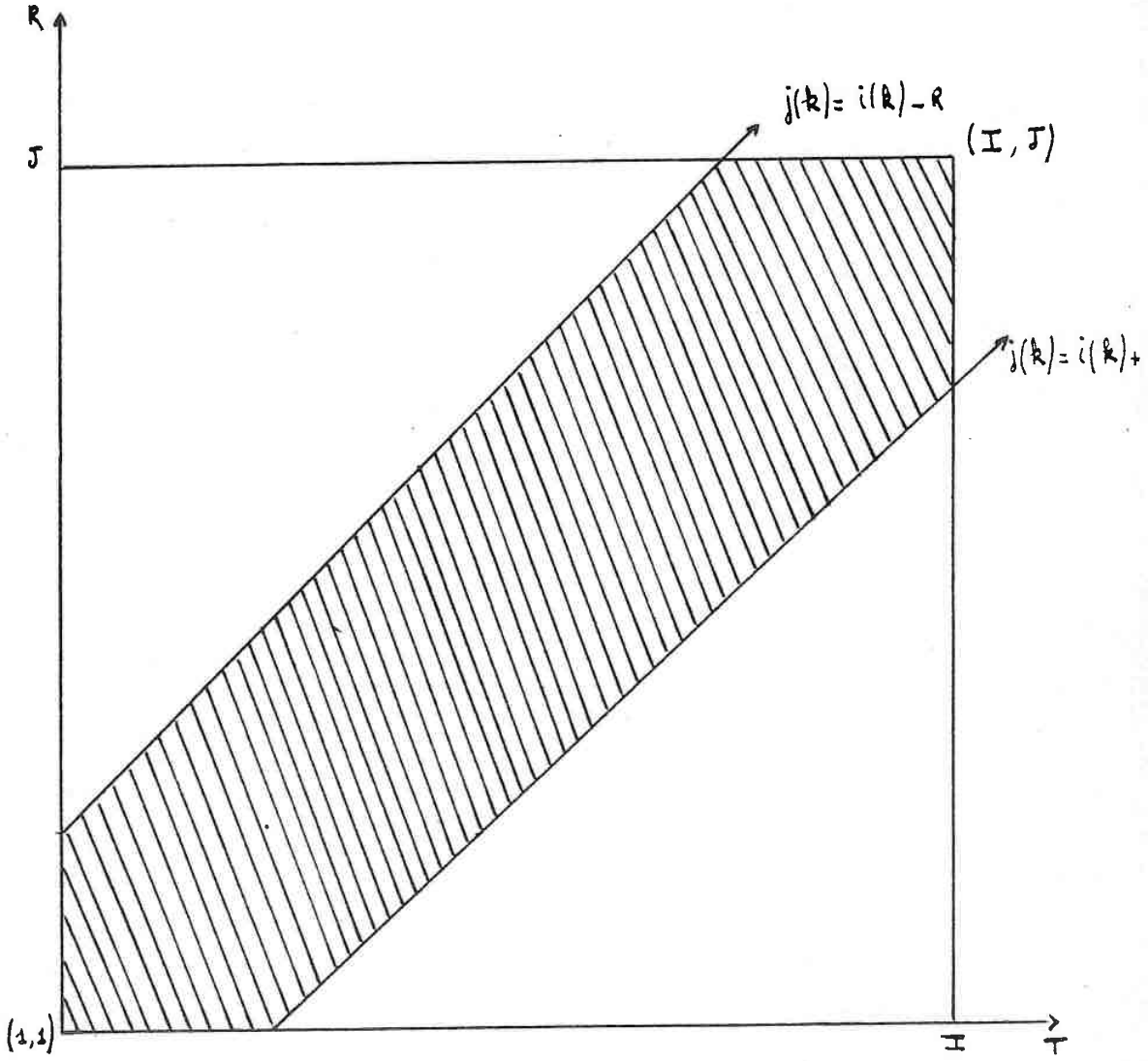


Figure 26 : La zone hachurée du plan de comparaison est le domaine de recherche du chemin optimal proposé par Sakoe et Chiba [SAKO, 1978] .

Les nouvelles contraintes que nous venons de définir sur les chemins de recalage sont désignées couramment par contraintes globales. Elles limitent considérablement le nombre de calcul de distances locales tout en préservant les performances des algorithmes.

V.7.- INFLUENCE DES CONTRAINTES LOCALES ASSUJETTISSANT LES CHEMINS DE RECALAGE SUR LA PERFORMANCE DES ALGORITHMES DE PROGRAMMATION DYNAMIQUE.

On a vue au paragraphe V.4.3. que les chemins de recalage, pour respecter l'évolution dans le temps du signal vocal et aussi pour ne pas réaliser des compensations de distorsions temporelles irréalistes, doivent être soumis à des contraintes locales qui définissent parmi l'ensemble des chemins de recalage possibles une classe de chemins dans laquelle sera recherché le chemin optimal. Les contraintes locales qui sont considérées le plus souvent dans la littérature apparaissent à la figure 27.

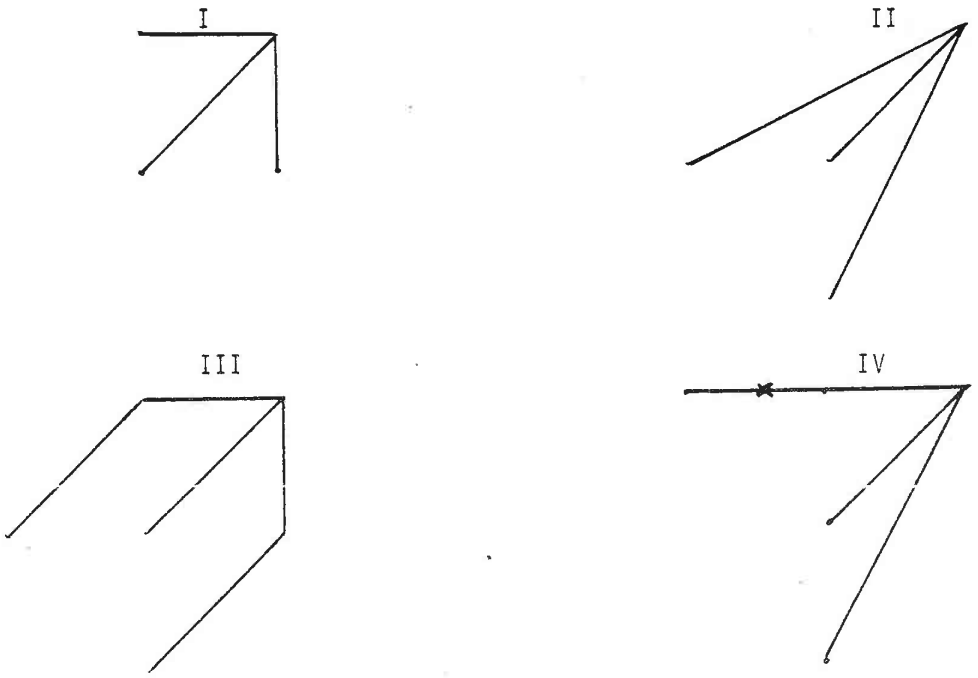


Figure 27 : Les contraintes locales les plus couramment adoptées dans les algorithmes de programmation dynamique.

La contrainte I est celle qui définit les critères de monotonie des chemins de recalage. C'est la plus simple des contraintes locales. Elle n'impose aucune condition sur la pente du chemin de recalage. Tous les chemins possibles donc sont pris en considération par cette contrainte locale.

La contrainte II diffère sensiblement de la contrainte I, d'une part, parce qu'elle impose que la pente globale des chemins de recalage soit comprise entre +2 et 1/2 - une telle contrainte, par conséquent, ne pourra effectuer que des compressions ou des dilatations de taux inférieur à 2 - et, d'autre part, car comme on peut le voir d'après son motif, elle autorise des omissions de prélèvements sur chacune des deux formes.

La contrainte III est la contrainte de Sakoe et Chiba que nous avons considérée précédemment à plusieurs reprises. Elle interdit deux déplacements consécutifs dans la même direction lorsque celle-ci est horizontale ou verticale. La pente globale des chemins de recalage est donc forcée à être comprise entre 2 et 1/2. Comme pour la contrainte II, la contrainte de Sakoe et Chiba ne pourra effectuer que des compressions ou des dilatations de taux inférieur à 2.

La containte IV proposée par Itakura interdit tout déplacement vertical ainsi que deux déplacements consécutifs dans la direction horizontale. Le taux de compression ou de dilatation maximal autorisé par cette contrainte est identique à celui des contraintes II et III.

Afin d'étudier l'influence de ces quatre contraintes sur la reconnaissance de chiffres, nous avons analysé pour de nombreuses comparaisons l'évolution des distances cumulées établie par différents algorithmes de programmation dynamique implémentant chacune des contraintes précédemment citées. Cette étude nous a permis de mieux appréhender les avantages et les inconvénients de ces contraintes.

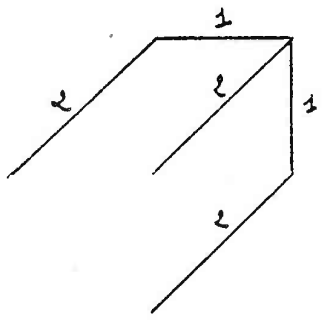
La contrainte I est celle qui a donné les moins bons résultats. Nous avons noté avec celle-ci de nombreuses ambiguités en particulier entre les mots "deux" et "neuf", "un" et "deux", "cinq" et "sept".

Nous avons pu mettre en évidence que ces confusions étaient dues pour une bonne part à la complète liberté sur la pente du chemin de recalage que permet cette contrainte. Celle-ci autorise en effet des compressions ou des dilatations très importantes et donc en pratique irréalistes. Ainsi en prenant le cas de la confusion anormale entre les mots "un" et "deux" il se produit systématiquement le phénomène suivant :
la plosive voisée /d/ est compressée au maximum et est mise en correspondance avec le début du mot "un" ; les voyelles des deux mots présentant des similitudes sont quant à elles mises en correspondance de façon correcte. Une telle contrainte locale peut donc fournir de bons taux de dissemblance lors de la comparaison de formes vocales ne présentant à priori que peu de similitude.

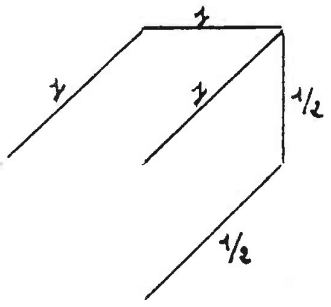
La contrainte II nous a permis d'obtenir des résultats légèrement meilleurs sans toutefois supprimer toutes les ambiguités anormales. Ici, la raison de cet état de fait est plus subtile car cette contrainte limite les variations de pente des chemins de recalage et donc exclut les compensations irréalistes. Les confusions que nous avons observées avec cette contrainte sont dues principalement au fait que celle-ci a la possibilité d'ignorer certains prélèvements sur chacune des deux formes comparées. Ainsi une transition contenant beaucoup d'information peut ne pas être prise en compte lors de la comparaison. DAS, [DAS, 1982] dans un article récent a mis en évidence ce phénomène pour la contrainte d'Itakura qui, comme la contrainte II, a la possibilité d'ignorer certains prélèvements de la forme projetée sur l'axe vectical.

La contrainte III, c'est-à-dire la contrainte de Sakoe et Chiba a fourni les meilleurs résultats. Le motif de cette contrainte est certainement le plus complet en ce sens qu'aucun prélèvement de chacune des deux formes n'est ignoré et qu'elle limite les variations de pente des chemins de recalage. Nous avons pourtant décelé avec celle-ci une faiblesse due à la présence d'une dissymétrie dans la pondération des

arcs de la contrainte. En effet dans le cas d'une pondération symétrique le motif de Sakoe et Chiba est le suivant :



On peut observer que les arcs horizontaux et verticaux présentent des poids moindres que ceux des arcs obliques. Ceci est un inconvénient car avec une telle pondération l'algorithme de programmation dynamique va avoir tendance à "lisser" les transitions en ne les franchissant qu'horizontalement ou verticalement. Dans le cas d'une pondération asymétrique le motif est le suivant :



Là encore on observe une dissymétrie dans la pondération : le chemin local de compression est pondéré deux fois moins que les autres d'un facteur 2. La contrainte de Sakoe et Chiba asymétrique a donc plutôt tendance à comprimer la forme projetée sur l'axe vertical.

La contrainte IV, c'est-à-dire la contrainte d'Itakura a fourni des résultats sensiblement identiques à ceux obtenus avec la contrainte de Sakoe et Chiba. Toutefois nous avons mis en évidence deux inconvénients relatifs à celle-ci qui ont dans certains cas perturbés la reconnaissance. Le premier est dû au fait qu'un chemin local n'est pas pris en considération en un point (i,j) du plan lorsqu'un chemin de recalage aboutit au point (i-1, j) par un déplacement horizontal.

La figure 28 visualise le chemin local non considéré par la contrainte d'Itakura :

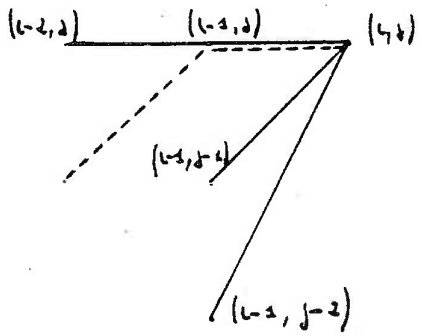


Figure 28 : Le chemin en pointillé se terminant au point (i,j) est le chemin non pris en considération par la contrainte d'Itakura lorsqu'un chemin de recalage aboutit du point (i-1,j) par un déplacement horizontal c'est-à-dire provenant du point (i-2,j).

Le deuxième point faible que nous avons noté provient du fait, comme pour la contrainte II, que certains prélèvements de la forme projetée sur l'axe vertical peuvent être ignorés.

Ces différentes considérations sur les contraintes locales assujetissant les chemins de recalage nous ont permis d'imaginer une contrainte

qui ne présente pas les différents défauts que nous avons mis en évidence expérimentalement pour les quatre contraintes considérées précédemment. Le motif de celle-ci est visualisé par la figure 29.

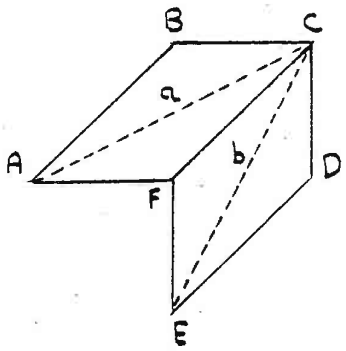


Figure 29 : Motif de la contrainte que nous proposons.

Les chemins en pointillés a et b peuvent être considérés comme des chemins virtuels qui peuvent être concrétisés respectivement par les chemins locaux $\overset{\frown}{ABC}$ ou $\overset{\frown}{AFC}$ et les chemins $\overset{\frown}{EFC}$ ou $\overset{\frown}{EDC}$. Pour le chemin virtuel a un choix est fait entre le chemin $\overset{\frown}{ABC}$ et le chemin $\overset{\frown}{AFC}$. Celui qui est opté est celui qui présente la plus petite distance locale aux points B et F . De même pour le chemin virtuel b le choix entre le chemin $\overset{\frown}{EFC}$ et $\overset{\frown}{EDC}$ est réalisé en prenant celui qui présente la plus petite distance locale aux points F et D . La distance locale associée finalement au chemin virtuel a est la plus grande des distances entre la distance locale au point C et la plus petite des distances aux points B et F . De même, la distance associée du chemin virtuel b est la plus grande des distances entre la distance locale au point C et la plus petite des distances aux

points F et D . Les relations récursives pour cette contrainte dans le cas d'une pondération symétrique s'écrivent donc :

$$(5.27)\quad D_{PP}(i,j) = \text{Min} \begin{cases} D_{PP}(i-2,j-1) + 3*\text{MAX}(d(i,j),\text{Min}(d(i-1,j), \\ \qquad\qquad d(i-1,j-1))) \\ D_{PP}(i-1,j-1) + 2*d(i,j) \\ D_{PP}(i-1,j-2) + 3*\text{MAX}(d(i,j), \text{Min}(d(i-1,j-1), \\ \qquad\qquad d(i,j-1))) \end{cases} .$$

Nous avons essayé expérimentalement cette contrainte avec un algorithme de reconnaissance de mots enchaînés où la détection de transitions rapides est primordiale. Celle-ci nous a permis d'obtenir des résultats tout à fait satisfaisants et nous avons pu mettre en évidence le fait que les frontières de mots étaient mieux détectées avec la contrainte que nous proposons, plutôt qu'avec les contraintes d'Itakura ou de Sakoe et Chiba.

V.8.- LES ALGORITHMES A OPTIMUMS LOCAUX.

L'algorithme de programmation dynamique général que nous avons explicité au paragraphe V.5 se propose de déterminer parmi l'ensemble de tous les chemins de recalage définis par la contrainte locale utilisée, le chemin optimal. Pour ce faire un grand nombre de distances locales d(i,j) doivent être évaluées dans le plan de comparaison. Avec les algorithmes à optimums locaux le principe d'optimalité global du problème est abandonné afin d'essayer de minimiser considérablement le nombre de distances locales à calculer.

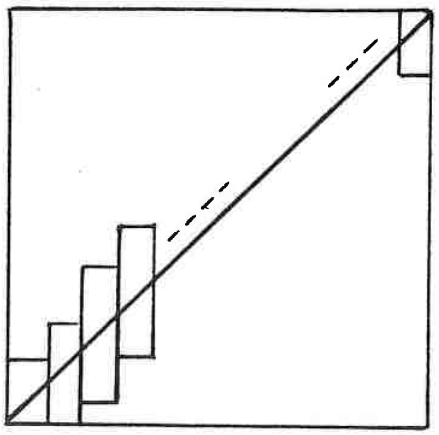
La fonctionnelle qui est associée aux algorithmes à optimums locaux est la somme des minimums locaux dans des pavés du plan de comparaison prédéfinis ou bien dont la position est évolutive, normalisée par le nombre de pavés considérés. La figure 30 montre les différents types de pavés qui sont utilisés habituellement.

Si l'on désigne par (i_{k_n}, j_{k_n}) un point du $k^{ième}$ pavé et par NP le nombre de pavés considérés lors de la comparaison entre deux formes T et R il vient donc que les algorithmes à optimums locaux évaluant le taux de dissemblance D(T,R) de la manière suivante :

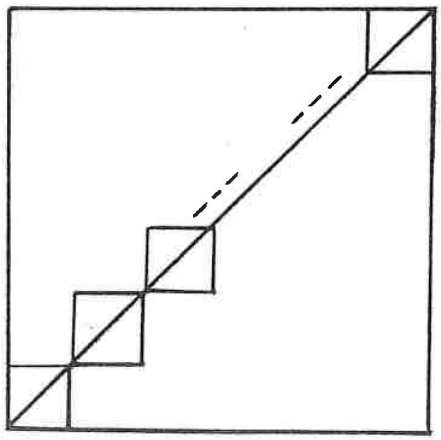
$$D(T,R) = \frac{1}{NP} \left\{ \sum_{k=1}^{NP} \underset{n}{\text{Min}}\ d\left(i_{k_n}, j_{k_n}\right) \right\} \tag{5.28}$$

D'après (5.28) il est clair que plus la taille du pavé est faible et plus le nombre de distances locales à évaluer est diminué. On voit donc qu'il faut dans un algorithme à optimums locaux effectuer un compromis sur ce paramètre afin de limiter au maximum le temps de calcul sans trop dégrader les performances de l'algorithme.

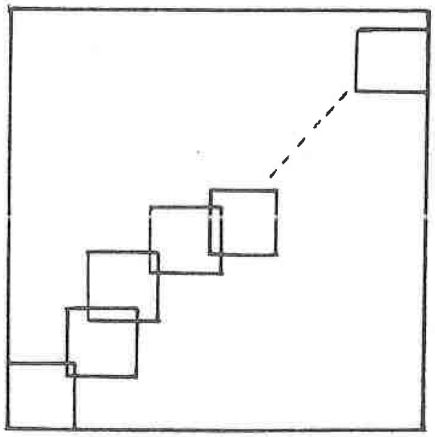
Les algorithmes à optimums locaux peuvent gagner par rapport à un algorithme de programmation dynamique classique jusqu'à un rapport dix en temps de calcul tout en conservant des taux de reconnaissance honorables. Toutefois ces algorithmes sont en général peu robustes en ce sens qu'un locuteur peu entraîné aux systèmes de reconnaissance vocaux, mettra souvent en défaut ces derniers. En effet, ce type de locuteur aura tendance à engendrer des distorsions temporelles qui nécessitent des chemins de recalage s'écartant sensiblement de la diagonale du plan de comparaison et donc ne pouvant être compensées par un algorithme à optimums locaux dont la stratégie est d'évaluer le score d'un chemin sous-optimal appartenant à un voisinage généralement proche de la diagonale.



a) Pavés unicolonnes centrés sur la diagonale.



b) Pavés matriciels centrés sur la diagonale



c) Pavés matriciels évolutifs : le sommet inférieur gauche du pavé i est positionné sur le minimum trouvé au pavé i-1. D'après HATON [HATO,7

Figure 30 : Différents types de pavés utilisés dans les algorithmes à optimums locaux.

V.9.- DESCRIPTION DE DEUX ALGORITHMES A OPTIMUMS GLOBAUX AUTORISANT UNE RELAXATION DES CONTRAINTES AUX FRONTIERES.

V.9.1.- Introduction.

Afin de pouvoir compenser les distorsions pouvant affecter les débuts ainsi que les fins de mots il serait souhaitable que les algorithmes de programmation dynamique puissent permettre aux chemins de recalage de débuter dans un voisinage du point (1,1) et se terminer dans un voisinage du point (I,J) . Autrement dit il faudrait que les contraintes aux frontières puissent être relachées. L'approche des deux algorithmes que nous allons expliciter est originale, précisément parce qu'elle autorise une relaxation des contraintes aux frontières. Une autre particularité attrayante de ces algorithmes est due au fait qu'ils ne figent pas statiquement le domaine légal de recherche du chemin optimal de recalage.

V.9.2.- L'U.E.L.M. : Unconstrained Endpoints Local Minimum [Rabi, 1978] .

V.9.2.1.- Le relachement des contraintes aux frontières effectué par l'U.E.L.M.

L'U.E.L.M. proposé en 1978 par Rabiner, Rosenberg et Levinson autorise une relaxation des contraintes initiales sur l'axe vertical et des contraintes terminales sur les deux axes en adoptant une contrainte locale asymétrique.

La figure 31 montre un chemin de recalage pouvant être considéré par l'U.E.L.M.

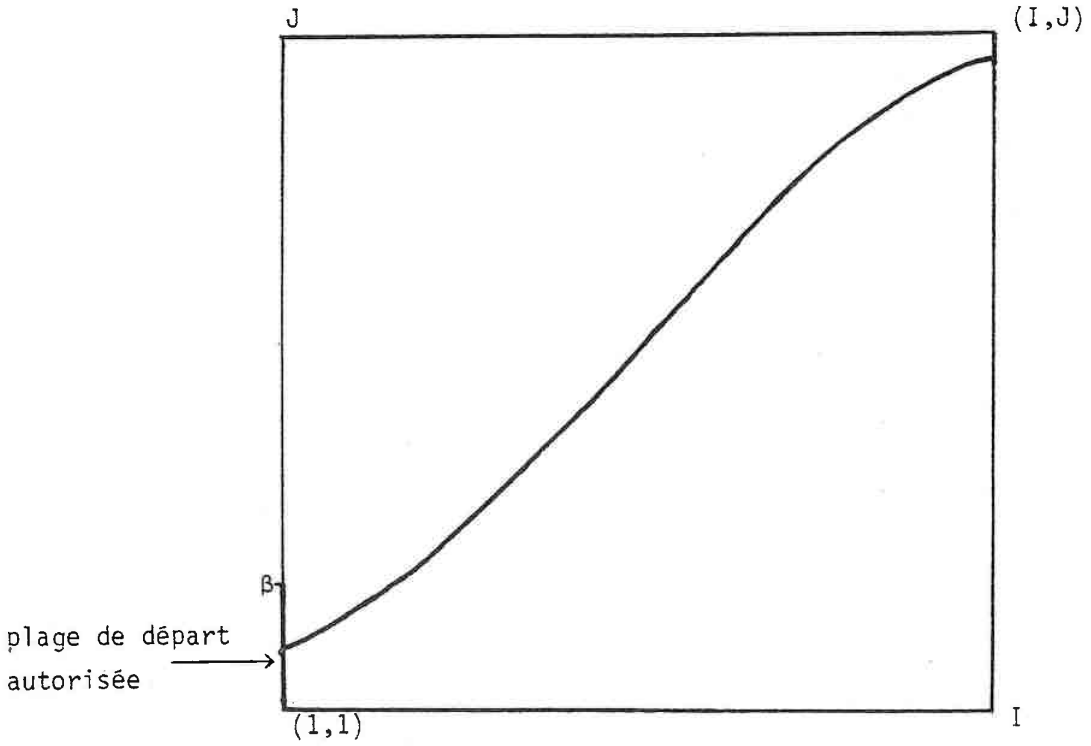


Figure 31 : Relachement des contraintes initiales par l'U.E.L.M.

Le relachement des contraintes aux frontières, comme on le verra plus en détail lors de la description de C.R.S.L.S., l'algorithme que nous proposons, n'est en effet permis avec les relations classiques de programmation dynamique qu'avec une contrainte locale asymétrique. Pour comprendre cela il faut se rappeler que les relations récursives de comparaison dynamique ont été déduites en particulier grâce à la propriété de constance du facteur de normalisation $N(P)$.

Or si les chemins de recalage ne sont pas contraints à débuter et à se terminer en des points fixes du plan de comparaison, cette propriété peut ne pas être vérifiée. Dans le cas où la contrainte est du type asymétrique et où le relachement des contraintes initiales s'effectue seulement sur l'axe vertical, la relation (5.15) donnant la valeur du facteur de normalisation pour une pondération asymétrique reste applicable et par conséquent la nouvelle approche proposée par l'U.E.L.M. est théoriquement valable.

V.9.2.2.- La contrainte globale dynamique de l'U.E.L.M.

Les différentes contraintes que l'on a explicitées précédemment ont la particularité d'être essentiellement statiques : elles prédéfinissent une zone du plan de comparaison dans laquelle est recherché le chemin optimal. Cette zone d'investigation doit être prévue suffisamment importante afin qu'il soit hautement improbable qu'un chemin optimal de recalage, lors d'une comparaison, déborde du domaine légal de recherche. La stratégie qui est utilisée dans l'U.E.L.M. pour diminuer le nombre de distances locales est sensiblement différente et rejoint celle proposée par Haton [HATO, 1974] dans son algorithme à optimums locaux utilisant des pavés matriciels dont l'emplacement évolue au cours de la comparaison - la position du pavé i est déterminée en fonction de la position du pavé i-1 -. En effet la méthode consiste à déterminer à chaque étape de la comparaison, c'est-à-dire pour chaque prélèvement i de la forme projetée sur l'axe horizontal la fenêtre susceptible de contenir un point du chemin optimal de recalage en tenant compte de l'emplacement de la fenêtre établie à l'étape précédente. Pour être plus précis, si l'on désigne par P(i) la position sur l'axe vertical du minimum des distances cumulées $D_{pp}(i-1, j)$ évaluées à l'étape i-1 dans la fenêtre d'explo-

ration correspondante, l'U.E.L.M. évalue alors les distances $D_{pp}(i,j)$ à l'étape i dans une fenêtre centrée autour de $P(i)$ c'est-à-dire pour j vérifiant l'inégalité (5.29).

(5.29) $$MAX(1, P(i)-\varepsilon) \leqslant j \leqslant MIN(P(i)+\varepsilon , j) \quad .$$

La figure 32 visualise une contrainte globale typique utilisée par l'U.E.L.M..

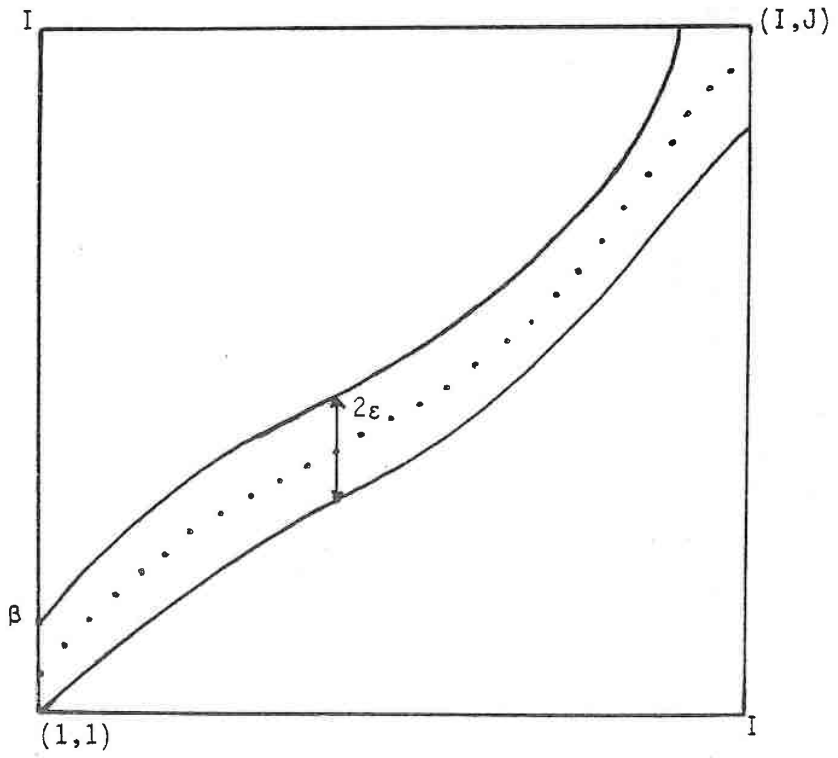


Figure 32 : La contrainte globale dynamique de l'U.E.L.M.

Avec ce type de contrainte globale une difficulté peut survenir si le chemin de recalage aboutit à l'ordonnée J avant d'arriver au bout de la forme située sur l'axe horizontal, c'est-à-dire à l'abscisse I comme le montre la figure 33.

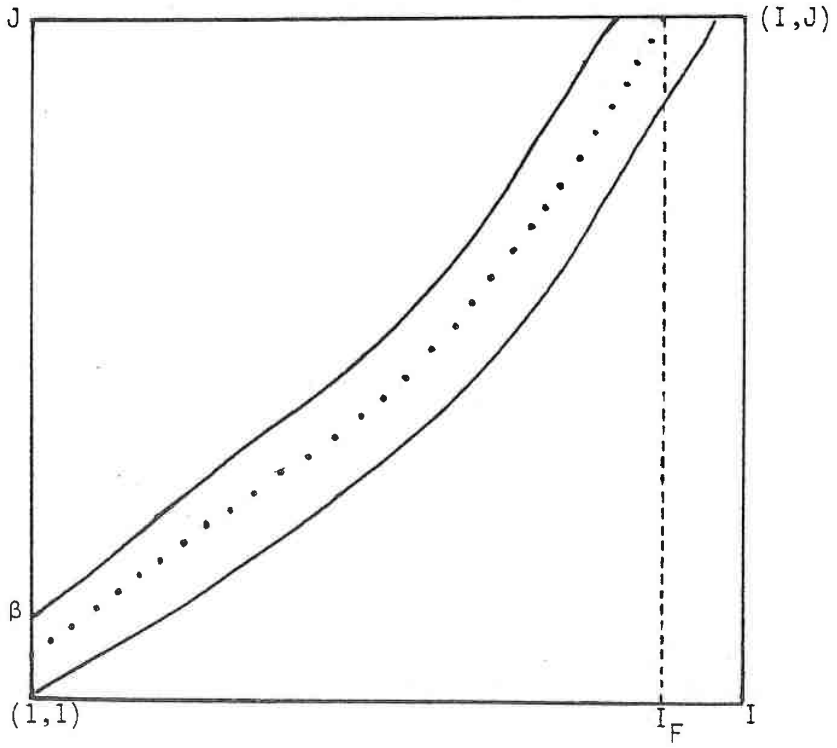


Figure 33 : un exemple de chemin de recalage se terminant en un point d'abscisse $< I$.

En effet dans ce cas le facteur de normalisation dépend de l'abscisse I_F du point terminal du chemin de recalage. Afin de rendre la distance cumulée indépendante de la longueur du chemin de recalage, celle-ci est pondérée par le facteur I/I_F .

V.9.2.3.- Spécification de l'U.E.L.M.

Algorithme 2 :

- 1°) Initialisation

. $P(1) = 1$

. $D_{pp}(1,j) = d(1,j)$ pour $1 \leqslant j \leqslant \beta$

(* β définit la plage de départ des chemins de recalage *)

. Booléen = Faux.

(* ce Booléen positionné à Vrai indique que la comparaison s'est terminée avant que le dernier prélèvement de la forme située sur l'axe horizontal n'ait été atteint *)

- 2°) Programmation dynamique

. tant que $2 \leqslant i \leqslant I$ et Booléen = Faux Faire 1 :

.. $P(i) = \underset{[MAX(1,P(i-1)-\varepsilon) \leqslant j \leqslant Min(P(i-1)+\varepsilon,J)]}{Arg\ min} D_{pp}(i-1,j)$

.. si $P(i) = J$ alors Faire Booléen = Vrai

$I_F = i-1$.

.. FIN DE SI

.. Pour $MAX(1,P(i)-\varepsilon) \leqslant j \leqslant Min(P(i)+\varepsilon,J)$ FAIRE 2

... Evaluer $D_{pp}(i,j)$ à l'aide d'une contrainte locale asymétrique.

.. Fin de Faire 2

. Fin de Faire 1, FIN DE TANT QUE.

.Si Booléen = Vrai alors $D(T,R) = \frac{I}{I_F} D_{pp}(I_F,J)$

sinon $P(I+1) = \underset{MAX(1,P(I)-\varepsilon) \leqslant j \leqslant MIN(P(I)+\varepsilon,J)}{Arg\ min} Dpp(I,j)$

$D(T,R) = D_{pp}(I,P(I+1))$.

. FIN DE SI

V.9.2.4.- Remarques concernant l'U.E.L.M.

L'U.E.L.M. ainsi présentent deux idées attrayantes : il autorise d'une part une relaxation des contraintes aux frontières et il minimise, d'autre part, considérablement le nombre de calculs de distances locales en suivant localement le chemin optimal.

La première idée est somme toute la plus intéressante. En effet, en autorisant les chemins de recalage à pouvoir débuter et se terminer autour du point (1,1) et (I,J) ,dans une certaine mesure, l'U.E.L.M. permet une certaine compensation de distorsions pouvant apparaître en début ou en fin de forme dues soit à une mauvaise segmentation parole - non parole, soit à des bruits parasites engendrés par exemple par des claquements de lèvres. Cette compensation toutefois n'est pas complète dans la mesure où elle n'opère que sur la forme se trouvant sur l'axe vertical.

La deuxième idée que nous avons évoquée permet d'obtenir des gains en temps de calcul non négligeables mais rend, en contrepartie,l'algorithme dépendant du paramètre ε -demi-largeur de la fenêtre d'exploration - qu'il faut régler convenablement. En effet, il est clair qu'il faut trouver un compromis judicieux expérimentalement pour le choix de ε : une valeur trop grande enlève tout intérêt à la méthode et une valeur trop faible risque de créer des "décrochages" dans le suivi du chemin optimal, pouvant altérer la performance de l'algorithme.

V.9.3.- C.R.S.L.S. - contraintes relachées, stratégie locale symétrique - : un algorithme de programmation dynamique autorisant une relaxation de contraintes aux frontières suivant les deux axes [DIMA, 1983].

V.9.3.1.- Introduction

Afin de pouvoir compenser les distorsions pouvant affecter les débuts ainsi que les fins de mots, il est souhaitable, on l'a vu, que les algorithmes de programmation dynamique puissent permettre aux chemins de recalage de débuter dans un voisinage du point (1,1) et se terminer dans un voisinage du point (I,J) . - Autrement dit, il faudrait que les contraintes aux frontières soient relachées suivant les deux axes du plan de comparaison. En fait ceci ne peut se faire sans que les relations récursives de programmation dynamique soient redéfinies globalement. En effet, il faut se souvenir que l'expression suivante du taux de dissemblance :

$$D(T,R) = \frac{1}{N(P)} \cdot D_{pp}(I,J) \qquad (5.30)$$

a été obtenue grâce, d'une part, aux contraintes aux frontières et aux fonctions de pondération qui ont été convenablement choisies pour que tous les chemins aboutissant en un point donné du plan de comparaison aient la même longueur et, d'autre part, au principe d'optimalité local dont l'énoncé faisait implicitement l'hypothèse que tous les chemins de recalage étaient issus du même point à savoir le point (1,1) . Dans la mesure où les contraintes aux frontières sont relachées il est manifeste que le principe d'optimalité local doit être revu afin que des relations de programmation dynamique tenant compte de l'inégalité de longueur des chemins de recalage en un point du plan de comparaison puissent être déduites.

V.9.3.2.- Nouvelle définition du principe d'optimalité local.

Avant d'énoncer le nouveau principe d'optimalité local, il est nécessaire de montrer clairement que quel que soit le type de contrainte utilisée la propriété de constance de la longueur des chemins de recalage en un point du plan de comparaison, à une exception près, n'est pas vérifiée lorsque les contraintes aux frontières sont relachées. Les figures 34 et 35 montrent les deux types de relaxation des contraintes aux frontières - sur ces figures, par souci de clarté, seules les contraintes initiales ont été considérées -.

Supposons que la contrainte locale soit symétrique. Si l'on considère les chemins de recalage de la figure 34 il vient :

(5.31) $L(C_1) = I + J - \mathfrak{V}_1 + 1$

(5.32) $L(C_2) = I + J - \mathfrak{V}_2 + 1$

(5.33) $L(C_3) = I + J - \mathfrak{V}_3 + 1$

et comme $N_1 \neq N_2 \neq N_3 \neq N_1$ il vient :

$$L(C_1) \neq L(C_2) \neq L(C_3) \neq L(C_1) \quad .$$

De même pour les chemins de la figure 35 on a :

(5.34) $L(C_1) = I + J - h_1 + 1$

(5.35) $L(C_2) = I + J - h_2 + 1$

(5.36) $L(C_3) = I + J - h_3 + 1$

et de nouveau on a, puisque $h_1 \neq h_2 \neq h_3 \neq h_1$:

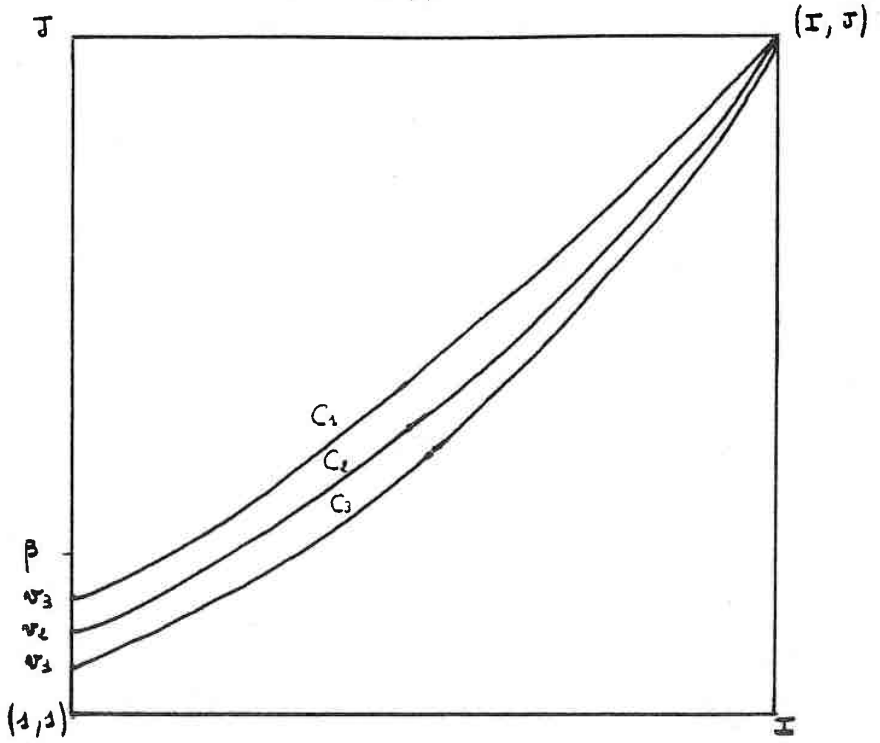


Figure 34 : Relaxation des contraintes initiales suivant l'axe vertical.

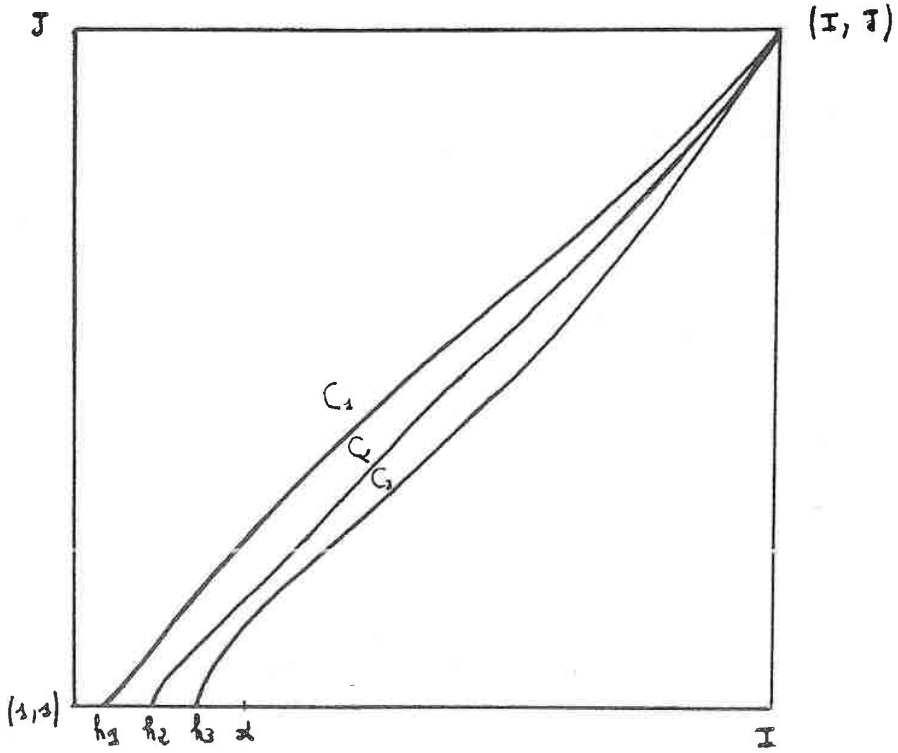


Figure 35 : Relaxation des contraintes initiales suivant l'axe horizontal.

$$L(C_1) \neq L(C_2) \neq L(3) \neq L(C_1) \quad .$$

De ceci on peut conclure que si la contrainte locale est symétrique alors quel que soit le type de relaxation de contraintes aux frontières effectuées, la propriété de constance de la longueur des chemins de recalage en un point donné du plan de comparaison n'est pas vérifiée.

Plaçons-nous maintenant dans le cas où la contrainte locale est asymétrique. Alors pour les chemins de la figure 34 il vient :

(5.37) $$L(C_1) = L(C_2) = L(C_3) = I \quad .$$

Ainsi lorsque la contrainte est asymétrique et que la relaxation s'effectue suivant l'axe vertical le facteur de normalisation est indépendant du chemin de recalage. C'est cette propriété qui a été mise à profit dans l'U.E.L.M.. En ce qui concerne les chemins de la figure 35 il vient :

(5.38) $$L(C_1) = I - h_1 + 1$$

(5.39) $$L(C_2) = I - h_2 + 1$$

(5.40) $$L(C_3) = I - h_3 + 1$$

et encore $L(C_1) \neq L(C_2) \neq L(C_3) \neq L(C_1)$.

Si la contrainte est asymétrique, donc, seule une relaxation verticale des contraintes initiales préserve la propriété de constance des longueurs des chemins de recalage.

A partir de toutes ces considérations pour tenir compte de la variabilité des longueurs des chemins de recalage en un point donné du plan de comparaison lorsque les contraintes aux frontières sont relachées, nous avons été amené à redéfinir le principe d'optimalité local de la façon suivante :

Soit $\mathcal{D}$ un sous-ensemble du plan de comparaison $\mathcal{P}$ contenant l'ensemble des points pouvant être origines d'un chemin optimal de recalage.

Soit $P : \mathcal{P} \longrightarrow \mathcal{D}$, la fonction qui a tout point du plan (i,j) associe l'origine du chemin optimal aboutissant au point (i,j) .

Soit $\hat{C}^{(i,j)}_{P[(i,j)]}$ le chemin optimal joignant les points $P(i,j)$ et (i,j) alors pour tout point $(i',j') \in C^{(i,j)}_{P[(i,j)]}$ on a les propriétés suivantes :

1)- le chemin de recalage $C^{(i',j')}_{P[(i',j')]}$ est optimal

2)- $P[(i',j')] = P[(i,j)]$.

V.9.3.3.- Généralisation des relations récursives de programmation dynamique.

Le nouveau principe d'optimalité que nous venons d'énoncer peut être considéré comme une généralisation de celui que nous avons considéré au paragraphe V.5.

Il va nous permettre de déduire une formulation plus générale des relations récursives de programmation dynamique.

Compte-tenu du nouveau principe d'optimalité local, pour calculer la distance cumulée $D_{pp}(i,j)$ au point (i,j) en fonction des

distances $D_{pp}(i',j')$ évaluées aux points (i',j') appartenant à un voisinage $V(i,j)$ du point (i,j) défini par la contrainte locale utilisée, il faut dans un premier temps déterminer quel est, parmi l'ensemble des chemins de recalage qui aboutissent au point (i,j) et qui passent par un des points (i',j'), celui qui satisfait au critère d'optimalité. Pour ce faire nous déterminons le point $(\hat{i}', \hat{j}')$ pour lequel passe le chemin optimal dont l'extrémité est le point (i,j) ainsi :

$$(5.41) \qquad (\hat{i}',\hat{j}') = \underset{\substack{(i',j') \\ \in V(i,j)}}{\text{Arg min}} \frac{D_{pp}(i',j') + d_p((i',j'), (i,j))}{L\left(C_{p[(i',j')]}^{(i,j)}\right)}$$

Connaissant, grâce à $(\hat{i}',\hat{j}')$ le chemin optimal aboutissant au point (i,j) nous évaluons ensuite $D_{pp}(i,j)$ par la relation :

$$(5.42) \qquad D_{pp}(i,j) = D_{pp}(\hat{i}',\hat{j}') + d_p((\hat{i}',\hat{j}'), (i,j)) \quad .$$

Les relations (5.41) et (5.42) fournissent de nouvelles relations récursives qui permettent de relacher effectivement les contraintes aux frontières. Elles peuvent être considérées comme une généralisation des relations récursives classiques de programmation dynamique. Il faut noter que les relations (5.41) et (5.42) autorisent désormais l'utilisation de contraintes locales symétriques quel que soit le type de relaxation des contraintes aux frontières effectuée. C'est cette propriété intéressante qui nous a incité à appeler notre algorithme C.R.S.L.S. : contraintes relachées, stratégie locale symétrique.

V.9.3.4.- Spécification de C.R.S.L.S.

Les notations que nous allons utilisées pour spécifier C.R.S.L.S. sont les suivantes :

- α : abscisse du dernier point sur l'axe horizontal susceptible d'être l'origine d'un chemin optimal de recalage.

- β : ordonnée du dernier point sur l'axe vertical susceptible d'être l'origine d'un chemin optimal de recalage.

- γ : nombre de prélèvements en fin de forme, projetée sur l'axe horizontal, pouvant ne pas être pris en compte lors de la comparaison, moins 1.

- δ : nombre de prélèvements en fin de forme, projetée sur l'axe vertical, pouvant ne pas être pris en considération lors de la comparaison, moins 1.

- $D_{PP}(i,j)$: distance cumulée au point (i,j)

- $P_H(i,j)$: abscisse de l'origine du chemin optimal aboutissant au point (i,j) .

- $P_V(i,j)$: ordonnée de l'origine du chemin optimal aboutissant au point (i,j) .

- (i',j') : point appartenant à un voisinage de (i,j) défini par la contrainte locale, susceptible d'appartenir au chemin optimal ayant pour extrémité le point (i,j) .

- $(\hat{i}',\hat{j}')$: point appartenant à un voisinage de (i,j) défini par la contrainte locale par lequel passe effectivement le chemin optimal ayant pour extrémité le point (i,j) .

Dans le cas d'une pondération symétrique C.R.S.L.S. peut être spécifié ainsi :

Algorithme 3 :

a) Initialisation

- * Pour $1 \leqslant j \leqslant \beta$ FAIRE
  . $D_{PP}(1,j) = 2*\ d(1,j)$
  . $P_H(1,j) = 1$
  . $P_V(1,j) = j$
  * FIN DE FAIRE

b) Programmation dynamique.

- * Pour $2 \leqslant i \leqslant I$ Faire 1
  ** Pour $1 \leqslant j \leqslant J$ Faire 2

$$\hat{d}(i,j) = \min_{(i',j') \in V(i,j)} \frac{D_{PP}(i',j')+d_P((i',j'),(i,j))}{i-P_H(i',j')1+j-P_V(i',j')+1}$$

$$(\hat{i}',\hat{j}') = \operatorname{Arg\,min}_{(i',j') \in V(i,j)} \frac{D_{PP}(i',j')+d_P((i',j'),(i,j))}{i-P_H(i',j')+1+j-P_V(i',j')+1}$$

. Si $i \leqslant \alpha$ et $j \leqslant \beta$
et $d_{(i,j)} < \hat{d}_{(i,j)}$ alors

.. $D_{pp}(i,j) = 2.d(i,j)$

.. $P_H(i,j) = i$

.. $P_V(i,j) = j$

.. Fin de Si

. sinon

.. $D_{pp}(i,j) = D_{pp}(\hat{i}',\hat{j}') + d_p((\hat{i}',\hat{j}'), (i,j))$

.. $P_H(i,j) = P_H(\hat{i}',\hat{j}')$

.. $P_V(i,j) = P_V(\hat{i}',\hat{j}')$

. Fin de Si

** Fin de Faire 2

* Fin de Faire 1

c) Evaluation du taux de dissemblance

$$D(T,R) = \min_{\substack{i > I-\gamma \\ J-\delta < j \leq J}} \frac{D_{pp}(i,j)}{i-P_H(i,j)+1+j-P_V(i,j)+1}$$

L'étape 1 initialise la première colonne du plan de comparaison par la distance locale entre le $i^{ème}$ prélèvement de la forme située sur l'axe horizontal et le $j^{ème}$ prélèvement de la forme située sur l'axe vertical pondérée par un facteur 2 car la contrainte locale a été supposée être symétrique. Le pointeur $P_H(i,j)$ qui en tout point (i,j) du plan donne l'abscisse de l'origine du chemin optimal aboutissant en ce point est initialisé par 1. Quant à $P_V(i,j)$, le pointeur complémentaire de $P_H(i,j)$ donnant l'ordonnée de l'origine du chemin optimal est naturellement initialisé par j .

L'étape 2 évalue $D_{pp}(i,j)$, d'une part à l'intérieur de la région $\mathcal{D}$ du plan de comparaison qui contient les points susceptibles d'être origines de chemins optimaux de recalage et d'autre part à l'extérieur de $\mathcal{D}$. L'origine du chemin optimal aboutissant au point (i,j) , $P[(i,j)]$, est déterminé grâce à la relation suivante fournie par le principe d'optimalité local :

$$P[(i,j)] = P[(\hat{i}',\hat{j}')] \tag{5.43}$$

qui en termes de pointeur horizontal et vertical s'écrit :

$$P_H[(i,j)] = P_H[(\hat{i}',\hat{j}')] \tag{5.44}$$

$$P_V[(i,j)] = P_V[(\hat{i}',\hat{j}')] \tag{5.45}$$

Il faut remarquer que ces relations effectuent un chaînage des pointeurs P_H et P_V et que par conséquent elles permettent de déterminer en tout point (i,j) du plan de comparaison l'abscisse et l'ordonnée du chemin optimal de recalage aboutissant en ce point.

L'étape 3 évalue les distances cumulées normalisées en tout point de la région terminale définie par les relations suivantes :

$$J - \delta < j \leqslant J \tag{5.46.a}$$

$$i > I - \Upsilon \tag{5.46.b}$$

La distance minimale obtenue est le taux de dissemblance entre les deux formes comparées.

V.9.3.5.- Remarques concernant C.R.S.L.S

C.R.S.L.S., par sa puissante fonction de normalisation, permet de compenser des distorsions pouvant apparaître en début ou en fin de mot, ce qui en fait un algorithme tout à fait adapté à la reconnaissance de mots isolés en ambiance bruitée. En revanche, de par la nature même de la compensation effectuée, il ne se comportera pas de façon optimale dans le cas de vocabulaires contenant des formes vocales phonétiquement dissemblables uniquement au niveau des phonèmes de début ou de fin de mot. Toutefois nous pensons que l'introduction de fonctions de pénalités, convenablement choisies, pondérant les scores associés aux extrémités des chemins de recalage peut rendre C.R.S.L.S. adéquat même pour le type de vocabulaires particulièrement difficiles que nous venons d'évoquer.

V.10.- RESULTATS EXPERIMENTAUX.

-a) Nous avons testé cinq algorithmes de reconnaissance dont les caractéristiques sont les suivantes :

* Algorithme I : contrainte symétrique I sans relaxation des contraintes aux frontières.

* Algorithme II : contrainte symétrique de Sakoe et Chiba sans relaxation des contraintes aux frontières.

* Algorithme III = C.R.S.L.S. : contrainte symétrique de Sakoe et Chiba et fenêtre d'exploration de taille 2ε gérée comme dans l'U.E.L.M. - $\alpha = \beta = \gamma = \delta = 3$ et $\varepsilon = 6$ -.

* Algorithme IV : algorithme à optimums locaux avec pavés unicolonnes centrés sur la diagonale -$\varepsilon = 3$ - .

* Algorithme V : algorithme à optimums locaux avec pavés unicolonnes aux emplacements évoluant lors de la comparaison - $\varepsilon = 3$ - .

-b) Le vocabulaire testé est le vocabulaire des dix chiffres pour un locuteur masculin. Le nombre d'essais réalisés pour chaque mot est de 20.

-c) Les formes vocales ont été paramétrisées par le vocoder numérique 16 canaux, précédemment décrit, avec une fréquence d'échantillonnage des données de 100 HZ qui a été réduite par programme à 50 HZ, des expériences préliminaires ayant montré que les taux de reconnaissance n'étaient pas affectés par une diminution de la fréquence d'échantillonnage jusqu'à cette valeur.

Les taux d'erreur que l'on a obtenus avec ces différents algorithmes apparaissent sur la figure 36.

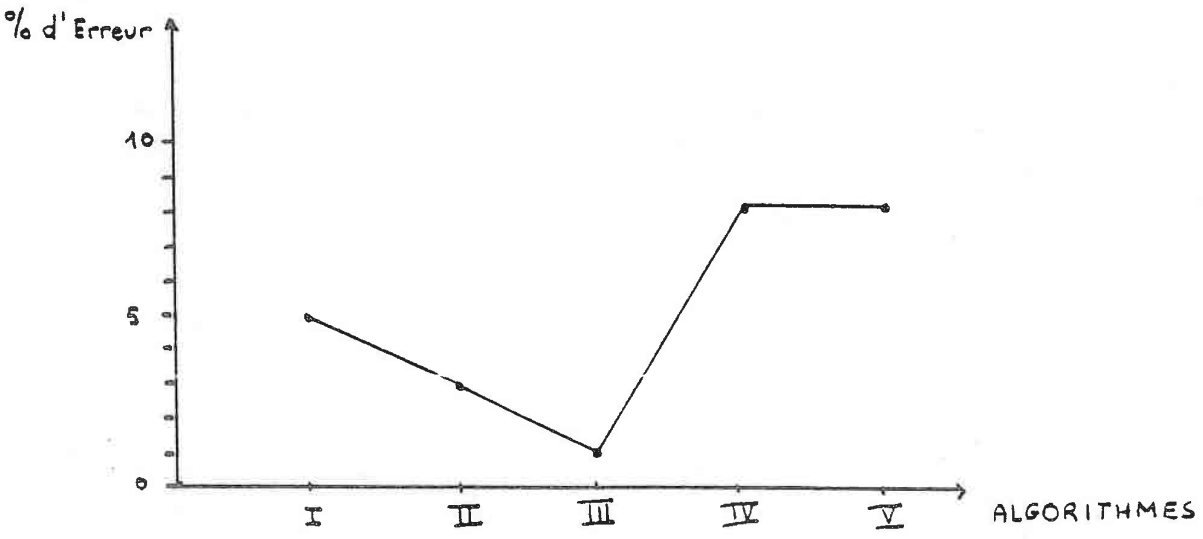


Figure 36 : Taux d'erreur obtenus avec les cinq algorithmes testés.

A partir de ces résultats on peut faire les constatations suivantes :

1) C.R.S.L.S. est l'algorithme le plus performant.
- 1 % d'erreur seulement -.

2) La contrainte I donne des résultats légèrement inférieurs à ceux fournis par la contrainte de Sakoe et Chiba.

3) Les deux algorithmes à optimums locaux ont fournis des résultats identiques et sensiblement moins bons que ceux obtenus avec C.R.S.L.S. - 7 % d'écart -.

C H A P I T R E VI

LA RECONNAISSANCE DE MOTS ENCHAINES

VI.1.- INTRODUCTION

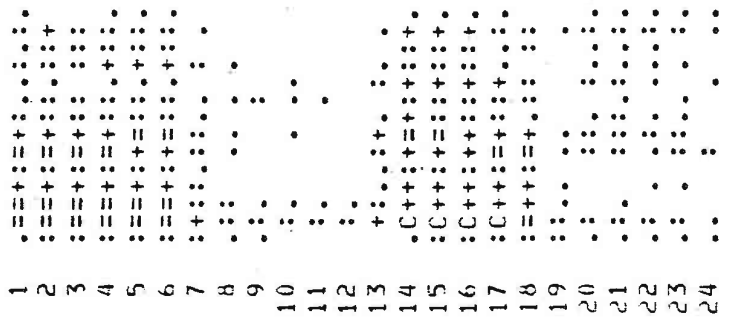
La reconnaissance de mots enchaînés est un des domaines de la reconnaissance de la parole des plus prometteurs et des plus passionnants. Ce constat est dû aux nombreux avantages de ce type de reconnaissance par rapport à la reconnaissance de mots isolés. En effet, alors que les systèmes de reconnaissance de mots isolés nécessitent pour leur bon fonctionnement qu'il y ait un silence entre les mots, les systèmes de reconnaissance de mots enchaînés autorisent une élocution continue. Le débit d'information avec de tels systèmes est par conséquent supérieur à celui que l'on obtient avec les systèmes de reconnaissance de mots isolés. Ce point de vue peut être perçu de façon plus explicite en considérant le nombre d'actions pouvant être exécutées par un système de reconnaissance de mots enchaînés par rapport à un système de reconnaissance de mots isolés. En effet si N est le nombre de mots du vocabulaire de l'application considérée, le nombre de travaux qui peuvent être mis en oeuvre après la phase de reconnaissance d'un programme de mots isolés est égal à N alors qu'un système de reconnaissance de mots enchaînés, en supposant que la longueur moyenne en mots des phrases prononcées soit égale à B pourra être à même d'entraîner l'exécution de B^N tâches différentes.

Ainsi les systèmes de reconnaissance de mots enchaînés se trouvent tout à fait adaptés à la commande de machines complexes et par conséquent devraient à court terme intéresser nombre de secteurs d'applications.

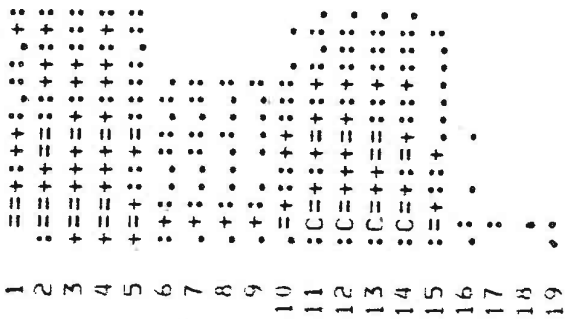
VI.2.- DIFFICULTES DU PROBLEME.

Le problème de la reconnaissance de mots enchaînés est beaucoup plus difficile que le problème de la reconnaissance de mots isolés pour plusieurs raisons :
d'une part on va retrouver les difficultés dues aux distorsions temporelles encore plus accentuées ici du fait de la plus grande variabilité de la vitesse d'élocution et d'autre part, un problème nouveau va rendre la tâche de reconnaissance particulièrement ardue, il s'agit du phénomène de la coarticulation qui a pour effet de déformer considérablement les zones initiale et terminale de mot et, par conséquent, de rendre les frontières de mot non apparentes.

La figure 37 visualise l'effet de la coarticulation sur la phrase "UN NEUF" en juxtaposant les sonogrammes relatifs à celle-ci dans le cas d'une élocution avec détachement des mots et dans le cas d'une élocution normale.



a) Elocution de la phrase "UN NEUF" avec détachement de mots



b) Elocution normale de la phrase "UN NEUF"

Figure 37 : L'effet de la coarticulation

On peut constater en effet que dans le cas de l'élocution normale, la consonne nasale /n/ du mot neuf est considérablement déformée.

VI.3.- DEFINITION EXPLICITE DU PROBLEME DE LA RECONNAISSANCE DE MOTS ENCHAINES.

Soit V un ensemble de formes de référence constituant le vocabulaire de l'application. Soit P une phrase inconnue prononcée par un locuteur à partir des éléments de V . Un système de reconnaissance de mots enchaînés se doit de déterminer tous les éléments de V se trouvant dans P , la vitesse d'élocution ainsi que le nombre de mots constituant la phrase à reconnaître étant des inconnues du problème.

VI.4.- PRINCIPE DE LA RECONNAISSANCE DE MOTS ENCHAINES.

Comme dans le cas de la reconnaissance de mots isolés, l'approche globale de la reconnaissance de mots enchaînés va mettre en jeu une forme inconnue - la phrase prononcée - et un ensemble de mots de référence constituant le vocabulaire de l'application. Plus précisément, le principe de la méthode, formulé la première fois par Sakoe [SAKO, 1979], consiste à comparer la phrase inconnue à l'ensemble des "super-formes" de références obtenues par concaténation d'un nombre quelconque de mots du vocabulaire. La super-forme de référence qui satisfait au mieux les critères de comparaison fournit la suite de mots constituant la phrase prononcée. La figure 38 montre de façon explicite le principe de la reconnaissance de mots enchaînés d'après le modèle de Sakoe. Comme on peut le constater d'ores et déjà, de par la définition même des super-formes de référence, le modèle de Sakoe ignore le problème de la coarticulation. Ultérieurement nous verrons comment il est possible de généraliser ce dernier afin que les distorsions dues à la coarticulation puissent être compensées.

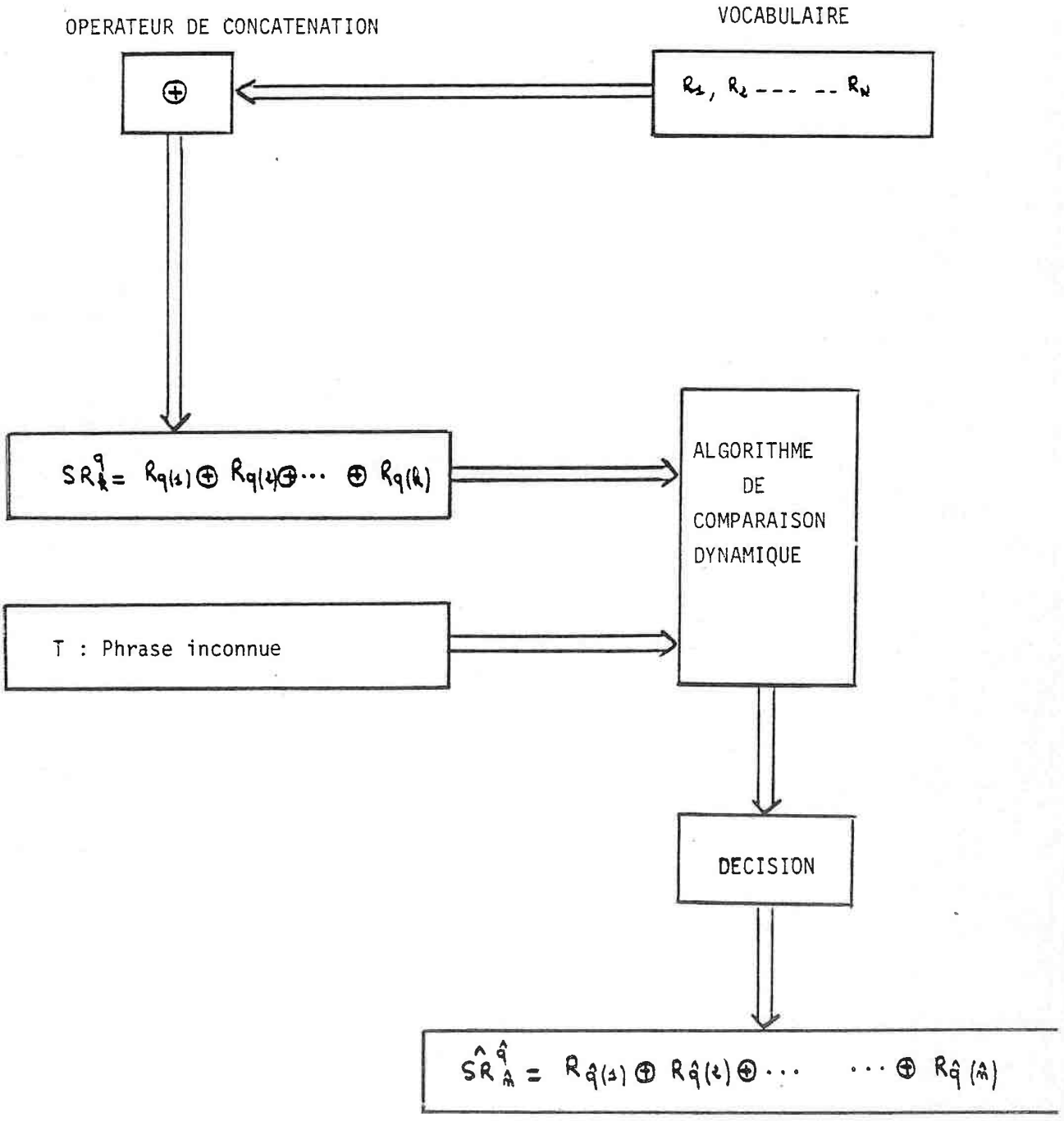


Figure 38 : Principe de la reconnaissance de mots enchaînés d'après le modèle de Sakoe [SAKO, 1979].

VI.5.- DECOMPOSITION DU PROBLEME DE LA RECONNAISSANCE DE MOTS ENCHAINES EN DEUX NIVEAUX : LE NIVEAU MOT ET LE NIVEAU PHRASE.

Désignons par :

- R_i la $i^{ème}$ forme de référence
- $r_i(j)$ le $j^{ème}$ prélèvement de la forme R_i .
- J_i la longueur en prélèvements de la forme de référence R_i .
- N le nombre de formes de références.
- T la phrase inconnue à reconnaître.
- t(i) le $i^{ème}$ prélèvement de la forme T .
- I la longueur de la phrase T .
- p_n une application de l'ensemble {0, 1, 2,..., n} vers lui-même réalisant une partition de T en n segments, comme le montrer la figure 39, satisfaisant les conditions aux frontières suivantes :

$$p_n(0) = 0 \quad \text{et} \quad p_n(n) = I$$

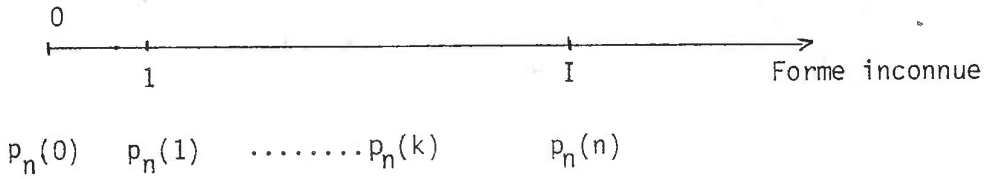


Figure 39 : Partition de la forme T réalisée par la fonction P_n .

- $T\,[p_n(k-1), p_n(k)]$ le $k^{ième}$ segment défini par la partition P_n autrement dit :

$$T\,[p_n(k-1),\ p_n(k)] = t_{p_n(k-1)+1} \cdots t_{p_n(k)} \tag{7.1.}$$

- SR_n^q la super-forme de référence engendrée par la concaténation des n formes $R_{q(k)}$ pour k allant de 1 à n et où q est une application de l'ensemble $\{1,2,\ldots, N\}$ vers lui-même, soit :

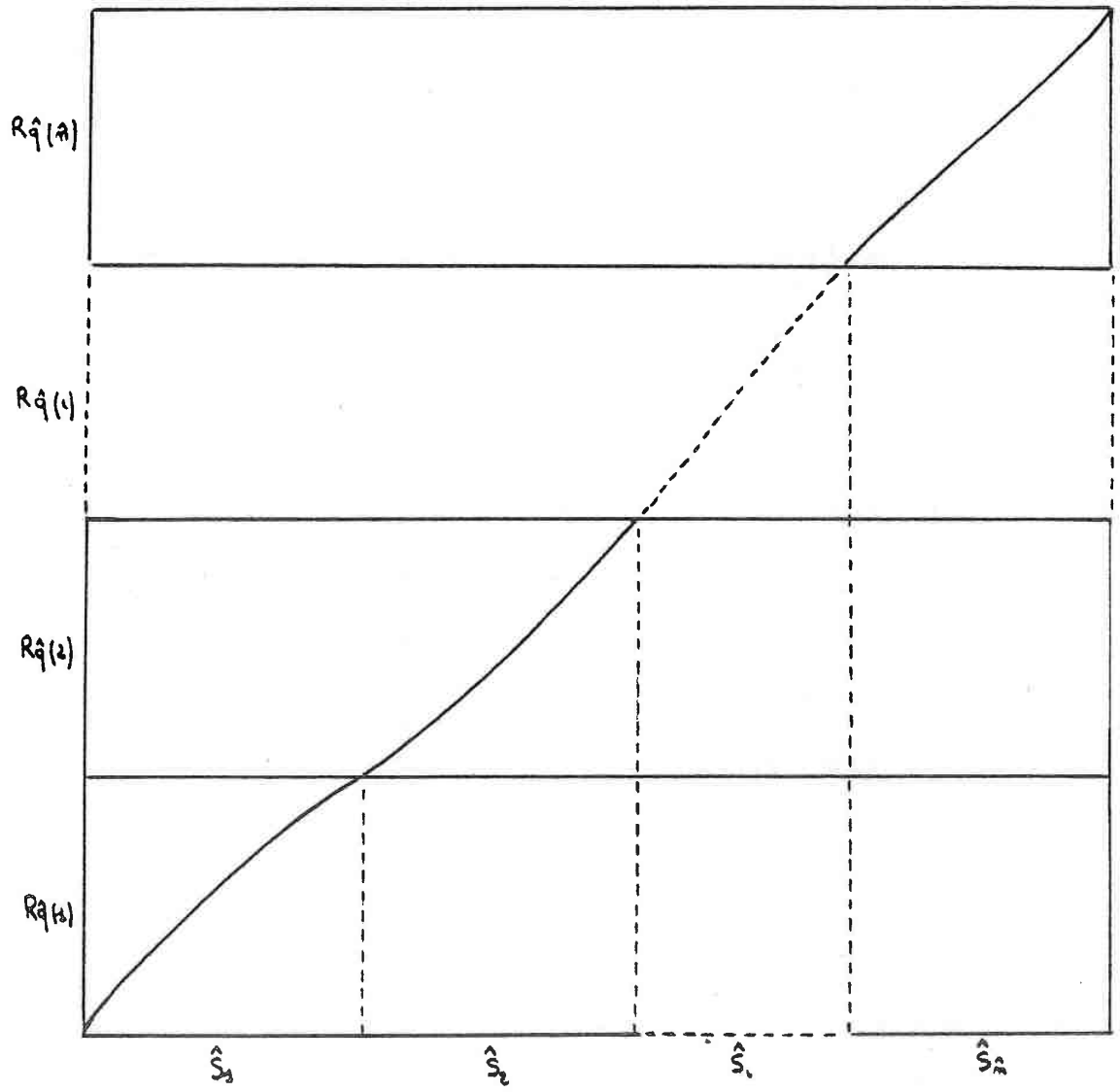
$$\begin{aligned} SR_n^q &= R_{q(1)} \oplus R_{q(2)} \oplus \ldots \oplus R_{q(n)} \\ &= r_{q(1)}(1) \ldots r_{q(1)}(J_1) \ldots r_{q(n)}(1) \ldots r_{q(n)}(J_n) \end{aligned} \tag{7.2}$$

La méthode de résolution du problème de la reconnaissance de mots enchaînés, dont on a vu le principe au paragraphe précédent, consiste à déterminer la super-forme de référence optimale $\hat{SR}_{\hat{n}}^{\hat{q}}$ solution de l'équation suivante :

$$D(T,\ \hat{SR}_{\hat{n}}^{\hat{q}}) = \min_{n,q} D(T,\ SR_n^q) \tag{7.3}$$

où D est la métrique que nous avons explicitée au paragraphe V.5 définissant le taux de dissemblance entre deux formes.

Pour résoudre l'équation (7.3) certaines considérations sur le chemin optimal effectuant le meilleur recalage temporel entre la forme T et la super-forme de référence optimale $\hat{SR}_{\hat{n}}^{\hat{q}}$ sont nécessaires. La figure 40 visualise un tel chemin de recalage.



Exemple 40 : Exemple de chemin de recalage optimal entre la forme T et la super-forme de référence optimale $R_{\hat{q}(1)} \oplus R_{\hat{q}(2)} \oplus \ldots \oplus R_{\hat{q}(\hat{n})}$

Comme on peut le constater sur la figure ci-dessus le chemin de recalage optimal peut être considéré comme étant une suite de sous-chemins $\hat{SC}_i$ effectuant chacun une normalisation temporelle entre un segment $\hat{S}_i$ - que l'on peut qualifier d'optimal - de la phrase inconnue et la forme de référence optimal $R_{\hat{q}(i)}$. Or, comme dans le cas de la reconnaissance de mots isolés il est avantageux de déterminer ce chemin de recalage par une technique de programmation dynamique. Il faut donc qu'il satisfasse au principe de Bellman. Ce qui implique que chaque sous-chemin doit être optimal, en ce sens que chaque $\hat{SC}_i$ doit effectuer le meilleur recalage entre le segment $\hat{S}_i$ et la forme de référence $R_{\hat{q}(i)}$. Il s'ensuit que le score associé au sous-chemin $\hat{SC}_i$ est obligatoirement $D[\hat{S}_i, R_{\hat{q}(i)}]$. En désignant par $D^*[\hat{S}_i, R_{\hat{q}(i)}]$ et $D^*[T, SR^{\hat{q}}_{\hat{n}}]$ les distances cumulées non normalisées associées respectivement au sous-chemin $\hat{SC}_i$ et au chemin de recalage optimal, de par la définition de D^* , il vient que la super-forme de référence optimale au sens de Bellman doit vérifier la relation suivante :

$$D^*[T, SR^{\hat{q}}_{\hat{n}}] = \sum_{i=1}^{\hat{n}} D^*[\hat{S}_i, R_{\hat{q}(i)}] \qquad (7.4)$$

Dans le cas où la contrainte locale utilisée est asymétrique, tous les chemins de recalage ont une longueur identique à savoir I, si bien que la quantité $D^*[T, SR^{\hat{q}}_{\hat{n}}]$ mesure effectivement le taux de dissemblance optimal entre les formes T et $SR^{\hat{q}}_{\hat{n}}$.

D'après l'équation (7.4) qui explicite la solution problème, il vient que (7.3) peut être résolue par la relation :

$$(7.5)\quad D^*[T, SR_{\hat{n}}^{\hat{q}}] = \underset{n, p_n(k), q(k)}{\text{Min}} \left\{ \sum_{k=1}^{n} D^* \left[T\left(p_n(k-1), p_n(k)\right), R_{q(k)} \right] \right\}$$

Par contre dans le cas où la contrainte locale est symétrique alors $D^*[T, SR_{\hat{n}}^{\hat{q}}]$ n'est pas représentatif du taux de dissemblance entre T et $SR_{\hat{n}}^{\hat{q}}$ car la longueur du chemin de recalage dépend de la longueur en prélèvements des formes $R_{q(k)}$. La solution du problème est alors donnée par la relation :

$$(7.6)\quad D(T, SR_{\hat{n}}^{\hat{q}}) = \underset{n, p_n(k), q(k)}{\text{Min}} \frac{1}{I + \sum_{k=1}^{n} J_{q(k)}} \cdot \sum_{k=1}^{n} D^* \left[T\left(p_n(k-1), p_n(k)\right), R_{q(k)} \right]$$

où le facteur $\dfrac{1}{I + \sum_{k=1}^{n} J_{q(k)}}$ permet de tenir compte de la longueur du chemin de recalage.

La relation (7.6) exprime un problème de minimisation de fonctionnelle avec variation des longueurs des chemins de recalage. Nous avons déjà résolu ce problème dans le cas de la reconnaissance de mots isolés. Nous verrons au paragraphe VI.7 lorsque nous discuterons de la relaxation des contraintes aux frontières dans le cadre de la reconnaissance de mots enchaînés comment il est possible de solutionner (7.6). Pour l'instant pour simplifier l'exposé nous nous plaçons dans le cas où la contrainte locale utilisée est asymétrique, et par conséquent nous allons nous intéresser essentiellement à la solution du problème de la reconnaissance de mots enchaînés formulée par la relation (7.5).

Cette dernière peut être réécrite de la façon suivante :

$$(7.7)\quad D(T, SR_{\hat{n}}^{\hat{q}}) = \min_{n, p_n(k)} \left[\sum_{k=1}^{n} \min_{q(k)} D^* \left[T\left(p_n(k-1), p_n(k)\right), R_{q(k)} \right] \right]$$

Cette nouvelle expression met en évidence deux niveaux de minimisation :

- un premier niveau, le niveau mot, nécessitant le calcul des quantités :

$$(7.8)\quad \hat{D}(i,j) = \min_{k} D\left(T(i,j), R_k\right)$$

pour tout i et j avec $j > i$ par un algorithme de programmation dynamique, et,

- un deuxième niveau, le niveau phrase, permettant de trouver la super-forme de référence optimale SR_n^q solution de l'équation :

$$(7.9)\quad D^*\left[T, SR_{\hat{n}}^{\hat{q}}\right] = \min_{n, p_n(k)} \sum_{k=1}^{n} \hat{D}\left[p_n(k-1), p_n(k)\right]$$

VI.6.- DESCRIPTION DES PRINCIPAUX ALGORITHMES DE RECONNAISSANCE DE MOTS ENCHAINES EXISTANTS.

VI.6.1.- Introduction.

Les trois algorithmes de reconnaissance de mots enchaînés qui vont être décrits dans ce paragraphe sont tous fondés sur les considérations théoriques que nous venons d'expliciter. Autrement dit il s'agit d'algorithmes qui conservent au problème son caractère optimal.

Le premier algorithme que nous allons présenter est celui de Sakoe [SAKO, 1979]. Cet algorithme est fondé directement à partir des relations (7.8) et (7.9) et constitue la première solution proposée pour résoudre celles-ci.

Le deuxième algorithme auquel nous allons nous intéresser a été présenté par Myers [MYER, 1981] . Il a été déduit à partir d'une reformulation des relations (7.8) et (7.9) qui permet de diminuer de façon considérable la charge de calcul devant être mise en oeuvre par l'algorithme de Sakoe.

Le troisième algorithme que nous allons décrire est celui de Bridle et de Nakagawa [BRID, 1982] , [NAKA, 1983]. Cet algorithme est très intéressant car, d'une part, il permet de solutionner (7.8) et (7.9) en ne parcourant la phrase inconnue qu'en une seule passe et, d'autre part, il nécessite une charge de calcul encore moindre que celle relative à l'algorithme de Myers.

D'autres algorithmes employant une stratégie sous-optimale ont été proposés [RABI, 1980], [GAUV, 1982] . Quoique ceux-ci fournissent actuellement des résultats tout à fait comparables à ceux obtenus avec les algorithmes optimaux, nous pensons toutefois qu'ils ne sont pas très bien adaptés pour évoluer vers des algorithmes plus généraux pouvant compenser les effets dûs à la coarticulation.

VI.6.2.- L'algorithme de Sako : "The Two level DP Matching" [SAKO, 1979].

VI.6.2.1.- Présentation de l'algorithme.

L'algorithme de Sako est issu directement des relations (7.8) et (7.9). C'est un algorithme à deux niveaux de programmation dynamique.

Le premier niveau évalue à l'aide d'une contrainte locale asymétrique les distances partielles optimales données par la relation (7.8) à savoir :

$$\hat{D}(i,j) = \min_{k} \left[D\left(T(i,j), R_k\right) \right] \tag{7.10}$$

pour tout i et j avec $j > i$, $1 \leqslant i \leqslant I$ et $i \leqslant j \leqslant J_k$ et mémorise dans une table $\hat{N}$ la forme de référence qui a permis d'obtenir $D(i,j)$

$$\hat{N}(i,j) = \text{Arg} \min_{k} \left[D\left(T(i,j), R_k\right) \right] \tag{7.11}$$

Le deuxième niveau de minimisation, relatif à la phrase, détermine le taux de dissemblance formulé par la relation (7.9), en calculant le score $\hat{D}_n^B(i)$ du meilleur chemin de recalage entre la super-forme de référence optimale constituée de n mots et les i premiers prélèvements de la phrase inconnue, récursivement pour n allant de 1 à LMAX, où LMAX désigne le nombre de mots maximum pouvant être contenus dans la phrase à reconnaître et pour tout indice i relatif à celle-ci, par les relations de programmation dynamique suivantes :

$$(7.12.a) \quad \hat{D}_0^B(0) = 0 \quad \text{et} \quad \hat{D}_0^B(i) = \infty \quad \text{pour} \quad 1 \leqslant i \leqslant I$$

$$(7.12.b) \quad \hat{D}_n^B(i) = \min_h \left(\hat{D}_{n-1}^B(h) + \hat{D}(h,i) \right)$$

De plus la valeur $\hat{h}$ vérifiant (7.12.b), qui donne en fait le dernier prélèvement de la super-forme de référence de $n-1$ mots qui a permis d'obtenir $\hat{D}_n^B(i)$ est rangée dans une table auxiliaire $\hat{F}_n^B(i)$:

$$(7.13) \quad \hat{F}_n^B(i) = \text{Arg} \min_h \left(\hat{D}_{n-1}^B(h) + \hat{D}(h,i) \right)$$

Lorsque la relation (7.12.b) a été évaluée pour tout n et pour tout i, le nombre de mots contenus dans la phrase inconnue est alors déterminé par

$$(7.14) \quad \hat{n} = \text{Arg} \min_n \hat{D}_n^B(I) \quad .$$

Grâce à $\hat{n}$ et à $\hat{F}_n^B$ il est possible de construire la partition optimale $\hat{p}_{\hat{n}}$ donnant les indices des prélèvements de fin de mot par les relations de chaînage arrière suivantes :

$$(7.15.a) \quad \hat{p}_{\hat{n}}(\hat{n}) = I$$

$$(7.15.b) \quad \hat{p}_{\hat{n}}(n) = \hat{F}_{n+1}^B \left(\hat{p}_{\hat{n}}(n+1) \right) \quad \text{pour} \quad n = \hat{n}-1, \ldots, 1$$

La partition optimale étant connue, la table $\hat{N}$ permet alors de déterminer les différents mots $\hat{q}(k)$ constituant la phrase inconnue :

(7.16) $\hat{q}(k) = \hat{N}\left[\hat{p}_{\hat{n}}(k-1),\ \hat{p}_{\hat{n}}(k)\right]$ pour $k = 1,2,\ldots,\hat{n}$.

La super-forme de référence trouvée est donc :

$$\hat{SR}_{\hat{n}}^{\hat{q}} = R_{\hat{q}(1)} \oplus R_{\hat{q}(2)} \oplus \ldots \oplus R_{\hat{q}(\hat{n})}$$

VI.6.2.2.- Spécification de l'algorithme de Sakoe.

En résumé l'algorithme de Sakoe peut être spécifié ainsi :

Algorithme 4 :

a) Niveau mot :

Pour $1 \leqslant i \leqslant I$, pour $1 \leqslant j \leqslant I$ avec $j > i$
évaluer à l'aide d'une contrainte locale asymétrique 1 et 2.

1- $\hat{D}(i,j) = \min_k \left[D\left(T(i,j), R_k\right)\right]$

2- $\hat{N}(i,j) = \text{Arg}\min_k \left[D\left(T(i,j), R_k\right)\right]$

b) Niveau phrase :

3- Faire $\hat{D}_0^B(0) = 0$ et $\hat{D}_0^B(i) = \infty$ pour $1 \leqslant i \leqslant I$.

4- Pour $1 \leqslant n \leqslant LMAX$, pour $1 \leqslant i \leqslant I$ Faire 5 et 6

5- $\hat{D}_n^B(i) = \min_h \left(\hat{D}_{n-1}^B(h) + \hat{D}(h,i)\right)$

6- $\hat{F}_n^B(i) = \text{Arg}\min_h \left(\hat{D}_{n-1}^B(h) + \hat{D}(h,i)\right)$

c) Décision

7- $\hat{n} = \underset{n}{\text{Arg min}} \ \hat{D}^B_n(I)$

8- $\hat{p}_{\hat{n}}(\hat{n}) = I$

9- $\hat{p}_{\hat{n}}(n) = \hat{F}^B_{n+1}\left(\hat{p}_{\hat{n}}(n+1)\right)$ pour $n = \hat{n}-1, \ldots, 1$.

10- $\hat{q}(k) = \hat{N}\left[\hat{p}_{\hat{n}}(k-1), \ \hat{P}_{\hat{n}}(k)\right]$ pour $k=1,2,\ldots, \hat{n}$

11- $\hat{SR}^{\hat{q}}_{\hat{n}} = R_{\hat{q}(1)} \oplus R_{\hat{q}(2)} \oplus \ldots \oplus R_{\hat{q}(\hat{n})}$

VI.6.2.3.- Remarques concernant l'algorithme de Sakoe.

L'algorithme de Sakoe est véritablement le premier algorithme de reconnaissance de mots enchaînés qui a permis par sa caractéristique de double programmation dynamique de conserver au problème son caractère optimal.

Afin de pouvoir comparer la performance de l'algorithme de Sakoe par rapport à d'autres algorithmes de reconnaissance de mots enchaînés, le meilleur moyen consiste à déterminer le nombre de distances locales, NDL, considérées par l'algorithme. Dans le cas de la "Two level DP Matching", NDL est donné approximativement par :

(7.17) $\qquad NDL = I.N.\bar{J}.\ (2R+1)$

où : - le facteur I.N est le nombre de comparaisons dynamiques effectuées par l'algorithme de Sakoe - une par prélèvement de la forme inconnue et par forme de référence afin d'évaluer les quantités $\hat{D}(i,j)$ -, - le facteur $\bar{J}(2R+1) - \bar{J}$ étant la longueur moyenne des formes de référence et R le paramètre de la contrainte globale de Sakoe et Chiba - est sensiblement le nombre de distances locales évaluées par comparaison dynamique.

En prenant pour I, N, $\bar{J}$ et R les valeurs suivantes respectivement : 40, 10, 16 et 6 alors NDL vaut : 83 200.

VI.6.3.- L'algorithme de Myers : "The level Building DP Matching" [MYER, 1981] .

VI.6.3.1.- Présentation de l'algorithme.

Pour déduire son algorithme, Myers a réécrit la relation (7.5) en inversant les minimisations par rapport à p et q, ce qui donne :

$$(7.18)\quad D(T, \hat{SR}_{\hat{n}}^{\hat{q}}) = \min_{n,q(k),p_n(k)} \sum_{k=1}^{n} D\left[T\left(p_n(k-1), p_n(k)\right), R_{q(k)}\right]$$

Cette nouvelle relation peut être évaluée comme pour l'algorithme de Sakoe à l'aide de $\hat{D}_n^B$, $\hat{F}_n^B$ et $\hat{N}$ - en fait ici $\hat{N}_n^B$ puisqu'étant donné que la première minimisation s'effectue pour toute partition, on sera à même de connaître au niveau n le dernier mot de la super-forme optimale, constituée par la concaténation de n formes de référence, coïncidant au $i^{ème}$ prélèvement de la phrase à reconnaître - récursivement pour n allant de 1 à LMAX et pour chaque prélèvement de la forme inconnue, à la seule différence qu'il faut inverser les minimisations par rapport à $p_n(k)$ et $q(k)$. A partir des relations (7.10), (7.12) et (7.13) de Sakoe il vient :

$$(7.19)\quad \hat{D}_n^B(i) = \min_k \min_h \left(\hat{D}_{n-1}^B(h) + D\left(T(h,i), R_k\right)\right)$$

$$(7.20)\quad \hat{F}_n^B(i) = \operatorname*{Arg\,min}_h \min_k \min_h \left(\hat{D}_{n-1}^B(h) + D\left(T(h,i), R_k\right)\right)$$

$$(7.21)\quad \hat{N}^B_n(i) = \underset{k}{\text{Arg}}\ \underset{k}{\text{Min}}\ \underset{h}{\text{Min}}\ \left(\hat{D}^B_{n-1}(h) + D\left(T(h,i),\ R_k\right)\right)$$

L'intérêt de ces relations par rapport à celles de Sakoe réside dans le fait que l'expression

$$\underset{h}{\text{Min}}\ \left(\hat{D}^B_{n-1}(h) + D\left(T(h,i),\ R_k\right)\right)$$

peut être implémentée en une seule comparaison dynamique en initialisant les points de départ d'abscisse h de l'algorithme de programmation dynamique qui doit évaluer $D\left(T(h,i),\ R_k\right)$ pour tout h et pour tout i par $D^B_{n-1}(h)$ comme le montre la figure 41.

Il vient donc que pour évaluer (7.19), (7.20) et (7.21) LMAX programmations dynamiques suffisent. On peut donc s'attendre avec l'algorithme de Myers à un gain de temps de calcul considérable par rapport à l'algorithme de Sakoe qui a nécessité I comparaisons dynamiques.

VI.6.3.2.- Spécification de l'algorithme.

En désignant par :

- $D^r_{pp_n}(i,j)$ le score, au $i^{ème}$ prélèvement de la phrase à reconnaître, associé à la super-forme de référence obtenue par la concaténation de la sous-chaîne optimale de $n-1$ mots, qui a permis d'obtenir $D^r_{pp_n}(i,j)$, avec les j premiers prélèvements de la forme de référence r ;

- $F^r_n(i,j)$ le dernier prélèvement de la sous-chaîne optimale de $n-1$ mots qui a permis d'obtenir $D^r_{pp_n}(i,j)$;

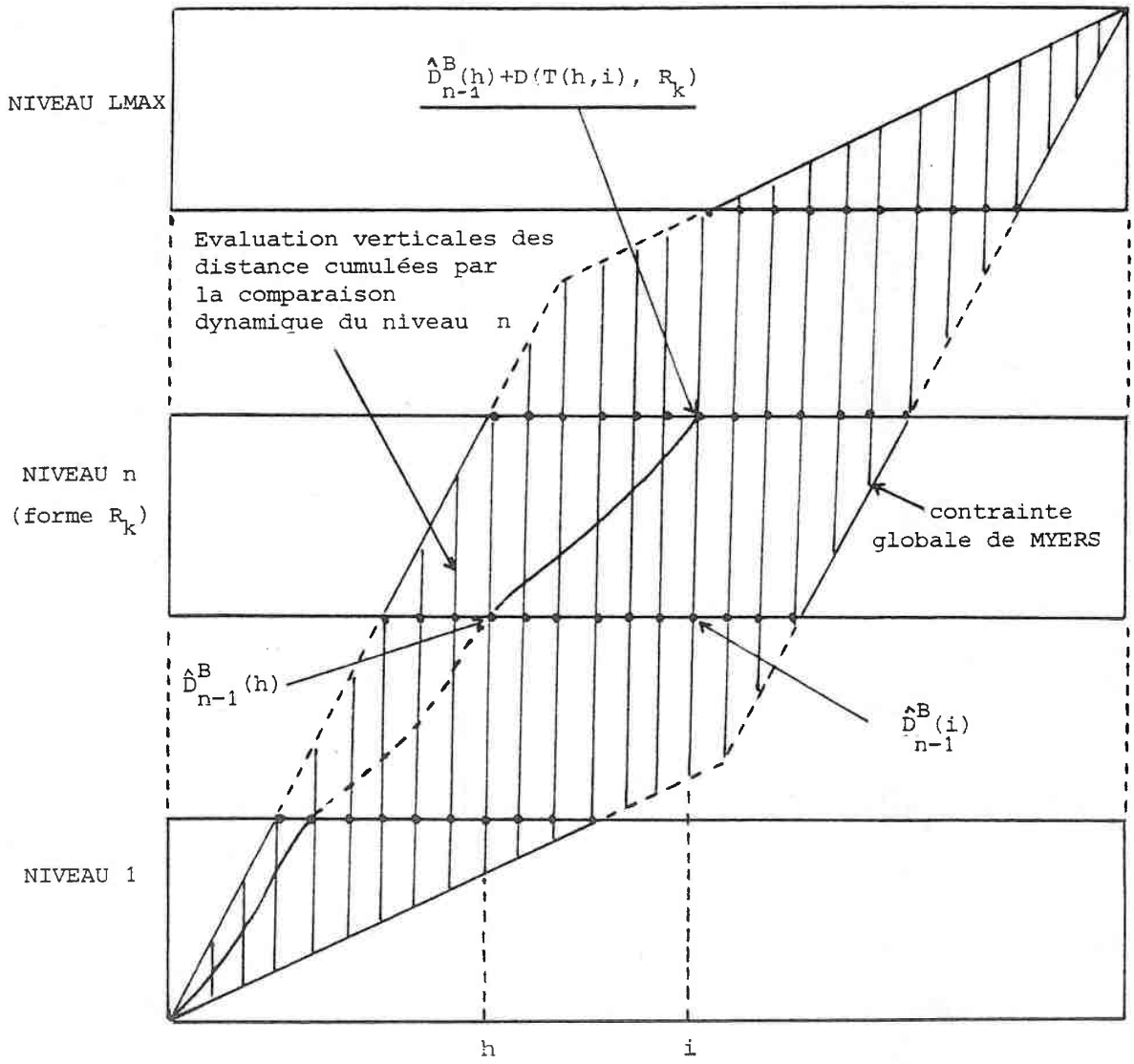


Figure 41 : Processus de minimisation par niveau de la "level DP Matching".

- $d^r(i,j)$ la distance locale entre le $i^{ème}$ prélèvement de la phrase à reconnaître et le $j^{ème}$ prélèvement de la forme r ;

et en supposant que la contrainte locale utilisée est la contrainte d'Itakura -asymétrique- simplifiée en ce sens qu'elle n'interdit pas deux déplacements horizontaux consécutifs, la spécification de l'algorithme de Myers est la suivante :

Algorithme 5 :

a) Initialisation générale

1- $\hat{D}_0^B(0) = 0$, $\hat{D}_0^B(i) = \infty$ pour $1 \leq i \leq I$

b) Programmation dynamique en niveau :

2- Pour $1 \leq n \leq LMAX$ Faire de 3 à 6

3- Pour $1 \leq r \leq N$ Faire de 4 à 5

4- Initialisations du niveau n

. Pour $1 \leq i \leq I$ Faire

.. $D_{pp_n}^r(i,0) = D_{n-1}^B(i)$

.. $D_{pp_n}^r(i,J_r) = \infty$

.. $F_n^r(i,0) = i$

. Fin de Faire

5- Comparaison dynamique du niveau n :

. Pour $1 \leq i \leq I$ et Pour $IMIN_n(i) \leq j \leq IMAX_n(i)$

Faire

(* $IMIN_n$ et $IMAX_n$ définissant la contrainte globale au niveau n).

.. $\hat{j} = \underset{\substack{MAX(0,j-2) \leq j' \leq j \\ IMIN_n(i-1) \leq j' \leq IMAX_n(i-1)}}{Arg\ min} D^r_{pp_n}(i-1, j')$

.. $D^r_{pp_n}(i,j) = D^r_{pp_n}(i-1,\hat{j}) + d^r(i,j)$

.. $F^r_n(i,j) = F^r_n(i-1,\hat{j})$

. FIN DE FAIRE

6- Minimisations du niveau phrase :

. POUR $1 \leq i \leq I$ Faire :

.. $\hat{D}^B_n(i) = \underset{r}{Min}\ D^r_{pp_n}(i,J_r)$

.. $\hat{N}^B_n(i) = \underset{r}{Arg\ min}\ D^r_{pp_n}(i,J_r)$

.. $\hat{F}^B_n(i) = F^{\hat{N}^B_n(i)}_n\left(i, J_{\hat{N}^B_{n(i)}}\right)$

. FIN DE FAIRE

C) Décision :

7- $\hat{n} = \underset{n}{Arg\ min}\ \hat{D}^B_n(I)$

8- $\hat{P}_{\hat{n}}(\hat{n}) = I$

9- $\hat{p}_{\hat{n}}(n) = \hat{F}^B_{n+1}\left(\hat{p}_{\hat{n}}(n+1)\right)$ pour $n = \hat{n}-1, \ldots, 1$.

10- $\hat{q}(k) = \hat{N}^B_k\left(\hat{p}_{\hat{n}}(k)\right)$ pour $k = \hat{n}, \ldots, 1$

11- $\hat{SR}^{\hat{q}}_{\hat{n}} = R_{\hat{q}(1)} \oplus R_{\hat{q}(2)} \oplus \ldots \oplus R_{\hat{q}(\hat{n})}$

VI.6.3.3.- Remarques concernant l'algorithme de Myers.

En supposant que la contrainte locale utilisée est caractérisée par une pente maximale de 2 et une pente minimale de 1/2 et que le chemin optimal est recherché dans le domaine du plan de comparaison défini par la contrainte globale de Myers, visualisée par la figure 41, alors le nombre de distances locales calculées par la "level Building DP Matching" est :

(7.22) $$NDL = LMAX.\ N.\ \bar{J}.\ I/3$$

où : - LMAX. N est le nombre de comparaisons dynamiques effectuées et

- $(\bar{J}.I)/3$ est le nombre de distances locales évaluées par comparaison dynamique.

En prenant LMAX = 5 et pour N, J et I les valeurs qui ont été prises lors de la description de l'algorithme de Sakoe pour évaluer la performance de ce dernier, à savoir, N = 10, J = 16, I = 40 alors NDL vaut 10 666. Il vient donc que, pour les valeurs particulières qui ont été choisies il existe un rapport de 8 entre le nombre de distances locales évaluées par l'algorithme de Sakoe et celui de Myers. On peut donc affirmer que la "level Building DP Matching" est beaucoup plus performante que la "Two level DP Matching".

L'algorithme de Myers, bien qu'étant de ce point de vue très intéressant, présente un défaut dû au fait qu'il n'est pas très bien orienté vers le temps réel. En effet, lors des changements de niveaux, l'algorithme est obligé d'effectuer un retour-arrière local au niveau de la forme inconnue pour poursuivre la comparaison.

De plus, à cause de ces retours-arrière locaux, le nombre de distances locales évaluées n'est pas optimal, un certain nombre d'entre elles étant recalculées lors de chaque début de niveau.

VI.6.4.- L'algorithme de Bridle et de Nakagawa . [BRID, 1982] , [NAKA, 1983].

VI.6.4.1.- Présentation de l'algorithme.

Bridle et Nakagawa ont déduit leur algorithme en faisant la remarque préliminaire suivante :
l'algorithme de Sakoe ainsi que celui de Myers nécessitent une minimisation pour tout n sur les scores $\hat{D}^B_n(I)$ associés aux meilleures superformes de référence de n mots, coïncidant au dernier prélèvement de la forme inconnue, - il s'agit de l'étape 7 des deux algorithmes dont le but est de déterminer le nombre de mots de la phrase testée - qui, étant donné qu'elle a lieu en fin de comparaison, impose l'introduction d'un paramètre LMAX, fixant un nombre de mots maximum arbitraire pouvant être contenus dans la phrase à reconnaître, afin que le processus itératif du niveau phrase ait une condition d'arrêt. Bridle et Nakgawa suggèrent de s'affranchir de cette étape et surtout de supprimer le mécanisme de décomposition du niveau phrase en sous-niveaux en effectuant une telle minimisation pour chaque prélèvement de la forme à identifier c'est-à-dire en minimisant pour tout n les relations (7.19) (7.20) et (7.21) de Myers :

$$(7.23)\quad \min_n \hat{D}^B_n(i) = \min_k \min_h \left(\min_n \hat{D}^B_{n-1}(h) + D\left(T(h,i), R_k\right)\right)$$

$$(7.24)\quad \min_n \hat{F}^B_n(i) = \operatorname{Arg}_h \min_k \min_h \left(\min_n \hat{D}^B_{n-1}(h) + D\left(T(h,i), R_k\right)\right)$$

$$(7.25)\quad \min_n \hat{N}^B_n(i) = \arg\min_k \min_h \left(\min_n \hat{D}^B_{n-1}(h) + D\left(T(h,i), R_k\right)\right)$$

Et en désignant par $\hat{D}^B(i)$, $\hat{F}^B(i)$, $\hat{N}^B(i)$ respectivement

$$\min_n \hat{D}^B_n(i)\ ,\quad \min_n \hat{F}^B_n(i)\ ,\quad \min_n \hat{N}^B_n(i)$$

(7.23), (7.24) et (7.25) s'écrivent :

$$(7.26)\quad \hat{D}^B(i) = \min_k \min_h \left(\hat{D}^B(h) + D\left(T(h,i), R_k\right)\right)$$

$$(7.27)\quad \hat{F}^B(i) = \arg\min_h \min_k \min_h \left(\hat{D}^B(h) + D\left(T(h,i), R_k\right)\right)$$

$$(7.28)\quad \hat{N}^B(i) = \arg\min_k \min_k \min_h \left(\hat{D}^B(h) + D\left(T(h,i), R_k\right)\right)$$

Les relations (7.26), (7.27) et (7.28) fournissent la meilleure super-forme de référence pour tout prélèvement i de la phrase inconnue et donc, a fortiori, la super-forme de référence optimale au dernier prélèvement de la phrase testée.

L'intérêt des relations (7.26), (7.27) et (7.28) est, à ce stade, assez difficile à saisir car il faut bien s'imprégner du fait qu'elles effectuent à la fois la minimisation du niveau mot et du niveau phrase en une seule passe, c'est-à-dire en parcourant la forme inconnue une seule fois alors que pour les algorithmes de Sakoe et de Myers, le niveau phrase, de par la décomposition du processus de comparaison en niveaux, nécessite LMAX parcours de la phrase à reconnaître. On peut donc s'attendre à obtenir avec de telles relations un algorithme de reconnaissance de mots enchaînés parfaitement adapté au temps réel. Et effectivement il en est ainsi grâce à la deuxième remarque astucieuse de Bridle et de Nakagawa :

si l'algorithme de programmation dynamique, qui doit évaluer au prélèvement i pour tout $h < i$ et pour la forme de référence k la quantité $D\left(T(h,i), R_k\right)$, a initialisé, d'une part, les points $(h,0)$ du plan de comparaison qui sont autant de points de départ pour l'algorithme, par $\hat{D}^B(h)$, alors, comme le visualise la figure 42, au point (i, J_k) celui-ci aura évalué :

$$\tilde{D}(i,k) = \min_h \left(\hat{D}^B(h) + D\left(T(h,i), R_k\right)\right) \tag{7.29}$$

Figure 42 : Représentation schématique du fonctionnement de l'algorithme de Bridle et de Nakagawa : l'évaluation des distances cumulées $D^k(i,j)$ au prélèvement i nécessite l'initialisation, lors des étapes précédentes, des points $(h,0)$, $h < i$, par les quantités $\hat{D}^B(h)$. Lorsque les valeurs $\tilde{D}(i,k) = D^k(i,J_k)$ ont été évaluées pour toutes les formes de référence alors la distance cumulée optimale $\hat{D}^B(i) = \min_k \tilde{D}(i,k)$ est associée aux points $(i,0)$ de tous les sous-plans afin que le processus puisse être réitéré.

Evaluation verticale des distances cumulées par l'algorithme de comparaison dynamique au prélèvement k .

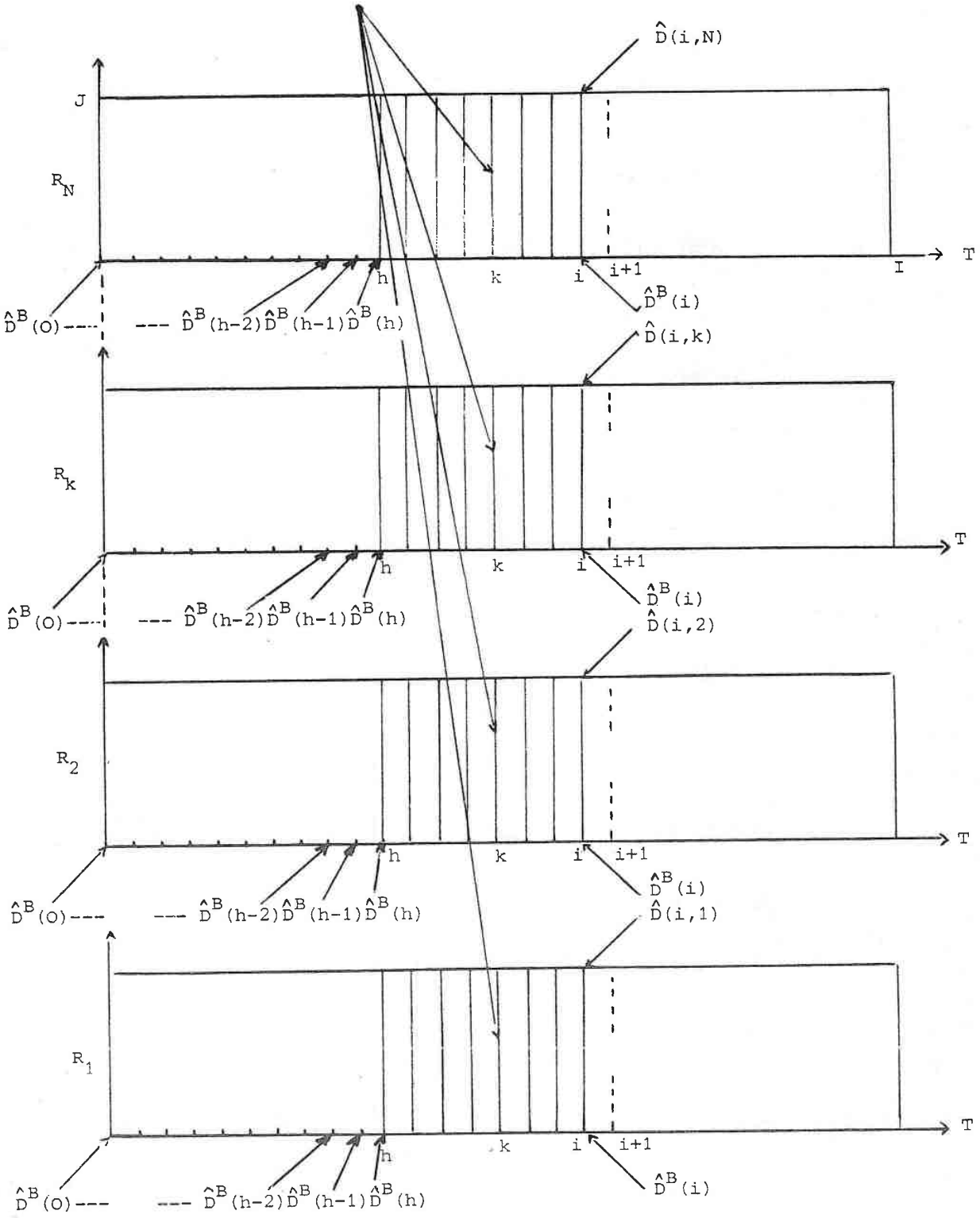


Figure 42 : Description du fonctionnement de l'algorithme de Bridle et de Nakagawa.

et il suffira, lorsque $\tilde{D}(i,k)$ auront été évaluées pour toutes les formes de référence k , de déterminer $\underset{k}{\text{Min}}\ \tilde{D}(i,k)$ pour connaître $\hat{D}^B(i)$ soit :

$$\hat{D}^B(i) = \underset{k}{\text{Min}}\ \tilde{D}(i,k) \quad ; \tag{7.30}$$

et si de plus, d'autre part, l'algorithme a conservé en mémoire, à l'étape $i-1$, les distances cumulées partielles

$D^k(i-1, j)$ - donnant $\underset{h}{\text{Min}} \left(\hat{D}^B(h) + D\left(T(h,i-1), R_k\ [1,j]\right)\right)$ ou

$R_k[1,j]$ désigne les j premiers prélèvements de la forme de référence R_k - pour toutes les formes de référence R_k et pour tout $j - 1 \leqslant j \leqslant J_k$ — alors — en supposant que la contrainte locale utilisée est la contrainte asymétrique d'Itakura — du fait que la détermination de $D^k(i,j)$ ne nécessite que les quantités $D^k(i-1,j)$, $D^k(i-1, j-1)$, $D^k(i-1, j-2)$ et $d^k(i,j)$, l'évaluation des distances cumulées à l'étape i pourra être engendrées à partir des distances calculées à l'étape $i-1$.

Il vient donc que l'algorithme de Bridle et de Nakagawa peut fonctionner en une seule passe et en temps réel.

VI.6.4.2.- Spécification de l'algorithme.

En posant :

$$D^k(i,j) = \underset{h}{\text{Min}} \left(\hat{D}^B(h) + D\left(T(h,i), R_k\ [1,j]\right)\right) \tag{7.31}$$

et

$$F^k(i,j) = \underset{h}{\text{Arg min}} \left(\hat{D}^B(h) + D\left(T(h,i), R_k\ [1,j]\right)\right) \tag{7.32}$$

l'algorithme de Bridle et de Nakagawa peut être spécifié ainsi :

Algorithme 6 :

a) Initialisation générale :

1- $\hat{D}^B(0) = 0$, $\hat{F}^B(0) = 0$

- $D^r(0,0) = 0$, $F^r(0,0) = 0$ pour $r = 1,2,\ldots, N$

- $D^r(0,j) = \infty$, $F^r(0,j) = 0$ pour $r = 1,2,\ldots, N$
et pour $j = 1,2,\ldots, J_r$

b) Programmation dynamique :

2- Pour i de 1 à I Faire de 3 à 6

3- Pour r de 1 à N Faire de 4 à 5

4- . $D^r(i-1,0) = \hat{D}^B(i-1)$

. $F^r(i,0) = i$

5- . Pour $1 \leqslant j \leqslant J_r$ Faire

.. $\hat{j} = \underset{MAX(0,j-2) \leqslant j' \leqslant j}{\text{Arg min}} D^r(i-1,j')$

.. $D^r(i,j) = D^r(i-1,\hat{j}) + d^r(i,j)$

.. $F^r(i,j) = F^r(i-1,\hat{j})$

. FIN DE FAIRE

6- . Pour i de 1 à N FAIRE

. $\hat{r} = \underset{r}{\text{Arg min}} D^r(i,J_r)$

. $\hat{D}^B(i) = D^r(i,J_{\hat{r}})$

. $\hat{N}^B(i) = \hat{r}$

. $\hat{F}^B(i) = F^{\hat{r}}(i,J_{\hat{r}})$

. FIN DE FAIRE

c) Décision

7- $\tilde{n} = 1$

- $\tilde{p}(\tilde{n}) = I$

- Booléen = vrai

8- Tant que Booléen Faire

. Si $\hat{F}^B(\tilde{p}(\tilde{n})) = 0$

alors Booléen = Faux

sinon $\tilde{n} = \tilde{n} + 1$

$\tilde{p}(\tilde{n}) = \hat{F}^B(\tilde{p}(\tilde{n} - 1))$ FIN DE SI

- FIN DE FAIRE , FIN DE TANT QUE

9- $\hat{n} = \tilde{n}$

10- $\hat{p}_{\hat{n}}(n) = \tilde{p}\ (\hat{n} - n + 1)$ pour $n = 1,2,\ldots, \hat{n}$

11- $\hat{q}(k) = \hat{N}^B(\hat{p}_{\hat{n}}(k))$ pour $k = 1,2,\ldots, \hat{n}$

12- $SR^{\hat{q}}_{\hat{n}} = R_{\hat{q}(1)} \oplus R_{\hat{q}(2)} \oplus \ldots \oplus R_{\hat{q}(\hat{n})}$

VI.6.4.3.- Performance de l'algorithme.

Du fait que l'algorithme de Bridle et de Nakagawa parcourt la phrase à reconnaître en une seule passe, le nombre de distances locales évaluées est donné par :

(7.33) NDL = I.N.J.

et en reprenant pour I, N et J les valeurs que l'on avait considérées pour les algorithmes de Sakoe et de Myers il vient que NDL = 6 400.

L'algorithme de Bridle et de Nakagawa, ainsi, est le plus performant des trois algorithmes que nous venons de décrire. De plus, étant donné qu'il n'effectue pas de retours arrière, il est tout à fait adapté au temps réel.

VI.6.5.- Quelques remarques générales concernant les algorithmes décrits.

Les différents algorithmes de reconnaissance de mots enchaînés que nous venons d'expliciter ont tous été déduits à partir du modèle de Sakoe et plus particulièrement à partir de la relation (7.5). On peut donc faire les assertions suivantes :

-1- Les trois algorithmes proposés sont fondamentalement équivalents en ce sens qu'ils fournissent des résultats identiques.

-2- La contrainte locale utilisée dans l'algorithme de programmation dynamique du niveau mot doit obligatoirement être asymétrique pour respecter les hypothèses qui ont permis d'établir (7.5).

-3- Le problème de la coarticulation n'est pas pris en considération par les trois algorithmes par définition du modèle de Sakoe.

VI.7.- RELACHEMENT DES CONTRAINTES AUX FRONTIERES DANS LES ALGORITHMES DE RECONNAISSANCE DE MOTS ENCHAINES.

VI.7.1.- Introduction

Les algorithmes de reconnaissance de mots enchaînés fondés à partir de la relation (7.5) ne peuvent, du fait qu'ils sont obligés d'utiliser une contrainte locale asymétrique, relacher les contraintes aux frontières que sur l'axe vertical.

Afin de pouvoir effectuer une relaxation de ces dernières suivant les deux axes nous avons été amenés à généraliser les relations de la reconnaissance de mots enchaînés.

VI.7.2.- Généralisation du formalisme de la reconnaissance de mots enchaînés.

Pour utiliser une contrainte locale symétrique ou pour relacher les contraintes aux frontières dans un algorithme de reconnaissance de mots enchaînés, il est impératif de tenir compte des longueurs des chemins de recalage dans la métrique définissant le taux de dissemblance entre la forme inconnue et une super-forme de référence quelconque, et par conséquent de formaliser la solution du problème à l'aide de la relation :

$$(7.34) \qquad D\left(T, SR_n^q\right) = \underset{n, p_n(k), q(k)}{\text{Min}} \frac{1}{N(p_n, q)} \sum_{k=1}^{n} D^*\left(T\left(p_n(k-1), p_n(k)\right), R_{q(k)}\right)$$

où le facteur $\frac{1}{N(p_n,q)}$ permet de rendre la fonctionnelle

$$(7.35) \qquad D^*_{p_n}\left(T, SR_n^q\right) = \sum_{k=1}^{n} D^*\left(T\left(p_n(k-1), p_n(k)\right), R_{q(k)}\right)$$

indépendante de la longueur du chemin de recalage.

La résolution de l'équation (7.34) nécessite une stratégie identique à celle mise en oeuvre dans le cas de la reconnaissance de mots isolés et qui consiste à effectuer dans une première phase les normalisations des scores des chemins de recalage aboutissant en un

point de l'espace de comparaison par leur longueur afin, d'une part, de connaître le chemin optimal aboutissant en ce point et d'autre part, de pouvoir déterminer dans une deuxième phase le score associé à celui-ci. Dans le problème de la reconnaissance de mots isolés, les longueurs des chemins de recalage ont pu être déterminées aisément en fonction de la pondération de la contrainte locale utilisée, car la minimisation de fonctionnelle spécifiant la solution du problème ne considérait qu'une seule forme de référence. Ici il en va de bien-entendu différemment si bien que l'évaluation de $N(p_n,q)$ est moins évidente et doit faire l'objet de quelques développements indispensables.

VI.7.3.- Détermination des longueurs des chemins de recalage dans le cas de la reconnaissance de mots enchaînés.

Si l'on adopte le formalisme de Myers - nous aurions pu prendre celui de Bridle et de Nakagawa mais cela aurait, selon nous, nuit à la clarté de l'exposé - l'espace de comparaison peut être décomposé en LMAX sous-espaces, comme l'indique la figure 43, et un point de cet espace peut être représenté par quatre composantes :
i, j, r et n désignant respectivement :

- le $i^{ème}$ prélèvement de la forme inconnue,
- le $j^{ème}$ prélèvement d'une forme de référence,
- la forme de référence considérée,
- le niveau de construction du chemin optimal de comparaison.

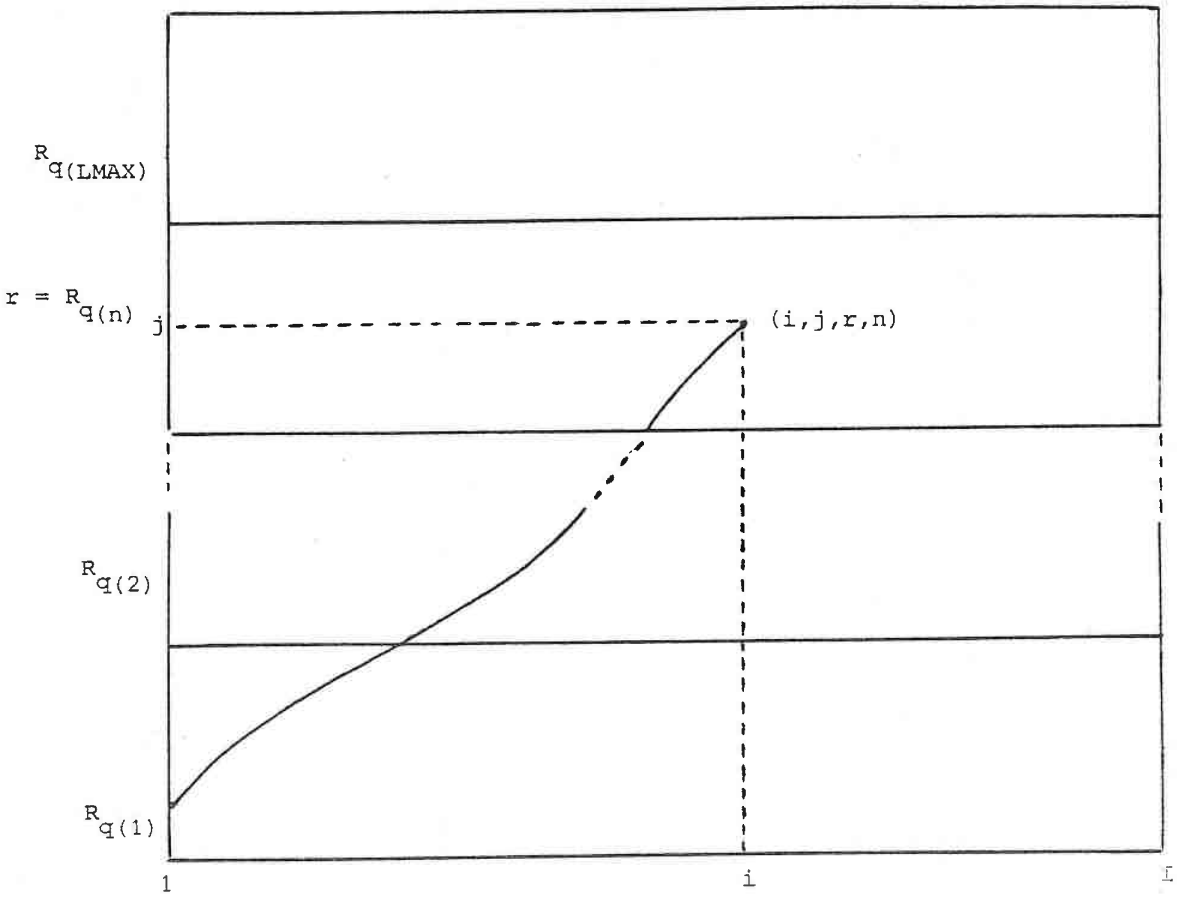


Figure 43 : L'espace de comparaison de MYERS.

Désignons par $P_H(i,j,r,n)$ et $P_V(i,j,r,n)$ respectivement l'abscisse et l'ordonnée de l'origine du chemin optimal aboutissant au point (i,j,r,n) et par $N(i,j,r,n,m)$ la fonction qui, pour tout $m < n$ - $n \neq 1$ - donne la forme de référence optimale trouvée au niveau m, qui a permis de construire le chemin optimal de comparaison aboutissant au point (i,j,r,n) et qui, lorsque $m = n$, retourne la forme de référence r. Si l'on se place dans le cas d'une contrainte locale asymétrique, la longueur d'un chemin de recalage quelconque aboutissant au point (i,j,r,n) ne dépend que de i et de l'abscisse de l'origine du chemin en question. Et si l'on s'intéresse parmi l'ensemble des chemins de recalage aboutissant au point (i,j,r,n) uniquement à celui qui contient le chemin optimal de recalage ayant pour extrémité le point (i',j',r,n) où (i',j',r,n) appartient à un voisinage - défini par la contrainte locale utilisée - alors la longueur de ce dernier est donnée par :

$$(7.35) \qquad L\left(C^{(i,j,r,n)}_{(P_H(i',j',r,n)\ ,\ P_V(i',j',r,n),\ N(i',j',r,n,1),\ 1)} \right)$$

$$= i - P_H(i',j',r,n) + 1 \quad .$$

Du fait que (7.34) ne fait pas intervenir les longueurs des formes de référence, la technique de normalisation des distances cumulées est, dans ce cas, exactement identique à celle que nous avons explicitée dans le cadre de la reconnaissance de mots isolés.

Si l'on se place maintenant dans le cas d'une contrainte locale symétrique, pour connaître la longueur du chemin optimal aboutissant en un point (i',j',r,n) appartenant à un voisinage du point (i,j,r,n) - pour pouvoir déterminer ensuite la longueur du chemin de recalage aboutissant au point (i,j,r,n) et construit à partir du chemin optimal passant par le point (i',j',r,n) - le problème essentiel consiste à

évaluer la somme des longueurs utiles des formes de référence optimales trouvées aux niveaux n-1, n-2,..., 1 qui ont permis de construire le chemin de comparaison ayant pour extrémité le point (i',j',r,n) . Nous entendons par longueur utile d'une forme de référence le nombre de prélèvements qui ont été effectivement considérés lors de la comparaison. Nous avons été amenés à formuler cette notion car en permettant à l'algorithme de ne pas tenir compte de certains prélèvements de début ou de fin de formes et référence, il est possible de compenser, dans une certaine mesure, les effets dus à la coarticulation. Désignons par $P_{HL}(i,j,r,n)$, $P_{VL}(i,j,r,n)$ et $\tilde{j}(i',r,n)$ les fonctions visualisées par la figure 44, donnant respectivement :

- l'abscisse de l'origine du sous-chemin optimal du niveau n aboutissant au point (i,j,r,n) .
- l'ordonnée de l'origine du sous-chemin optimal du niveau n aboutissant au point (i,j,r,n) .
- l'ordonnée de l'extrémité du sous-chemin optimal d'abscisse i' .

et par $\hat{\mathcal{L}}_n^B(i)$ la fonction qui fournit au prélèvement i , lorsque le niveau n a été évalué la somme des longueurs utiles des formes de référence. Grâce à P_{HL} , P_{VL} et $\tilde{j}$ il est possible d'évaluer cette dernière récursivement :

$$(7.36)\quad \hat{\mathcal{L}}_n^B(i) = \hat{\mathcal{L}}_{n-1}^B\left(P_{HL}\left(i, \tilde{j}\left(i, \hat{N}_n^B(i), n\right), \hat{N}_n^B(i), n\right)\right) +$$

$$\tilde{j}\left(i, \hat{N}_n^B(i), n\right) - P_{VL}\left(i, \tilde{j}\left(i, \hat{N}_n^B(i), n\right), \hat{N}_n^B(i), n\right) + 1$$

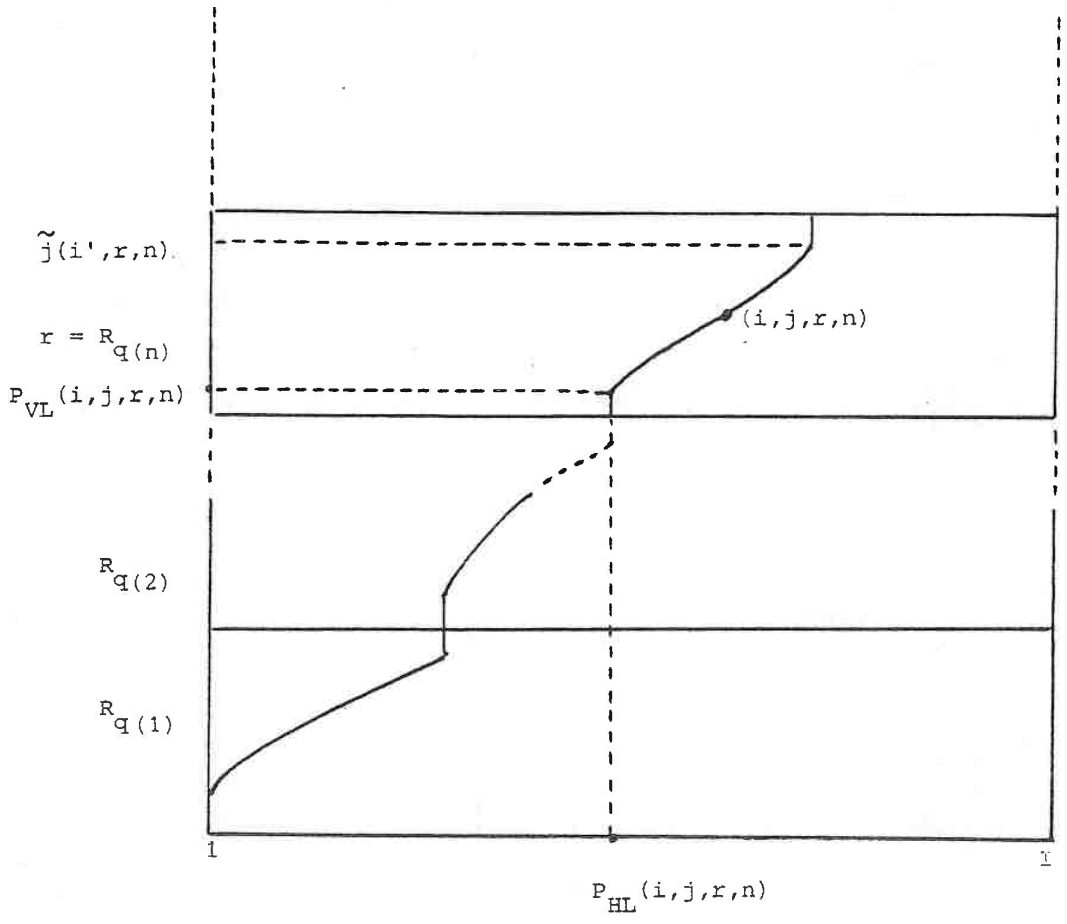


Figure 44 : Visualisation des fonctions $P_{HL}(i,j,r,n)$, $P_{VL}(i,j,r,n)$ et $\tilde{j}(i',r,n)$

où : $\hat{\mathcal{L}}^{B}_{n-1}\left(P_H\left(i,\tilde{j}\left(i,\hat{N}^{B}_{n}(i),n\right),\hat{N}^{B}_{n}(i),n\right)\right)$ est la somme des longueurs utiles des formes de référence optimales trouvées aux niveaux $n-1, n-2, \ldots, 1$ qui ont permis de construire le chemin optimal ayant pour extrémité le point

$$\left(i, \tilde{j}\left(i, \hat{N}^{B}_{n}(i), n\right), \hat{N}^{B}_{n}(i), n\right)$$

et

$$\tilde{j}\left(i, \hat{N}^{B}_{n}(i), n\right) - P_{VL}\left(i,\tilde{j}\left(i, \hat{N}^{B}_{n}(i), n\right), \hat{N}^{B}_{n}(i), n\right) + 1$$

est le longueur du sous-chemin optimal au niveau n aboutissant au point

$$\left(i, \tilde{j}\left(i, \hat{N}^{B}_{n}(i), n\right), \hat{N}^{B}_{n}(i), n\right) .$$

La somme des longueurs utiles des formes de référence optimales, qui ont permis de construire le chemin optimal d'extrémité $\left(i, \tilde{j}\left(i, \hat{N}^{B}_{n}(i), n\right), \hat{N}^{B}_{n}(i), n\right)$, ayant été déterminée par (7.36) il nous est possible désormais d'évaluer la longueur du chemin de recalage optimal aboutissant au point (i',j',r,n) :

$$\begin{aligned} (7.37) \qquad L\left(C^{(i',j',r,n)}_{(P_H(i',j',r,n),\, P_V(i',j',r,n),\, N(i',j',r,n,1),\, 1)}\right) &= \hat{\mathcal{L}}^{B}_{n-1}\left(P_{HL}(i',j',r,n)\right) \\ &+ j' - P_{VL}(i',j',r,n) + 1 \\ &+ i' - P_H(i',j',r,n) + 1 \end{aligned}$$

et par voie de conséquence la longueur du chemin de recalage, construit à partir du chemin optimal précédent, d'extrémité le point (i,j,r,n) :

$$(7.38) \qquad L\left(C^{(i,j,r,n)}_{(P_H(i',j',r,n),\ P_V(i'j',r,n),\ N(i',j',r,n,1),\ 1)}\right)$$

$$= \hat{\mathcal{L}}^{B}_{n-1}\left(P_{HL}(i',j',r,n)\right)$$

$$+ j - P_{VL}(i',j',r,n) + 1$$

$$+ i - P_{H}(i',j',r,n) + 1 \quad .$$

Grâce à (7.35) et (7.38) les longueurs des chemins de recalage aboutissant en un point quelconque de l'espace de comparaison, pour une contrainte locale symétrique ou asymétrique, avec ou sans relaxation des contraintes aux frontières peuvent être déterminées. Par conséquent, les techniques de normalisation que nous avons employées dans le cas de la reconnaissance de mots isolés s'appliquent tout aussi bien à la reconnaissance de mots enchaînés.

VI.7.4.- Généralisation de l'algorithme de MYERS permettant un relachement des contraintes aux frontières.

VI.7.4.1.- Spécification de l'algorithme

Afin d'illustrer le dernier point du paragraphe précédent voici comment nous avons généralisé l'algorithme de MYERS dans le cas d'une contrainte locale symétrique avec relachement de contraintes aux frontières :

Algorithme 7 :

(* pour harmoniser les nouvelles notations que nous venons d'introduire avec celles utilisées dans l'algorithme 5, nous faisons les réécritures suivantes :

$P_H(i,j,r,n) \longrightarrow P^r_{H_n}(i,j)$

$P_{HL}(i,j,r,n) \longrightarrow P^r_{HL_n}(i,j)$

$P_{VL}(i,j,r,n) \longrightarrow P^r_{VL_n}(i,j)$

$\tilde{j}(i,r,n) \longrightarrow \tilde{j}^r_n(i)$ *)

a) Initialisation :

1- $\hat{D}^B_0(i) = 0$ pour $1 \leqslant i \leqslant \alpha$

(* α définit la plage de départ des chemins sur l'axe horizontal *)

- $\hat{D}^B_0(i) = \infty$ pour $\alpha < i \leqslant I$

- $\hat{\mathcal{L}}^B_0(i) = 0$ pour $1 \leqslant i \leqslant \alpha$

- * Pour $1 \leqslant r \leqslant N$ et Pour $0 \leqslant j \leqslant \beta(r)$ Faire

(* $\beta(r) - 1$ est le nombre de prélèvements de début de la forme r pouvant ne pas être pris en considération lors de la comparaison *)

. $D^{r}_{PP_1}(0, j) = 0$

. $P^{r}_{H_1}(0, j) = 1$

. $P^{r}_{VL_1}(0, j) = j$

* FIN DE FAIRE

b) Programmation dynamique :

2- Pour $1 \leqslant n \leqslant LMAX$ FAIRE de 3 à 6

3- Pour $1 \leqslant r \leqslant N$ Faire de 4 à 5

4- Initialisation du niveau n

Pour $1 \leqslant i \leqslant I$ Faire

. $D^{r}_{PP_n}(i, 0) = \hat{D}^{B}_{n-1}(i)$

. $P^{r}_{HL_n}(i, 0) = i$

. $P^{r}_{VL_n}(i, 0) = 1$

. $\tilde{d}^{r}_{n}(i) = \infty$

$\Big(* \ \tilde{d}^{r}_{n}(i)$ est la distance normalisée associée à

$D^{r}_{PP_n}\Big(i, \tilde{j}^{r}_{n}(i)\Big) \qquad *\Big)$

FIN DE FAIRE

5- Comparaison dynamique du niveau n .

- Pour $1 \leqslant i \leqslant I$ et Pour $IMIN_n(i) \leqslant j \leqslant IMAX_n(i)$

Faire :

. $d^{r}_{n}(i,j) = \underset{\substack{(i',j') \in V(i,j) \\ (i') \leqslant j' \leqslant IMAX_n(i')}}{Min}$

$$\frac{D^{r}_{PP_n}(i',j') + d^{r}_{P_n}\Big((i',j'),\ (i,j)\Big)}{\hat{\mathcal{L}}^{B}_{n-1}\Big(P^{r}_{HL_n}(i',j')\Big) + j - P^{r}_{VL_n}(i',j') + 1 + i - P^{r}_{H_n}(i',j') + 1}$$

$\Big(* - d^{r}_{P_n}\Big((i',j'),\ (i,j)\Big)$ est la distance locale pondérée entre le point (i,j',r,n) et le point (i,j,r,n) .

- $d^{r}_{n}(i,j)$ est la distance normalisée associée au chemin optimal aboutissant au point (i,j,r,n) dans le cas où les prélèvement de la forme r d'indice inférieur à j sont pris en considération

- au premier niveau : si $i = 1$ ou $j = 1$

alors $V(i,j) = \{(i-1,j-1)\}$ et

$d^{r}_{P_n}\Big((i',j'),\ (i,j)\Big) = 2 * d^{r}(i,j) \qquad *\Big)$

. $(i',j') = \underset{\substack{(i',j') \in V(i,j) \\ IMIN_n(i') \leq j' \leq IMAX_n(i')}}{\text{Arg min}}$

$$\frac{D^r_{PP_n}(i',j') + d^r_{p_n}\left((i',j'),\,(i,j)\right)}{\hat{\mathcal{L}}^B_{n-1}\left(P^r_{HL_n}(i',j')\right) + j - P^r_{VL_n}(i',j') + 1 + i - P^r_{H_n}(i',j') + 1}$$

. Si $j < \beta(r)$

.. si $n = 1$ alors $d'^r_n(i,j) = 2 * d^r(i,j)$

.. sinon

$$d'^r_n(i,j) = \frac{\hat{D}^B_{n-1}(i) + d^r(i,j)}{\hat{\mathcal{L}}^B_{n-1}(i) + 1 + i - P_{H_{n-1}}\left(i, \tilde{j}^{\hat{N}^B_{n-1}(i)}_{\hat{N}^B_{n-1}(i)}\right) + 1}$$

.. FIN de SI

($* d'^r_n(i,j)$ est la distance normalisée associée au chemin optimal aboutissant au point (i,j,r,n) dans le cas où les prélèvements de la forme r d'indice inférieur à j ne sont pas pris en considération $*$)

.. si $d'^r_n(i,j) < d^r_n(i,j)$ alors

... si $n = 1$ alors

.... $D^r_{PP_n}(i,j) = 2 * d^r(i,j)$.

.... $P^r_{H_n}(i,j) = i$

.... $P^r_{HL_n}(i,j) = i$

.... $P^r_{VL_n}(i,j) = j$

... FIN de si

.. sinon

... $D^r_{PP_n}(i,j) = D^r_{PP_n}(i',j') + d^r_{P_n}\Big((i',j'), (i,j)\Big)$

... $P^r_{H_n}(i,j) = P^r_{H_n}(i',j')$

... $P^r_{VL_n}(i,j) = P^r_{VL_n}(i',j')$

.. FIN de SI .

. Si $j > \beta(r)$ alors

.. $D^r_{PP_n}(i,j) = D^r_{PP_n}(i',j') + d^r_{P_n}\Big((i',j'), (i,j)\Big)$

.. $P^r_{H_n}(i,j) = P^r_{H_n}(i',j')$

.. $P^r_{HL_n}(i,j) = P^r_{HL_n}(i',j')$

.. $P^r_{VL_n}(i,j) = P^r_{VL_n}(i',j')$

. FIN DE SI , FIN DE FAIRE

- * Pour $1 \leqslant i \leqslant I$ et Pour $J_r - \delta(r) \leqslant j \leqslant J_r$

$IMIN_n(i) \leqslant j \leqslant IMAX_n(i)$

FAIRE :

(* $\delta(r)-1$ est le nombre de prélèvements de fin de la forme r pouvant ne pas être pris en considération *)

. $\tilde{j}^r_n(i) = \underset{j}{\text{Arg min}}$

$$\frac{D^r_{PP_n}(i,j)}{\hat{\mathcal{L}}^B_{n-1}\left(P^r_{HL_n}(i,j)\right) + j - P^r_{VL_n}(i,j) + 1 + i - P^r_{H_n}(i,j) + 1}$$

. $\tilde{\ell}^r_n(i) = \tilde{j}^r_n(i) - P^r_{VL_n}(i, \tilde{j}^r_n(i)) + 1$

(* $\tilde{\ell}^r_n(i)$ est la longueur utile de la forme de référence r qui a permis de construire le sous-chemin optimal du niveau n d'extrémité $(i, \tilde{j}^r_n(i), r, n)$ *)

. $\tilde{d}^r_n(i) =$

$$\frac{D^r_{PP_n}(i, \tilde{j}^r_n(i))}{\hat{\mathcal{L}}^B_{n-1}\left(P^r_{HL_n}\left(i,\tilde{j}^r_n(i)\right)\right) + j - P^r_{VL_n}\left(i,\tilde{j}^r_n(i)\right) + 1 + i - P^r_{H_n}\left(i,\tilde{j}^r_n(i)\right) + 1}$$

. · FIN de Faire

6- Minimisation du niveau phrase

- * Pour $1 \leqslant i \leqslant I$ Faire

. $\hat{d}^B_n(i) = \underset{r}{\text{MIN}}\ \tilde{d}^r_n(i)$

. si $\hat{d}^B_n(i) \neq \infty$ alors

.. $\hat{N}^B_n(i) = \underset{r}{\text{Arg min}}\ \tilde{d}^r_n(i)$

.. $\hat{\mathcal{L}}^B_n(i) = \hat{\mathcal{L}}^B_{n-1}\left(P_{HL_n}^{\hat{N}^B_n(i)}\left(i, \tilde{j}_n^{\hat{N}^B_n(i)}(i)\right)\right) + \ell_n^{\hat{N}^B_n(i)}(i)$

.. $\hat{F}^B_n(i) = P_{HL_n}^{\hat{N}^B_n(i)}\left(i, j_n^{\hat{N}^B_n(i)}(i)\right)$

.. $\hat{D}^B_n(i) = D_{PP_n}^{\hat{N}^B_n(i)}\left(i, \tilde{j}_n^{\hat{N}^B_n(i)}(i)\right)$

. FIN de Faire

c) Décision

7- $\hat{n} = \underset{n}{\text{Arg min}} \underset{I-\gamma<i<I}{\text{Min}} \hat{d}^B_n(i)$

(* γ-1 est le nombre de prélèvement pouvant ne pas être pris en considération en fin de forme inconnue *)

8- $\hat{i} = \underset{i}{\text{Arg}}\ \underset{n}{\text{Min}} \underset{I-\gamma<i<I}{\text{Min}} \hat{d}^B_n(i)$

9- $P_{\hat{n}}(\hat{n}) = \hat{i}$

10- $p_{\hat{n}}(n) = \hat{F}^B_{n+1}\left(p_{\hat{n}}(n)\right)$ pour $n = \hat{n}-1, \ldots, 1$

11- $\hat{q}(k) = \hat{N}^B_k\left(p_{\hat{n}}(k)\right)$ pour $k = 1, 2, \ldots, n$

12- $SR^{\hat{q}}_{\hat{n}} = R_{\hat{q}}(1) \oplus R_{\hat{q}}(2) \oplus \ldots \oplus R_{\hat{q}}(n)$

VI.7.4.2.- Complexité de l'algorithme.

Le nouvel algorithme que nous venons de présenter, du fait qu'il réalise la normalisation des distances cumulées des chemins de recalage par leur longueur, doit effectuer en tout point de l'espace de comparaison n divisions, où n est le nombre de chemins locaux définis par la contrainte locale utilisée - n est égal à 3 pour toutes les contraintes que nous avons considérées -. Si l'on se place dans le cas particulier que nous avons envisagé pour déterminer la complexité des algorithmes de reconnaissance de mots enchaînés que nous avons décrits précédemment, il vient que notre algorithme nécessite l'évaluation de :

- 10 666 distances locales et
- 31 998 divisions.

Ainsi la charge de calcul supplémentaire qu'impose un relachement des contraintes aux frontières, tout en n'étant pas prohibitive, est tout de même non négligeable. L'utilisation de circuits VLSI spécialisés, afin de permettre une implémentation efficace de notre algorithme s'avère donc nécessaire.

VI.8.- PROGRAMMATION DYNAMIQUE ET COARTICULATION

VI.8.1.- Introduction

Le modèle de reconnaissance de mots enchaînés de Sakoe construit les super-formes qui sont comparées à la forme inconnue par simple concaténation de formes de référence qui ont été obtenues par élocution isolée. Ce modèle suppose donc implicitement que le coarticulation n'entraîne pas de modifications importantes des formes en contexte par rapport aux références. Or, malheureusement, en pratique il en est tout autrement : en effet il arrive fréquemment que les zones de début ou de fin de certains mots du vocabulaire soient modifiées de façon considérable

lorsque ceux-ci sont prononcés dans un continuum de parole. Ce qui provoque dans ces cas là, si les algorithmes de reconnaissance de mots enchaînés ne prévoient pas l'apparition de ce genre de phénomènes, des erreurs de reconnaissance. Aussi est-il indispensable, afin que la performance et la robustesse des systèmes de reconnaissance de mots enchaînés soient accrues, que les algorithmes puissent compenser les distorsions coarticulatoires.

VI.8.2.- Généralisation du modèle de Sakoe.

Pour tenir compte de la coarticulation nous avons été amenés à distinguer pour toute forme de référence R_i trois zones : une zone de début et une zone de fin de mot que nous notons respectivement $D(i)$ et $F(i)$ susceptibles de subir des distorsions coarticulatoires et une zone de milieu de mot $M(i)$ en général peu modifiable par coarticulation. Une forme R_i, ainsi, peut-être considérée comme étant constituée par la concaténation de $D(i)$, $M(i)$ et $F(i)$ soit :

$$R_i = D(i) \oplus M(i) \oplus F(i) \quad . \tag{7.39}$$

Dans le modèle de reconnaissance de mots enchaînés que nous proposons, qui est visualisé par la figure 45, $D(i)$ et $F(i)$ peuvent être substituées au cours de la comparaison respectivement par les formes $CAR(i)$ - CAR comme coarticulation arrière - et $CAV(i)$ - CAV comme coarticulation avant - représentant les formes $D(i)$ et $F(i)$ modifiées par des effets dues à la coarticulation. Du fait qu'à chacune des zones $D(i)$ et $F(i)$ nous n'associons qu'une seule forme de déformation nous supposons que les distorsions que peuvent subir $D(i)$ et $F(i)$ ne varient pas de façon significative en fonction du contexte. Cette hypothèse - comme nous avons pu la vérifier expérimentalement - est tout à fait réaliste dans le cadre d'une reconnaissance de mots enchaînés à vocabulaire limité. Par contre, pour de grands

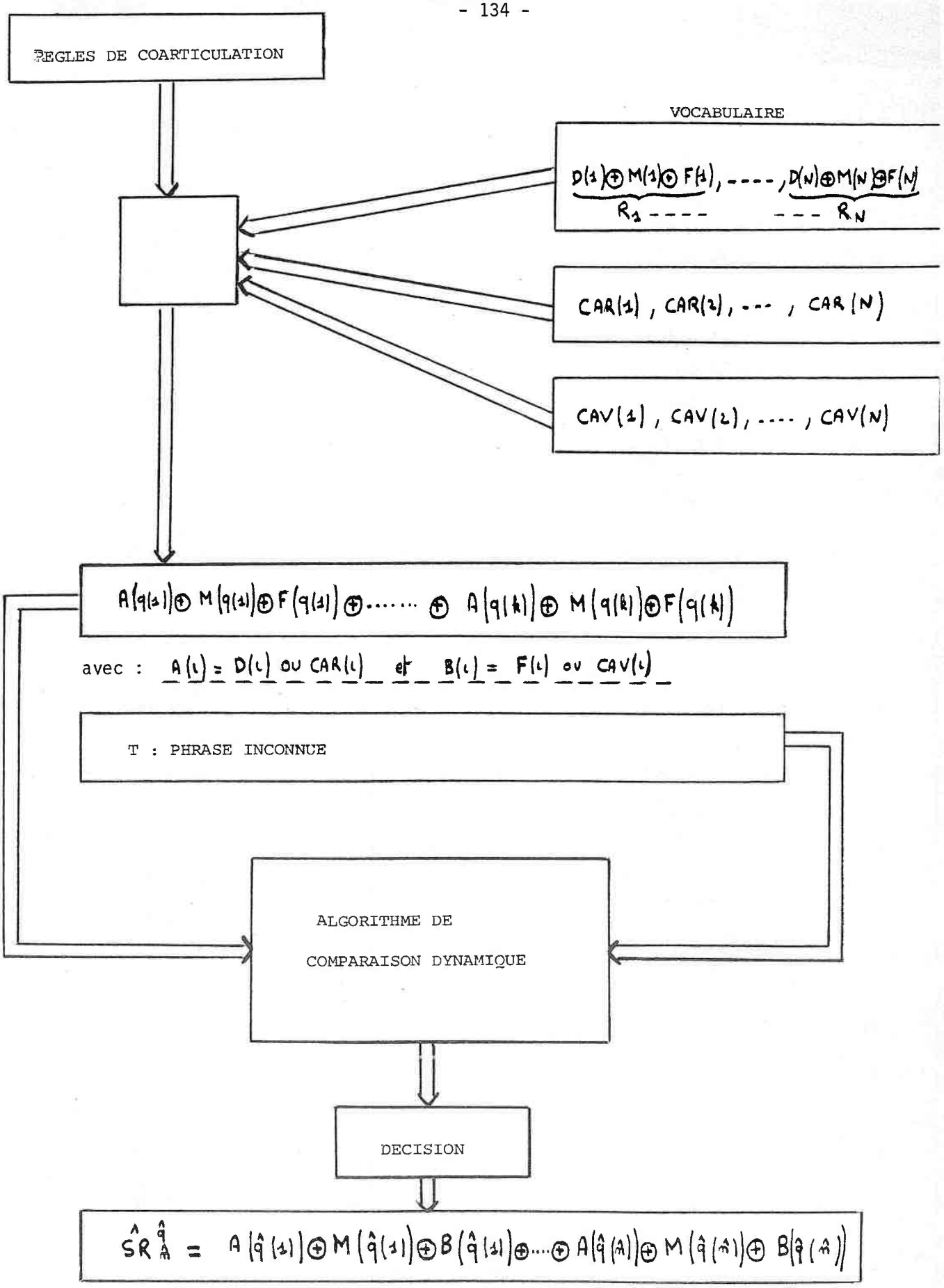


Figure 45 : Un modèle de reconnaissance de mots enchaînés tenant compte de de la coarticulation.

vocabulaires, nous pensons qu'une telle hypothèse ne serait pas appropriée et qu'il faudrait alors avoir autant de formes de déformation qu'il y a de types d'effets de coarticulation.

En fait dans le modèle que nous avons adopté les substitutions de début ou de fin de formes de référence n'ont pas lieu de façon arbitraire. En effet, s'il en était ainsi, l'espace des solutions dans lequel serait recherchée la super-forme de référence optimale serait trop vaste et de nombreuses confusions ne pourraient alors être évitées. Pour restreindre ce dernier nous avons introduit dans le modèle des règles de coarticulation qui gèrent l'opérateur de concaténation en fonction du contexte déterminé par la comparaison pour interdire les substitutions inopportunes.

Les règles que nous avons considérées ont été déduites de la visualisation de nombreux spectres de phrases comportant exclusivement des chiffres. A partir de cette observation nous avons pu mettre en évidence que les distorsions coarticulatoires apparaissant - dans les conditions expérimentales dans lesquelles nous nous sommes placés - principalement dans les cas suivants :

-1) lors de la prononciation de deux voyelles consécutives.

-2°) lorsqu'une voyelle nasale est suivie d'une consonne nasale ou d'une plosive voisée comme dans les phrases "UN NEUF" /œ̃$^{(n)}$ n œ F/ ou "UN DEUX" /œ̃$^{(n)}$ d Ø/

-3°) lorsqu'une fricative sourde est suivie d'une occlusive sonore comme dans "SIX DEUX"

- /SIS, d Ø/ / SIZ dØ / -

-4°) lorsqu'une plosive sourde précédée par une voyelle nasale est suivie d'une occlusive sonore comme dans

"CINQ DEUX" - /sɛ̃K, dØ/ / sɛ̃g dØ/ -

Le premier cas de coarticulation que nous avons noté ne présente pas de difficultés majeures car il peut être généralement compensé en ne prenant pas en considération certains prélèvements en début ou en fin de voyelle. C'est à cause de ce type de coarticulation que nous nous sommes donné la possibilité de ne pas tenir compte dans la comparaison dynamique, explicitée par l'algorithme 7 du paragraphe VI.7.3, de certains prélèvements situés aux extrémités des formes de référence. Le deuxième cas est l'exemple typique de ce que nous appelons une coarticulation arrière car il s'agit du début du mot contenant la consonne nasale ou la plosive voisée qui est déformé. Le troisième et le quatrième cas sont des exemples contraires du cas précédent car il mettent en évidence une coarticulation avant dont l'effet a lieu sur la fin du mot précédent l'occlusive sonore.

A partir de ces considérations nous avons déduit les règles suivantes :

Règle 1 : une coarticulation arrière peut affecter une forme si celle-ci, d'une part, débute par une plosive voisée ou par une consonne nasale et si, d'autre part, elle est précédée par une voyelle nasale.

Règle 2 : une coarticulation avant peut affecter une forme si celle-ci, d'une part, se termine par une fricative sourde ou par une plosive non voisée précédée par une voyelle nasale et si, d'autre part, elle est suivie d'une occlusive sonore.

VI.8.3.- Détermination expérimentale des formes D(i), F(i), CAR(i) et CAV(i).

Dans le système que nous avons réalisé les zones de début ou de fin de mots susceptibles de subir des distorsions coarticulatoires sont définies manuellement lors d'une première phase d'apprentissage. Lors de celle-ci, plus précisément, nous effectuons pour chaque forme de référence, d'une part, le rangement en mémoire du spectre numérique la caractérisant et, d'autre part, au vu du spectre transcodé nous fournissons du système de reconnaissance quatre paramètres : deux indices pour définir le début et la fin de la zone D(i) et deux autres pour définir la zone F(i) .

Afin de créer les formes CAR(i) et CAV(i) nous prononçons, au cours d'une deuxième phase d'apprentissage, la forme R(i) en contexte, un premier contexte, s'il y a lieu, pour réaliser une déformation due à une coarticulation arrière et un deuxième contexte, éventuellement, pour engendrer une déformation due à une coarticulation avant. Dans les deux cas en visualisant le spectre obtenu, nous extrayons un prélèvement significatif de la zone où s'est produit une distorsion coarticulatoire. Ce dernier est transmis à CAR(i) s'il s'agit d'une coarticulation arrière et à CAV(i) s'il s'agit d'une coarticulation avant. Nous avons décidé de définir les formes CAR(i) et CAV(i) comme étant constituées par une succession virtuelle de prélèvements identiques, d'une part, afin de ne pas augmenter de façon considérable la complexité de l'algorithme de comparaison dynamique et, d'autre part, car nous avons constaté expérimentalement que les régions ayant subi des déformations coarticulatoires sont caractérisées par de légères variations spectrales.

La procédure que nous utilisons pour définir les sous-formes D(i), F(i), CAR(i) et CAV(i) est ainsi essentiellement manuelle et nécessite une expérience certaine de la part de l'utilisateur dans la lecture des spectres.

Aussi elle ne peut être raisonnablement réalisée par une personne peu experte en traitement de la parole.

Dans une extension future de notre programme de reconnaissance de mots enchaînés nous nous proposons d'automatiser cette phase d'apprentissage afin que celui-ci puisse être abordé par n'importe quel utilisateur.

VI.8.4.- Introduction des règles de coarticulation dans la comparaison dynamique.

VI.8.4.1.- Principe de la méthode

En utilisant le formalisme de Myers, le principe de la procédure que nous utilisons pour mettre en oeuvre les règles de coarticulation dans la comparaison dynamique est le suivant :

-1)- l'algorithme de programmation dynamique décide dans un premier temps si au point (i,j,R_k, n) il y a lieu de tenir compte d'un effet de coarticulation en vérifiant que le couple de formes constitué par la dernière forme de référence de la meilleure super-forme de $n-1$ mots qui a permis de construire le chemin optimal aboutissant au point (i,j, R_k, n) et la forme R_k unifie une règle de coarticulation -arrière ou avant -.

-2)- si la décision précédente est affirmative, l'algorithme de programmation dynamique valide ou non dans un deuxième temps l'hypothèse de coarticulation au point (i,j, R_k, n) en associant au chemin optimal aboutissant en ce point le minimum des distances cumulées que l'on obtient en tenant compte ou pas d'un effet dû à la coarticulation.

VI.8.4.2.- Limitations des substitutions par les règles de coarticulation arrière.

Le déclenchement des règles de coarticulation arrière est somme toute aisé dans la mesure où en un point quelconque (i,j, R_k, n) de l'espace de comparaison, grâce aux tables $\hat{N}^B_n(i)$ et $P_{HL}(i,j,r,n)$, il est possible de déterminer la dernière forme de référence, $R_{n-1}(i,j, R_k, n)$, qui a permis de construire le sous-chemin du niveau $n-1$ inclus dans le chemin optimal aboutissant au point (i,j, R_k, n) :

$$R_{n-1}(i,j, R_k, n) = N^B_{n-1}\left(P_{HL}(i,j, R_k, n)\right) \qquad (7.40)$$

De ce fait aucun problème ne se pose pour décider si au point (i,j, R_k, n) il y a lieu de tenir compte d'un effet de coarticulation arrière. Il suffit, pour cela, de vérifier que les deux conditions suivantes soient satisfaites :

(7.41.a) $j \in D(k)$

(7.41.b) $\left(R_{n-1}(i,j, R_k, n) , R_k\right)$ unifie une règle de coarticulation arrière.

Dans le cas où la décision précédente est affirmative, pour valider ou non l'hypothèse de coarticulation au point (i,j, R_k, n), nous associons en ce point la distance locale $d^{R_k, CAR(k)}(i,j)$ définie par :

$$(7.42) \qquad d^{R_k, CAR(k)}(i,j) = MIN\left(d^{R_k}(i,j),\ d\left(i,\ CAR(k)\right)\right)$$

où $d\left(i,\ CAR(k)\right)$ est la distance locale entre le $i^{ème}$ prélèvement de la forme inconnue et le prélèvement $CAR(k)$.

VI.8.4.3.- Difficulté de la mise en oeuvre des règles de coarticulation avant .

L'utilisation des règles de coarticulation avant dans la comparaison dynamique est nettement plus complexe que celle concernant les règles de coarticulation arrière. En effet, en un point (i,j, R_k, n) de l'espace de comparaison avec $j \in F(k)$, il est évident qu'il est impossible de déclencher une règle de coarticulation avant dans la mesure où la deuxième forme de référence qui doit être associée à R_k pour permettre l'unification des règles ne peut être connue qu'au niveau $n+1$. Ceci implique que les règles de coarticulation avant ne peuvent intervenir qu'en début de niveau $n+1$ et en conséquence aucune décision ne peut être prise au niveau n pour tenir compte ou non d'un effet de coarticulation. Il vient donc qu'à ce niveau les deux hypothèses - effet de coarticulation, pas d'effet de coarticulation - doivent être considérées en parallèle au cours de la comparaison et le seul moyen - du moins le seul que nous ayons trouvé jusqu'à présent - pour réaliser cela est d'associer en tout point (i,j, R_k, n) avec $j \in F(k)$ deux distances cumulées globales, l'une étant évaluée en ignorant une éventuelle distorsion due à la coarticulation, l'autre au contraire, étant calculée en supposant que la zone en cours d'analyse de la phrase inconnue subit une déformation due à une coarticulation avant. Or la structure des algorithmes de programmation dynamique se prête assez mal à la mise en oeuvre de ce type de procédure. De ce fait la prise en compte à chaque niveau des deux hypothèses concernant une éventuelle distorsion due à une coarticulation avant nécessite un accroissement considérable de la complexité de l'algorithme. Devant ce constat - et surtout devant le fait que nous

n'avions plus d'espace mémoire disponible dans la machine avec laquelle nous travaillions jusqu'à présent - nous avons décidé d'implanter le mécanisme de déclenchement des règles de coarticulation avant dans une extension future de notre système de reconnaissance.

VI.8.4.4.- Conclusion.

Nous venons de montrer dans ce paragraphe, grâce au modèle de reconnaissance de mots enchaînés que nous proposons, qu'il est possible d'introduire dans les algorithmes de reconnaissance de mots enchaînés une source d'information phonologique permettant de tenir compte des effets dûs à la coarticulation. Ceci constitue selon nous un résultat intéressant qui va permettre - comme nous l'avons observé par les expériences que nous avons réalisées - la déduction d'algorithmes plus performants que ceux qui sont proposés actuellement et le commencement de certaines recherches concernant la réalisation de systèmes de reconnaissance de mots enchaînés à grand vocabulaire.

VI.9.- EXPERIENCES REALISEES EN RECONNAISSANCE DE MOTS ENCHAINES.

VI.9.1.- Remarques préliminaires.

1)- Les expériences de reconnaissance ont été effectuées dans une salle machine à fort bruit ambiant, dû essentiellement au système de ventilation du Mitra 125 et aux unités de disque, par un locuteur masculin.

2)- Le vocabulaire que nous avons considéré est le vocabulaire des dix chiffres.

3)- Pour paramétriser les différentes formes vocales nous avons utilisé le vocoder 16 canaux décrit précédemment avec une période d'échantillonnage ramenée par programme à 50 HZ.

4)- Afin de limiter la place mémoire occupée par le vocabulaire et aussi afin de permettre au locuteur d'effectuer des pauses de durée quelconque entre les mots nous avons réalisé une procédure d'acquisition qui supprime tous les silences à l'aide de seuils énergétiques et à l'aide d'un seuil de pitch.

VI.9.2.- Description de la première expérience.

- Pour réaliser cette première expérience nous avons implanté sur Mitra 125 notre algorithme de reconnaissance de mots enchaînés décrit au paragraphe VI.7.3 avec la contrainte locale symétrique à chemins locaux virtuels.

- Au cours de la phase d'apprentissage nous avons, pour chaque forme de référence, initialiser les tables $\beta(r)$ et $\delta(r)$ et conserver en mémoire un seul spectre caractérisant la forme de référence.

- Les paramètres de relachement des contraintes aux frontières que nous avons adoptés au cours de cette expérience sont les suivants :

$$\alpha = \beta = \gamma = \delta = 3 \quad .$$

- Nous avons pris comme demi-largeur de la fenêtre d'exploration $\varepsilon = 6$.

- Le corpus des phrases testées est un corpus de 40 phrases de deux à quatre chiffres et figure en annexe I.

- Sur les 40 phrases testées, 6 sont tombées en erreur. En annexe II nous donnons les 6 phrases mal reconnues et les phrases erronnées correspondantes qui ont été trouvées par le système de reconnaissance.

- Le temps de reconnaissance d'une phrase par ce système est de l'ordre de 3 mn - ce temps est la somme des temps nécessaires pour charger les différentes procédures en mémoire à partir d'une unité de disque - au cours de l'éxécution du programme; et pour réaliser l'éxécution de celles-ci -.

- Cette première expérience nous a permis de mettre en évidence le phénomène de la coarticulation car toutes les phrases mal reconnues sont des phrases sujettes à des effets de coarticulation.

VI.9.3.- Description de la deuxième expérience.

- Nous avons introduit, pour réaliser cette deuxième expérience, dans le système précédent les tables CAR et CAV ainsi que la procédure de déclenchement des règles de coarticulation arrière. En ce qui concerne les règles de coarticulation avant nous avons autorisé le programme à effectuer les substitutions de façon systématique.

- Cette nouvelle version ayant un temps de réponse de beaucoup supérieur à la version précédente - de l'ordre de 10 mn en moyenne par phrase - dû au fait qu'il nous a fallu segmenter les sous-programmes effectuant la comparaison dynamique afin que la branche principale du programme puisse tenir en mémoire centrale, nous nous sommes intéressés exclusivement au corpus des phrases mal reconnues au cours de l'expérience précédente.

- Nous avons adopté pour cette expérience les paramètres de relachement des contraintes aux frontières et la demi-largeur de la fenêtre d'exploration qui ont été utilisés au cours de la première expérience.

- Toutes les phrases qui avaient été mal reconnues au cours de l'expérience précédente ont été correctement analysées par ce nouveau système de reconnaissance.

C H A P I T R E VII :

CONCLUSION

Le travail que nous exposons dans ce mémoire a concerné la reconnaissance globale de la parole et plus particulièrement deux sujets parmi les plus importants de ce vaste domaine : *la reconnaissance de mots isolés et la reconnaissance de mots enchaînés.*

Dans le cadre du premier volet de notre étude, à savoir la reconnaissance de mots isolés, nous avons présenté un nouvel algorithme de programmation dynamique, plus puissant que ceux qui ont été proposés jusqu'à présent, qui, tout en conservant au problème son caractère optimal, autorise un relachement des contraintes aux frontières des chemins de recalage temporels afin de compenser certains bruits parasites pouvant affecter les débuts ou fins de mots, ou bien, certaines erreurs dues à l'algorithme de segmentation parole - non parole. Ce nouvel algorithme, tel que nous l'avons présenté dans ce rapport est, à notre avis, encore loin d'avoir atteint sa spécification optimale car à l'heure où nous écrivons cette conclusion nous avons à l'esprit des idées qui pourraient le rendre encore plus performant et que nous comptons mettre en oeuvre prochainement.

En ce qui concerne la deuxième partie de notre travail, nous avons montré comment il est possible d'utiliser une contrainte locale symétrique ou bien d'effectuer un relachement des contraintes aux frontières dans les algorithmes de reconnaissance de mots enchaînés en mettant en oeuvre des fonctions de normalisation similaires à celles qui ont été employées dans l'algorithme de programmation dynamique que nous avons conçu. Nous avons mis en évidence, d'autre part, grâce aux expériences que nous avons réalisées, que la coarticulation

ne peut être ignorée comme le font systématiquement tous les algorithmes présentés jusqu'à présent. A ce sujet nous avons proposé un nouveau modèle de reconnaissance de mots enchaînés qui, par l'intermédiaire d'une source d'information phonologique, permet de tenir compte des distorsions coarticulatoires, et, nous avons explicité comment celle-ci est à même de communiquer avec l'algorithme de comparaison dynamique par le biais des règles de coarticulation.

Nous pensons, par les différentes idées et les différents résultats qui sont contenus dans ce mémoire, avoir contribué à la reconnaissance globale de la parole et nous serions heureux de pouvoir le faire encore.

A N N E X E I

PHRASES TESTEES PAR LE PREMIER SYSTEME DE RECONNAISSANCE.

UN	QUATRE		UN	TROIS	SEPT		UN	DEUX	TROIS	QUATRE
SIX	CINQ		SIX	CINQ	TROIS		UN	HUIT	TROIS	NEUF
SEPT	TROIS		QUATRE	TROIS	NEUF		SIX	CINQ	SEPT	HUIT
QUATRE	NEUF		DEUX	UN	HUIT		UN	ZERO	HUIT	DEUX
HUIT	TROIS		ZERO	TROIS	QUATRE		TROIS	QUATRE	CINQ	SIX
CINQ	DEUX		SIX	CINQ	DEUX		UN	NEUF	ZERO	QUATRE
NEUF	SIX		UN	HUIT	NEUF		DEUX	SIX	DEUX	TROIS
ZERO	QUATRE		TROIS	QUATRE	SIX		TROIX	CINQ	SIX	QUATRE
UN	DEUX		SEPT	HUIT	QUATRE		ZERO	TROIS	QUATRE	NEUF
TROIS	CINQ		DEUX	NEUF	CINQ		SIX	SIX	TROIS	TROIS
UN	SIX		UN	ZERO	DEUX					
DEUX	NEUF		TROIS	TROIS	QUATRE					
TROIS	ZERO		DEUX	UN	DEUX					
SIX	HUIT		SIX	NEUF	TROIS					
ZERO	DEUX		UN	SEPT	HUIT					

A N N E X E II

PHRASES MAL RECONNUES PAR LE SYSTEME DE RECONNAISSANCE			
CINQ	DEUX		
UN	DEUX		
DEUX	UN	HUIT	
SIX	CINQ	DEUX	
UN	NEUF	ZERO	QUATRE
DEUX	SIX	DEUX	TROIS

PHRASES ERRONNEES DETERMINEES PAR LE SYSTEME DE RECONNAISSANCE			
DEUX	DEUX		
NEUF	DEUX		
DEUX	HUIT		
SIX	DEUX	DEUX	
QUATRE	NEUF	ZERO	QUATRE
DEUX	DEUX	DEUX	TROIS

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

[BELL, 1957] R. BELLMAN
"DYNAMIC PROGRAMMING". Princeton, N.F. : Princeton, Univ. Press, 1957

[BRID, 1982] J.S. BRIDLE, MD. BROWN, R.M. CHAMBERLAIN :
"An algorithm for connected word recognition"
PROC 1982 IEEE International Conference on Acoustics, Speech and Signal Processing, Paris, France, pp. 899-902, May 1982.

[DAS, 1982] S.K. DAS
"Some Experiments in Discrets Utterance Recognition"
IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. assp-30, n° 5, October 1982.

[DIMA, 1983] J. DI MARTINO, J.P. HATON, M.C. HATON
"Evaluation d'Algorithmes en Reconnaissance Automatique de la Parole". 11th International Congress on Acoustics, Paris, France, pp. 192-202, juillet 1983.

[FLAN, 1972] J.L. FLANAGAN
"Speech Analysis, Synthesis and Perception"
2nd ed., Springer-Verlag, New-York, 1972.

[GAUV, 1982] J.L. GAUVIN, J. MARIANI
"A Method for Connected Word Recognition and Word Spotting on a Microprocessor", Proc 1982, IEEE International Conference on Acoustics, Speech and Signal Processing, Paris, France, pp. 891-894, May 1982.

[HATO, 1974] J.P. HATON
"Contribution à l'Analyse, la Paramétrisation et la Reconnaissance de la Parole".
Thèse d'Etat, Université de Nancy I, 1974.

[ITAK, 1975] F. ITAKURA
"Minimum Production Residual Principle Applied to Speech Recognition", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-23, pp. 67-72, February 1975.

[LAZR, 1983] M. LAZREK
"Decodage acoustico-phonétique en compréhension automatique de la parole continue"
Thèse de 3ème cycle, Université de Nancy I, 1983

[MYER, 1980] C.S. MYERS, L.R. RABINER, A.E. ROSENBERG
"Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition"
IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. Assp-28, n° 6, December 1980.

[MYER, 1981] C.S. MYERS, L.R. RABINER
"A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition"
IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-29, n° 2, April 1981.

[NAKA, 1983] S. NAKAGAWA
"A Connected spoken word Recognition Method by O(n) Dynamic Programming Pattern Matching Algorithm",
Proc 1983, IEEE International Conference on Acoustics, Speech and Signal Processing, Boston, pp. 296-299, April 1983.

[PERO, 1984] J.Y. PEROT
Thèse de 3ème cycle - A paraître.

[RABI, 1975] L.R. RABINER and B. GOLD
"Theory and Application of Digital Signal Processing"
Englewood Cliffs, NJ : Prentice-Hall, 1975.

[RABI, 1978] L.R. RABINER, A.E. ROSENBERG
"Considerations in Dynamic Time Warping Algorithms for Discrete Word Recognition"
IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. Assp-26, n° 6, December 1978.

[RABI, 1980] L.R. RABINER, C.E. SCHMIDT
"Application of Dynamic Time Warping to Connected Digit Recognition".
IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. Assp-28, n° 4, August 1980.

[SAKO, 1978] H. SAKOE, S. CHIBA
"Dynamic Programming Algorithm Optimization for Spoken Word Recognition".
IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. Assp- 26, n° 1, February 1978.

[SAKO, 1979] H. SAKOE
"Two-level DP-Matching - A Dynamic Programming - Based Pattern Matching Algorithm for Connected Word Recognition".
IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. Assp-27, n° 6, Decembre 1979.

[SCHA, 1972] R.W. SCHAFER
"A Survey of Digital Speech Processing Techniques"
IEEE Transactions on Audio and Electroacoustics AU-20, n° 4, pp. 28-37, March 1972.

NOM DE L'ETUDIANT : DI MARTINO Joseph

NATURE DE LA THESE : Doctorat Ingénieur en Informatique

VU, APPROUVE ET PERMIS D'IMPRIMER

NANCY, le - 3 AVR. 1984 CG3

LE PRESIDENT DE L'UNIVERSITE DE NANCY I

R. MAINARD

L'objet de ce travail concerne la reconnaissance globale de la parole et plus particulièrement la reconnaissance de mots isolés et la reconnaissance de mots enchaînés.

Dans le cadre de la reconnaissance de mots isolés nous présentons un nouvel algorithme de programmation dynamique qui permet, tout en conservant au problème son caractère optimal, de relacher les contraintes aux frontières assujettissant les chemins de recalage temporels afin de compenser certaines distorsions pouvant affecter les début ou fin de formes dues soit à des bruits parasites, soit à des erreurs de détection de parole.

Dans l'étude consacrée à la reconnaissance de mots enchaînés nous montrons dans un premier temps comment il est possible d'utiliser une contrainte locale symétrique ainsi que de relâcher les contraintes aux frontières.

Dans un deuxième temps nous mettons en évidence le fait que la coarticulation ne peut être ignorée comme le font systématiquement pratiquement tous les algorithmes de reconnaissance de mots enchaînés existants et nous proposons un nouveau modèle de reconnaissance pouvant tenir compte des effets dûs à la coarticulation grâce à une source d'information phonologique.

Mots-clefs : Mots isolés, mots enchaînés, programmation dynamique, contrainte locale, relâchement des contraintes aux frontières, coarticulation.

L'objet de ce travail concerne la reconnaissance globale de la parole et plus particulièrement la reconnaissance de mots isolés et la reconnaissance de mots enchaînés.

Dans le cadre de la reconnaissance de mots isolés nous présentons un nouvel algorithme de programmation dynamique qui permet, tout en conservant au problème son caractère optimal, de relacher les contraintes aux frontières assujettissant les chemins de recalage temporels afin de compenser certaines distorsions pouvant affecter les début ou fin de formes dues soit à des bruits parasites, soit à des erreurs de détection de parole.

Dans l'étude consacrée à la reconnaissance de mots enchaînés nous montrons dans un premier temps comment il est possible d'utiliser une contrainte locale symétrique ainsi que de relâcher les contraintes aux frontières.

Dans un deuxième temps nous mettons en évidence le fait que la coarticulation ne peut être ignorée comme le font systématiquement pratiquement tous les algorithmes de reconnaissance de mots enchaînés existants et nous proposons un nouveau modèle de reconnaissance pouvant tenir compte des effets dûs à la coarticulation grâce à une source d'information phonologique.

Mots-clefs : Mots isolés, mots enchaînés, programmation dynamique, contrainte locale, relâchement des contraintes aux frontières, coarticulation.