Rapport de Stage DEA ATIAM

Suivi de Voix Parlée grâce aux Modèles de Markov Cachés

Thomas Pellegrini et Raphaël Duée
avec la participation de Lise Georges

Encadrant: Diemo Schwarz

Lieu: IRCAM
1, place Igor Stravinsky
75004 PARIS

Juin 2003

Table des matières

De l'intérêt du travail à deux.

Nous avons réalisé le stage présenté dans ce document à deux. Pour nous, le fait de travailler en binôme sur le même sujet nous a beaucoup apporté. Cela nous a, en effet, permis de partager nos connaissances et de nous répartir le travail à effectuer. Cela a été une expérience très enrichissante. Nous avions chacun à tout moment un interlocuteur avec qui parler, expliquer ses idées et demander des explications mais aussi un personne apte à critiquer et démentir nos affirmations personnelles.

Il est souvent difficile, lorsque l'on travaille seul, de prendre du recul sur son étude et ses conclusions. Le fait de travailler à deux a, selon nous, limité le travail inutile et permis d'avancer plus rapidement dans l'étude du dispositif.

De plus, Lise Georges, auditrice libre au DEA ATIAM, en venant participer à nos recherches, nous a procuré un regard extérieur et une aide précieuse grâce à son entousiasme et sa capacité à synthétiser des problèmes complexes. L'équipe ainsi formée a su étudier dans une ambiance de travail et de bonne humeur


Les devises Shadok
ROUXEL
EN ESSAYANT CONTINUELLEMENT
ON FINIT PAR RÉUSSIR. DONC:
PLUS ÇA RATE, PLUS ON A
DE CHANCES QUE ÇA MARCHE.

Remerciements

Nous remercions M. le Directeur Bernard Stiegler, de nous avoir permis d'effectuer notre stage de DEA au sein de l'institut IRCAM.

Nous remercions Norbert Schnell de nous avoir acceuilli au sein de son équipe, l'équipe Applications Temps Réel (ATR). Les différentes réunions auxquelles il a participé ont permis de bien planifier et diriger nos recherches.

Nous remercions vivement Diemo Schwarz, chargé de recherche et développement au sein de l'équipe ATR, qui nous a encadré tout au long du stage. Son engouement pour notre sujet, son aide, ses idées, ses questions et surtout ses connaissances illimités nous ont beaucoup stimulé et aidé.

Nous remercions également l'équipe Systèmes qui nous a supporté pendant ces quelques mois sur leur mezzanine ainsi que Vincent Goudard et Remy Muller. Nous ne remercierons pas la machine à café qui ne nous a pas beaucoup aidé durant ce travail et dédions ce travail à nos Mamans.

Chapitre 1

Introduction

La reconnaissance et le suivi sonores sont des domaines qui ont été étudiés depuis plusieurs décennies. Dans l'industrie ils ont de multiples sujets d'application en téléphonie, en multimédia mais aussi en commande vocale ou en ergonomie. Dans le domaine artistique, plus récemment, des metteurs en scène et des compositeurs de spectacles vivants s'y sont aussi intéressé. Ils cherchent à utiliser la capacité d'un système à reconnaître des sons, des paroles ou des musiques et à les suivre en temps réel. Un tel système aiderait énormément les ingénieurs du son à agir en temps réel en déclenchant des lumières, des effets ou des sons synchonisés avec le déroulement du spectacle.

C'est dans cette optique que Norbert Schnell, responsable de l'équipe Applications Temps Réel de l'Ircam, et Diemo Schwarz, chargé de recherche, nous ont proposé un stage et que nous avons accepté. Tous les deux élèves du DEA ATIAM de l'Université Paris VI, nous avions en effet un stage obligatoire de quatre mois à réaliser. C'est donc à l'Ircam et sous la tutelle de Diemo Schwarz que nous l'avons fait.

Ce stage a pour thème : "Le suivi de voix parlée grâce aux Modèles de Markov Cachés" (HMM). Il visait à théoriser et développer un système de suivi de parole en temps réel le plus précis possible. Cette précision de suivi en temps réel devait être de l'ordre du phonème (un son) pour pouvoir être directement appliqué à un spectacle à venir. Connaissant le texte par avance, il fallait réussir à le suivre lorsqu'il était dit par un acteur pour pouvoir déclencher des évènements suivant le déroulement des paroles. L'idée de cette application est venue de Gilles Grand, compositeur à l'Ircam, travaillant sur la pièce de théâtre "Retour Définitif et Durable de l'Etre Aimé". Il désirait réussir à déclencher une voix synthétisée synchronisée en temps réel avec

la voix véritable de l'acteur disant le texte. Ainsi un acteur pourrait parler avec la voix d'une autre personne.

Nous avons résumé dans ce document les différentes études que nous avons réalisées. Après un bref aperçu du contexte dans lequel se place le sujet de notre stage, le concept du dispositif est décrit en détail dans le chapitre 3. Ce concept est basé sur la création d'un Modèle de Markov Caché cherchant à s'approcher le plus possible du flot de parole pour pouvoir le suivre. Différents algorithmes sont décrits ainsi que l'architecture du modèle. Ensuite, nous avons décrit les tests réalisés pour affiner les nombreux paramètres du modèles et donnés nos résultats. Ce stage a permis de réaliser une grosse partie de l'étude sur le suivi de voix parlée grâce aux HMM mais beaucoup de travail reste à faire. Nous avons donc décrit les limites et les évolutions possibles du système dans le chapitre 5. Le dernier chapitre explique les différents problèmes que Diemo Schwarz et nous-même avons rencontrés lors de la création de l'interface jMax nécessaire au fonctionnement en temps réel du système.

La principale difficulté de cette étude a été de réussir à trouver les paramètres pertinents qui permettaient un suivi de la voix. En effet, les HMM sont des modèles que l'on entraine préalablement sur des données d'apprentissage afin ensuite de pouvoir suivre des données de test. Si les paramètres sont trop précis et l'entrainement trop bien réalisé, le système ne marchera que sur les données d'apprentissage et pas sur les données de test. Il faut donc réussir à trouver un juste milieu entre un modèle pertinent qui donne la liberté adéquate pour suivre des données différentes mais du même type.

Chapitre 2

Etat de l'Art

Voici pour commencer un bref aperçu des théories et des recherches passées que nous avons utilisées durant notre stage. Avant de commencer nos recherches, une étude bibliographique à été réalisée pour nous permettre de bien cerner et de maîtriser les différents sujets à aborder. Nous décrirons tout d'abord brièvement la théorie des Modèles de Markov Cachés générale. Les modèles que nous utiliserons ensuite sont particuliers. Ils seront décrits dans le chapitre 3, Spécificités du stage. Nous cherchons à suivre la parole grâce à un modèle HMM. Pour ce faire, il est nécessaire de la segmenter en zones : ce sont le phonèmes. Ce seront ces entitées linguistiques qui seront suivies. Enfin, la reconnaissance vocale est un domaine qui est extrêmement proche du suivi de la voix. C'est pourquoi nous l'avons étudié et en présentons un bref aperçu.

2.1 Des Modèles de Makov Discrets aux Modèles de Markov Cachés

Les Modèles de Markov Discrets sont basés sur une suite (ou boucle) d'états dans lesquels on navigue par des probabilités de transition et suivant des observations. Nous les illustrerons par l'exemple de la Météo. Le modèle créé est une boucle d'état de temps : pluis nuageux ensoleillé. Ils sont représentés figure 2.1.

Grâce à ces modéles, on peu connaître, sachant l'observation passée (La météo de la veille), quel va être la probabilité des observations futures. Dans l'exemple que nous avons pris, ce serait par exemple la probabilité

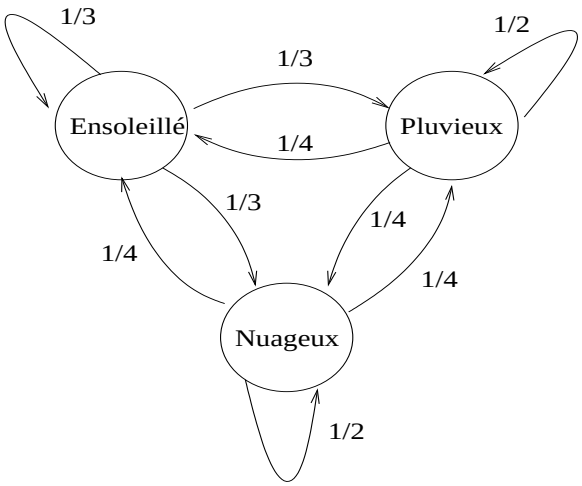


FIG. 2.1 – Modèles de Markov Discrets

qu'il fasse beau pendant 10 jours.

La nouveauté qu'apportent les Modèles de Markov Cachés par rapport aux Modèles de Markov Discrets, réside dans le fait que les états sont caractérisés par des distributions de probabilité sur l'espace des observations possibles, pluie, nuageux ensoleillé. Ces états correspondent alors à une variable qui n'est plus observée durectement. Ceci pourrait par exemple correspondre au fait d'être ou non dans un dépression atmosphérique. On a alors deux modèles stochastiques liés : le temps et la depression atmosphérique. On peut aussi illustrer ces modèles par le lancement de plusieurs pièces de monnaie biaisées les unes après les autres dont on ne retient que le résultat des lancements. Les deux séquences stochastiques liées seraient alors, la suite des états pile ou face et la suite des choix des pièces. Le modèle caché permet alors de connaître la probabilité d'avoir une séquence de pièces connaissant la séquence d'état pile ou face.

Les Modèles de Markov Cachés (MMC), dont nous préférerons employer l'acronyme anglais HMM pour Hidden Markov Models, sont, à l'heure actuelle, les outils de modélisation les plus employés en reconnaissance de la parole continue. Les HMM ont montré leur adéquation à traiter la parole. Ce sont des automates stochastiques permettant de déterminer la probabilité d'une suite d'observations. Ils sont définis par l'ensemble de données suivantes :

– une matrice A qui permet la définition de la topologie du HMM en indiquant les probabilités de transition d'un état q_i vers un autre

état q_j (ou lui-même)
- une matrice B qui contient les probabilités d'émission des observations dans chaque état $b_j(x_n) = p(x_n|q_j)$.
- une matrice donne la distribution de départ des états.

Le lecteur trouvera de plus amples informations sur les HMM dans [Rabiner et Juang. 1993][5] et [Rabiner et Al.1989][6]. La parole étant un phénomène temporel, l'utilisation de HMMs pose comme postulat que la parole est une suite d'évènements stationnaires. La topologie principalement employée dans la littérature est un modèle gauche-droit d'ordre 1 dit de Bakis.

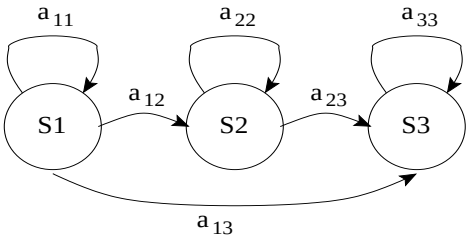


FIG. 2.2 – HMM

Cette topologie permet de prendre en compte les variations d'élocution dans le signal de parole. Ainsi, sur de la parole lente, il y a répétition d'états, c'est pour cela que ce modèle de HMM permet de boucler sur un état (transitions a_{ii} sur la figure précédente). A contrario, si la parole est rapide, il est aussi possible de sauter l'état suivant (arc a_{13}). Cette topologie permet la modélisation des variations temporelles au sein du signal de parole.

Hormis l'élocution, un autre phénomène variable dans la parole provient des variations de prononciation de chaque locuteur, et des différences entre locuteurs. Il est aisément compréhensible que deux personnes, même si elles prononcent le même énoncé, n'ont pas exactement le même résultat acoustique. Cela s'accentue encore entre hommes et femmes. La variation liée à un locuteur peut-être dûe à un état émotionnel particulier, le stress par exemple, ou à une altération temporaire de la voix suite à une maladie. C'est pour prendre en compte ces variations que les modèles HMMs utilisés de nos jours sont basés sur des fonctions multigaussiennes. Elles permettent de prendre en compte la variabilité autour de la moyenne calculée sur les données d'apprentissage.

Pour conclure, les HMMs et leurs caractéristiques représentent donc deux processus stochastiques distincts imbriqués : les modèles de Markov

sont dits cachés parce que la suite d'états parcourus pour générer la séquence O n'est pas directement observable.

- le premier est la suite d'observations produites $O = O_1, O_2, \ldots, O_n$
- le second est la suite d'états parcourus $Q = q_1, q_2, \ldots, q_n$

Ce type de modèle est utilisé pour un nombre de plus en plus grand d'analyses temps réel, pour l'analyse de la voix parlée en particulier. Nous verrons plus loin pourquoi il a été choisi dans notre cas particulier.

2.2 Le phonème : explications

Il existe beaucoup d'approches différentes concernant la parole. La phonétique étudie la production des sons de la parole, la transmission et leur perception. La phonologie étudie comment ces sons participent au fonctionnement de la parole.

Définition :
La phonologie introduit la notion de phonème. Le phonème est "la plus petite unité phonique fonctionnelle, i.e. distinctive"[10]. Un phonème n'a de sens qu'au sein d'autres phonèmes pour former des mots par exemple. On parle de "triphone" ou d"'allophone" pour désigner un phonème et ses plus proches voisins.
Nous avons utilisé une liste de phonèmes "SAMPA"(Speech Assessment Methods Phonetic Alphabet) qui est la liste officielle des phonèmes du français pouvant être utilisée en informatique (les caractères utilisés sont des caractères ASCII). Cette liste est la suivante :

Notation	SAMPA Exemple	Exemple en SAMPA
p	pont	po
b	bon	bo
t	temps	ta
d	dans	da
k	quand	ka
g	gant	ga
f	femme	fam
v	vent	va
s	sans	sa
z	zone	zon
S	champ	Sa
Z	gens	Za
j	ion	jo
m	mont	mo
n	nom	no
J	oignon	oJo
N	camping	ka piN
l	long	lo
R	rond	Ro
w	coin	kwe
H	juin	ZHe
i	si	si
e	ses	se
E	seize	sEz
a	patte	pat
A	pâte	pAt
O	comme	kOm
o	gros	gRo
u	doux	du
y	du	dy
2	deux	d2
9	neuf	n9f
@	justement	Zyst@ma
e	vin	ve
a	cent	va
o	bon	bo
9	brun	bR9

Tab. 2.1 – Liste SAMPA des phonèmes du français

Les phonèmes sont regroupés par classes selon leurs similarités :

Les consonnes.

- "p b t d k g"sont les six plosives, qui peuvent être elles-même séparées en 2 sous-classes : les plosives voisées (b, d et g) et les plosives non-voisées (p, t et k). Ces dernières se caractèrisent par un silence puis un bruit donc sont très faciles à segmenter de visu,
- "f v s z S Z j"sont les six fricatives. "v z Z"sont les fricatives voisées, les autres sont non-voisées,
- "m n J N"sont les nasales,
- "l R w H"sont les liquides,

Les voyelles.

- "i e E a A O o u y 2 9 @" sont les douzes voyelles "orales".
- "e∼ a∼ o∼ 9∼"sont les voyelles nasales.

2.3 La reconnaissance de la parole

Le problème de la reconnaisance automatique de parole est étudié depuis plusieurs décennies. Il constitue un terrain de recherche passionnant dans la mesure où les outils nécessaires sont très divers : traitement du signal, modéles mathématiques statistiques puissants, classification de formes, algorithmique... Les applications sont multiples : systèmes de commande et contrôle sur PC, systèmes de dictée vocaux, le monde du spectacle... Lorsque l'on veut mettre au point un système de reconnaissance de parole, les différentes questions à se poser sont les suivantes :

- La reconnaissance doit-elle être mono- ou multi-locuteurs ? Il est clair qu'il est plus simple de mettre au point un système mono-locuteur dans la mesure où les variabilités de prononciation sont réduites. Pour réaliser un système multi-locuteur, l'apprentissage doit être réalisé sur de nombreux exemples suffisamment différents pour couvrir tout l'éventail des prononciations possibles, pour des voix d'hommes, de femmes, ou d'enfants.
- La reconnaissance porte-t'elle sur des mots isolés ou de la parole continue ?
- Dans quelles conditions va être utilisé le système ? Beaucoup de facteurs altèrent le bon fonctionnement d'un reconnaisseur de pa-

role : les bruits ambiants, les caractéristiques du matériel utilisé (limitations des bandes passantes), élocution inhabituelle...

Les modèles les plus couramment utilisés aujourd'hui sont les modèles de Markov cachés introduits au paragraphe précédent ([6]). Leur fonctionnement pour le système que nous avons étudié est détaillé dans le chapitre 3. La spécificité de notre stage réside en l'application particulière de la reconnaissance de parole qui est de suivre la parole d'acteurs disant un texte connu à l'avance. Dans la suite, nous parlerons donc plus de suivi de parole plutôt que de reconnaissance de parole.

Chapitre 3

Spécificité du Stage : Le suivi de parole

Le modèle HMM que nous avons utilisé cherche à s'approcher le plus possible de l'énonciation d'une phrase pour pouvoir la suivre. Comme nous l'avons vu précédemment, ces modèles ont besoin d'être entrainés. Nous l'avons fait sur deux niveaux, au niveau du phonème et au niveau de la phrase entière, c'est à dire de la liaison entre phonèmes. Après une courte introduction expliquant les antécédents en suivi departition et de parole, nous décrirons le modèle créé et les algorithmes utilisés pour l'entrainement et le suivi temps réel.

3.1 Antécédents

L'équipe Applications Temps Réel de l'Ircam, avec Diemo Schwarz et Nicolas Orio, a déjà travaillé sur le développement d'outils Temps Réel appliqués à l'alignement de musique avec une partition (cf [Orio et Déchelle. 2001][3]) et au suivi de partition (cf [Orio et Schwarz. 2001][2]). Ces deux sujets sont très liés car le suivi d'une partition nécessite un alignement de la musique avec une partition. L'alignement a été précédemment réalisé grâce à l'algorithme DTW (Dynamic Time Warping) calculant les distances locales entre les paramètres spectraux utilisés. L'évolution de cette étude a conduit à utiliser les modèles de Markov cachés pour le développement de l'algorithme de suivi de partition. Ces outils statistiques sont utilisés sur deux niveaux différents, au niveau de la note et au niveau de la partition (Bas niveau et Haut niveau). En effet, deux modèles de Markov sont imbriqués. Chaque

note a son petit modèle et ces petits modèles de notes sont régis grâce à un second modèle de Markov de plus haut niveau qui permet de créer le modèle de la phrase musicale. A.Loscos, P.Cano et J.Bonada ont aussi beaucoup travaillé sur le suivi de voix. Ils ont mis au point un système de suivi de voix chantée grâce aux HMM (cf [Cano, Loscos et Bonada. 1999][7] et [Loscos, Cano et Bonada. 1999][8]) et un ensemble de probabilités temporelles de rester sur une note (Time Duration). Les paramètres qu'ils utilisent sont très divers mais prennent surtout en compte la fréquence fondamentaledu signal.

Les HMM sont très adaptés l'analyse de la parole car leur structure leur donne la capacité de reboucler sur un même état. Cela permet de prendre en compte la variabilité temporelle de la diction d'une phrase. L'apport des modèles dits Cachés par rapport aux Modèles de Markov classiques réside dans le fait que l'on ne sait jamais dans quel état on est. Il y a toujours une description probabiliste de la possibilité d'être dans tous les états. C'est ce que l'on appelle la logique floue.

Ces études passées ont beaucoup inspiré nos recherches pour le suivi de voix parlée. Nous avons, par exemple, repris le système a deux niveaux de HMM. Cependant, la nécessité de suivre la voix parlée et non chantée implique un problème nouveau par rapport à ces anciennes recherches : quels paramètres prendre pour décrire la voix parlée ?

3.2 Le Modèle HMM utilisé

Comme ce qui a été fait en suivi de partition [4], nous avons raisonné en définissant deux niveaux :

- Le bas-niveau, constitué des modèles de Markov cachés des phonèmes. Il y a un modèle par phonème.

- Le haut-niveau, qui représente les phrases. Il est défini par des probabilités de transition entre les modèles des phonèmes (du bas-niveau).

3.2.1 Le Bas niveau

Chaque phonème est modélisé par un cluster à 2 ou 3 états : les plosives (p, b, t, d, k, g) et le phonème silence sont usuellement à 2 états. Tous les autres phonèmes sont à 3 états. Les états modélisent un équivalent de la décomposition “attack “ “sustain “ et “release “ pour les notes de musique. Dans le cas des plosives, comme elles commencent toutes par un silence (ce qui permet de les reconnaître facilement) puis par une partie plus ou moins bruitée selon le contexte, elles ne sont modélisées que par deux états. Chaque modèle est composé de :

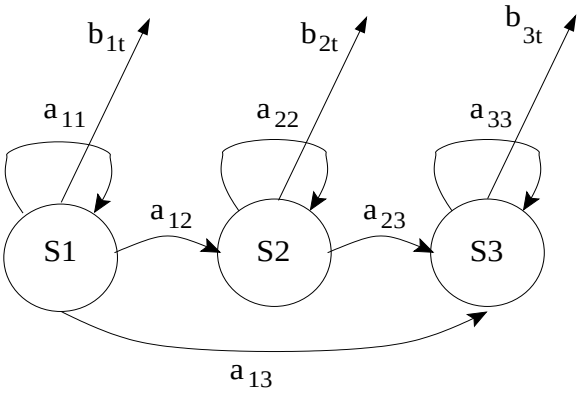


FIG. 3.1 – Modèle de phonème à 3 états

– une matrice de transition $A = [a_{ij}]$ de taille $3 * 3$ ou $2 * 2$ selon le nombre d'états du phonème. Pour les matrices $3 * 3$:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix}$$

Pour les matrices $2 * 2$:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} \\ 0 & 1 \end{pmatrix}$$

Ces matrices sont triangulaires supérieures, ce qui traduit le caractère gauche-droite du modèle (left-to-right model). Toutes les transitions ont lieu de droite à gauche. Les coefficients de la diagonale sont les probabilités de boucler sur un même état. Les coefficients de la première diagonale supérieure sont les probabilités de transition à l'état suivant. Enfin, pour les matrices 3*3, le scalaire de la segonde diagonale supérieure représente la probabilité de sauter le premier état. Chaque ligne est normalisée à 1, puisqu'il s'agit de probabilités.

– un vecteur décrivant les probabilités d'être dans l'un des états au temps initial.
– un mélange de gaussiennes qui décrit les probabilités $B = [b_{it}]$ d'observer un vecteur de données suivant l'état dans lequel est le système et suivant la gaussienne choisie.

Avoir plusieurs gaussiennes revient à prendre en compte les différentes manières de prononcer un même phonème. Nous avons choisi de travailler avec 2 gaussiennes. Chaque sous-modèle comporte une matrice de moyennes d'observation (sample mean vector) pour chaque vecteur de données sachant que le système est dans un état précis et suit une gaussienne précise. Il contient également une matrice de covariance des paramètres (sample covariance matrix).

– une matrice appelée mixmat dans les programmes qui donne la probabilité de choisir une gaussienne connaissant l'état dans lequel se trouve le système.

3.2.2 Haut niveau

Le modèle haut-niveau est créé en assemblant les modèles bas-niveau, suivant la liste des phonèmes de la phrase à apprendre. La matrice des transitions est obtenue en assemblant les matrices de transition élémentaires et en définissant des probabilités de transition entre ces dernières. Un saut de phonème peut être ajouté pour éviter le bloquage du suivi.

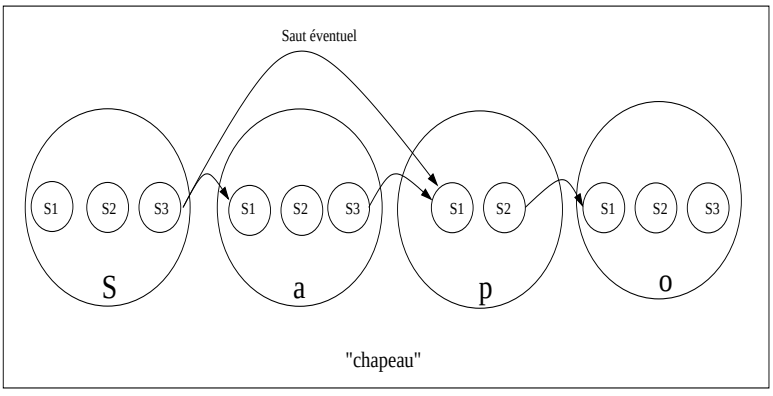


FIG. 3.2 – structure haut-niveau.

Dans la figure 3.2, le mot modélisé est "chapeau". Le "p"est une plosive, qui est modélisée par deux états seulement. Un saut éventuel du phonème "a" est figuré. Remarque : dans la version en C, des pseudo-états ont été ajoutés entre chaque phonème. Il s'agit d'états qui n'ont pas d'obser-

vation (probabilités d'observations nulles), ils servent à définir la probabilité de sortir d'un phonème ("α_{in}") et celle de passer au premier état du suivant ("α_{out}").

3.2.3 Mixture de gaussiennes

Le modèle gaussien simple

Il faut définir un modèle pour les probabilités d'observation $p(o|\lambda)$ où λ est le modèle HMM défini dans le paragraphe précédent. Partant d'exemples dont nous soyons sûrs qu'ils caractérisent le phonème que l'on veut modéliser, le modèle gaussien est créé en calculant simplement le vecteur μ des moyennes de chaque paramètre et la matrice de covariance U :

$$\mu_i = \frac{1}{N}\sum_{n=1}^{N} o_n \tag{3.1}$$

$$U_i = \frac{1}{N-1}\sum_{n=1}^{N}(o_n - \mu_i)'(o_n - \mu_i) \tag{3.2}$$

Ultérieurement, pour déterminer la probabilité d'un vecteur observé, il suffit d'utiliser la formule classique d'une loi gaussienne :

$$p(o|\lambda) = \mathcal{N}(o; \mu_i, U_i) \tag{3.3}$$

$$\mathcal{N}(o; \mu, U) = \frac{1}{\sqrt{(2\pi)^p \|U|}} exp(-\frac{1}{2}(o_n - \mu_i)U^{-1}(o_n - \mu_i)') \tag{3.4}$$

Le modèle mélange de gaussiennes

L'intérêt de définir un modèle multi-gaussiennes consiste à généraliser le modèle précédent pour permettre au système de suivre différentes prononciations d'un même phonème (par exemple, voix normale, voix chuchotée, accent étranger...). En effet, cela prend en compte des variations de paramètres qui élargiraient trop les gaussiennes si l'on se contentait d'un modèle gaussien simple. Pour avoir des phonèmes bien distingués, il faut des gaussiennes qui ne se recouvrent pas. Chaque multi-gaussienne, constituée d'autant de gaussiennes qu'il y a de paramètres, est indépendante des autres. Ce modèle peut être compris comme un choix au hasard entre différentes "sous-classes",

qui sont des modèles gaussiens simples. Les probabilités de choisir une sous-classe G_{jk} (j : index de l'état, k : index de la multi-gaussienne) sont prises constantes :

$$p(G_{jk}|\lambda_j) = c_{jk} \tag{3.5}$$

La probabilité $p(o|\lambda_j)$ est ensuite calculée ainsi :

$$p(o|\lambda_j) = \sum_{k=1}^{M} p(o|G_{jk})p(G_{jk}|\lambda_j) \sum_{k=1}^{M} c_{jk}\mathcal{N}(o; \mu_{jk}, U_{jk}) \tag{3.6}$$

Dans le suivi de parole, les paramètres sont des variables aléatoires continues, les probabilités sont donc calculées avec des distributions gaussiennes continues.

3.2.4 Les états fantômes (Ghosts States)

Pour gérer les erreurs potentielles que peut faire l'acteur en temps réel (bégaiement, oubli, substitution de mot...), il est utile de définir des états dits fantômes. Lorsque le système ne reconnait plus la partition ou le texte normal, il peut se mettre dans un état d'erreur puis reprendre le suivi par la suite. Chaque modèle de phonème a son double fantôme. Ils ont exactement la même structure. La différence se fait lors de l'apprentissage : un modèle fantôme est entraîné sur tous les phonèmes des classes autres que celle du phonème qu'il double.

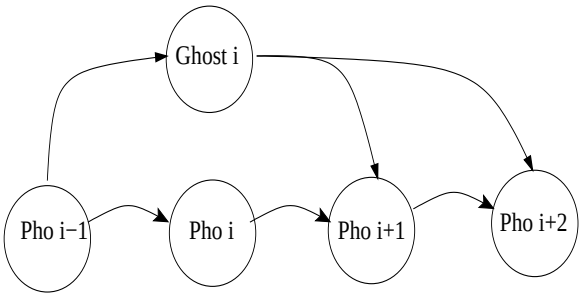


FIG. 3.3 – structure avec états fantômes.

La figure 3.3 montre un chemin haut-niveau (chaque cercle représente un cluster à 2 ou 3 états). Les possibilités de transition entre états normaux et états fantômes sont montrées. Dans un souci de clarté, un seul état fantôme a été figuré. Les choix suivants ont été faits :

– pas de saut possible à partir d'un état normal mais possible à partir d'un état fantôme,

– pas de transition possible entre 2 états fantômes successifs. Les transitions haut-niveau pour accéder à un g-state ou en sortir sont fixées arbitrairement. Elles ne sont pas modifiées par l'entraînement, puisque pour entraîner des probabilités d'erreurs, il faudrait disposer d'exemples comportant toutes les erreurs possibles et imaginables.

3.3 Les paramètres utilisés

Les paramètres utilisés pour le suivi sont extrêmement importants. Ces paramètres doivent, en effet, bien décrire les différents phonèmes séparément mais aussi ne pas être trop précis. Une précision trop grande des paramètres rendrait le système beaucoup trop axé sur une façon de parler. C'est pourquoi il faut réussir à équilibrer ces deux contraintes. Nous avons testé trois types de paramètres, des MFCC (Mel Frequency Cepstral Coefficient), les Delta-MFCC et le Taux de passage par zéro (Zero-crossing rate en anglais).

3.3.1 les MFCC

Les premiers paramètres sont directement tirés du spectre du signal. Nous donnerons brièvement les étapes de leur calcul pour ensuite les détailler plus précisément.

1) Fenêtrage du signal avec la fenêtre de Hamming.
2) Transformée de Fourier.
3) Filtrage par banc de filtres triangulaires espacés selon l'échelle Mel.
4) Transformée en Cosinus Discrète.
5) Transformée de Fourier inverse.
6) Centrage et mise à 1 de la variance du vecteur calculé.

Nous analysons le signal par fenêtres se recouvrant les unes les autres (Overlap). Chaque fenêtre produisant un jeu de paramètres acoustique. On filtre le signal par une fenêtre de Hamming pour éviter les effets de bord et on peut aussi faire du zéro Padding si la taille de la fenêtre ne convient pas à une transformée de Fourier rapide (1) (figure 3.5).

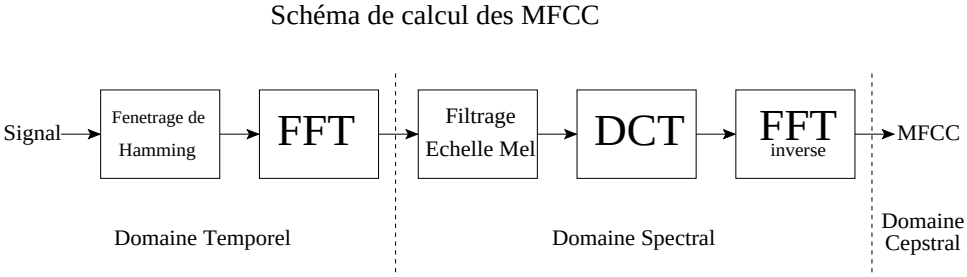


FIG. 3.4 – MFCC

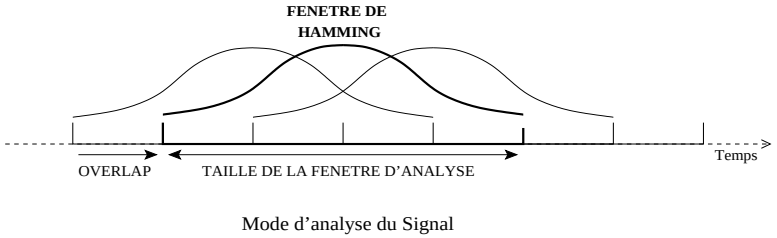


FIG. 3.5 – Méthode d'analyse du signal

Après avoir effectué une transformée de Fourier sur le signal (2), le spectre obtenu est traité par des filtres triangulaires en peigne (3). Nous cherchons à suivre la voix comme l'homme le ferait. Si on admet que l'oreille humaine est le meilleur outil pour suivre la voix, il faut utiliser ce que l'oreille entend réellement comme signal. Pour simuler l'oreille humaine, nous filtrons le signal par un banc de filtre qui ont chacun une réponse de bande passante triangulaire. Les filtres utilisés sont espacés de telle façon que leur évolution correspond à l'échelle Mel. Cette échelle tente de mettre en relation la tonie et les hauteurs perçues (figure 3.6). Elle n'établit pas une correspondance entre les fréquences et les mels mais tente d'établir un lien entre les sensations de hauteurs perçues. Elle met principalement une chose en valeur, c'est que sur le plan de la perception une octave entre deux sons graves, 250 et 500 Hz par exemple (250 mels), paraît plus petite qu'une octave entre deux sons aigus, 1000 et 2000 Hz (1800 mels).

Il est possible de limiter les bornes du banc de filtre pour ainsi plus cibler les MFCC sur une bande de fréquence particulière. Ainsi on peut enlever une partie des fréquences basses ou les hautes fréquences. Pour la voix, il est bon d'utiliser la bande de fréquence téléphonique 300-3400Hz qui donne une qualité sonore moindre mais une bonne intélligibilité du message. On réduit de plus les différences entre locuteurs. Voici la formule approchée

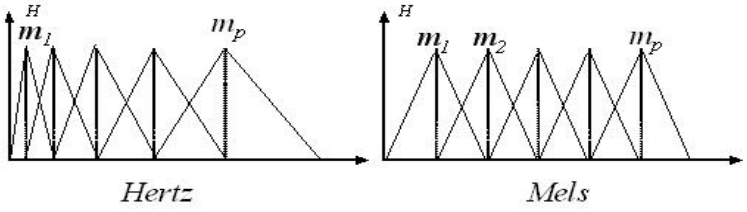


FIG. 3.6 – Figure du banc de filtres espacés sur l'échelle Mel en Hertz et en Mel

de l'échelle Mel :

$$Mel(f) = 2595 * log(1 + f/700) \tag{3.7}$$

On obtient ainsi un jeu de valeurs caractéristiques du signal et basé sur la perception de l'oreille. Le nombre de filtres pris peut varier. Nous en avons pris 29 pour avoir une bonne précision de calcul (dans la littérature, la valeur prise en général est 24).

Après une Transformée en Cosinus Discrète (4), on applique la transformée de Fourier inverse (5). On entre alors dans le domaine Cepstral ou Quéfrentiel. On obtient les coefficients MFCC avec comme premier coefficient l'énergie du signal. En général, une dizaine de coefficients MFCC est utilisée. Nous n'utilisons pas l'énergie car c'est un paramètre qui dépend trop du locuteur et de son volume sonore. Nous ne dépasserons pas la dizaine de coefficients lors des tests. La figure (3.7) montre un jeu de vecteurs de MFCC.

La "normalisation" réalisée ensuite (6) permet aux paramètres de rester dans un même ordre de grandeur sans en privilégier un en particulier. Après centrage du vecteur de MFCC (retrait de la moyenne), la variance du vecteur est mise à 1 (division de chaque coefficient par la variance du vecteur). Cette normalisation sera discutée dans la partie tests et résultats.

3.3.2 Les Delta-MFCC

Ces paramètres sont les dérivées temporelles des MFCC. Ils permettent de prendre en compte la variabilité temporelle de la parole ce qui est une de ses caractéristiques importante. Ils sont calculés grâce aux vecteurs passés de MFCC. Nous les avons testé sur plusieurs exemples. Nous verrons plus loin leur utilité. Les dérivées sont prises sur les paramètres de

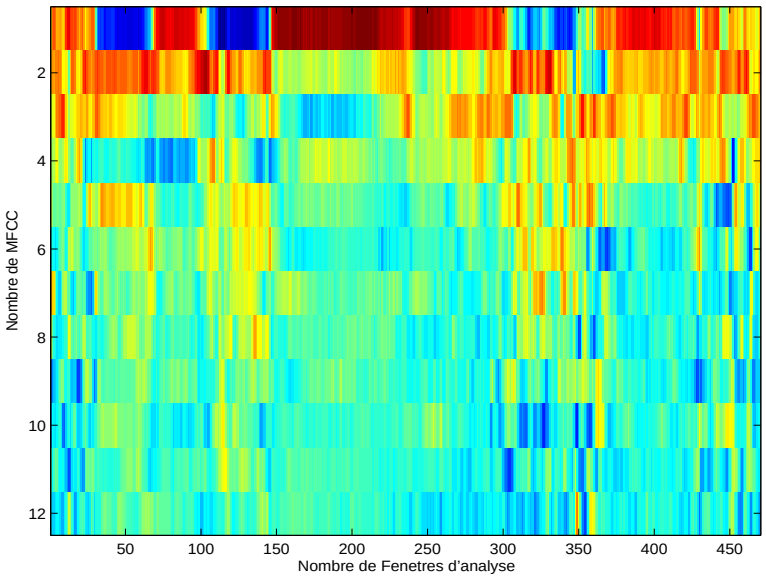


FIG. 3.7 – Coefficients MFCC. 12 coefficients pour la phrase "Je suis Robinson"

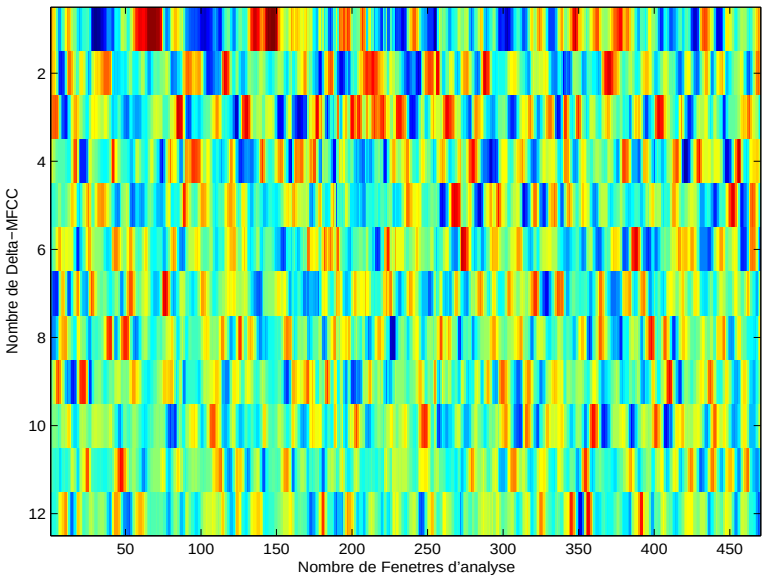


FIG. 3.8 – Coefficients Delta-MFCC. 12 coefficients pour la phrase "Je suis Robinson"

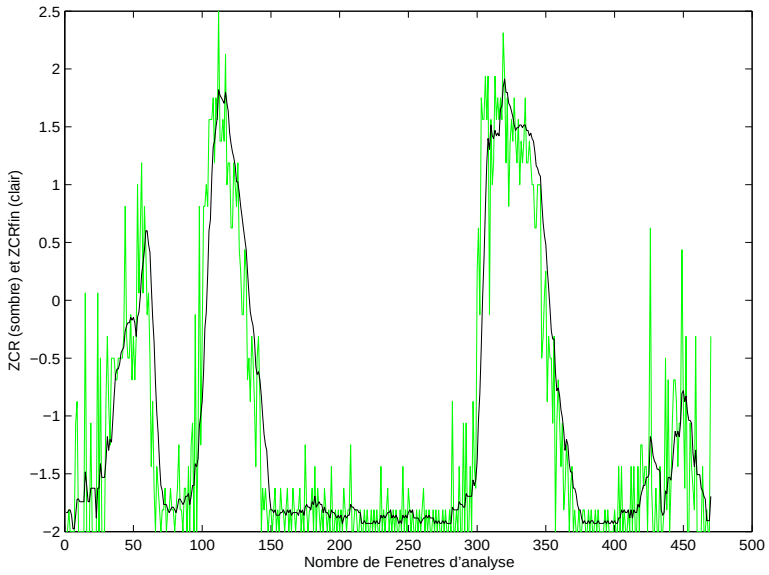


FIG. 3.9 – Coefficients ZCR (Foncé) et ZCR fin (Clair) pour la phrase "Je suis Robinson"

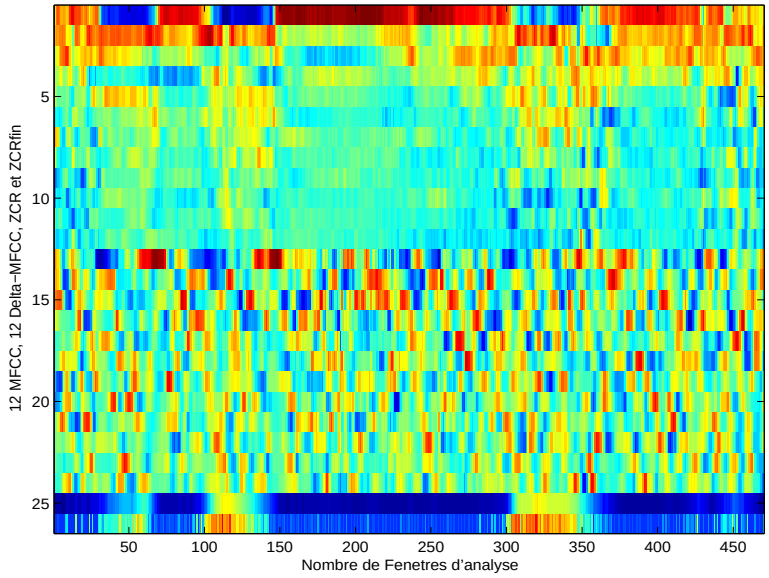


FIG. 3.10 – Tous les coefficients sus-cités pour la phrase "Je suis Robinson"

trois fenêtres se suivant (ceux des deux fenêtres précédentes et ceux de la fenêtre actuelle). Un exemple de Delta-MFCC est représenté figure 3.8.

3.3.3 Taux de passage par zéro (Zeros-crossing rate ou ZCR)

Le taux de passage par zéro (ZCR) représente le nombre de fois que le signal, dans sa représentation amplitude/temps, passe par la valeur centrale de l'amplitude (généralement zéro). Il est fréquemment employé pour des algorithmes de détection de section voisée/non voisée dans un signal. En effet, du fait de sa nature aléatoire, le bruit possède généralement un taux de passage par zéro supérieur à celui des parties voisées. Nous l'utiliserons pour représenter le caractère voisé du signal. Cependant, afin de rendre le modèle le plus réactif posible aux changements de phonème, et tout particulièrement aux attaques, nous avons pensé à introduire un taux de passage par zéro calculé uniquement sur la fin de la fenêtre d'analyse (ZCR fin). Nous obtenons ainsi un second paramètre de passage par zéro qui est moins sounmis à l'effet de moyennage induit par la longueur de la fenêtre. Cette évolution a été motivée par la nécessité de détecter le plus rapidement possible l'arrivée d'un nouveau phonème ou même l'arrivée du signal après un silence. Ce paramètre ne prend en compte que la fin du signal et donc uniquement la partie du signal qui est la plus récente. Ces deux paramètres ZCR sont reproportionnés pour entrer dans la dynamique de variation des coefficients MFCC. Ils ont ainsi un poid plus important dans la détection. La représentation de ces coefficients est figure 3.9.

Pour résumer ce paragraphes sur les coefficients utilisés pour le suivi, voici la figure représentant tous ces coefficients en couleurs en fonction du temps (figure 3.10). Les 12 premiers sont les MFCC, les 12 suivants sont les Delta-MFCC et les 2 derniers les ZCR.

3.4 L'apprentissage

L'apprentissage est la phase qui permet de donner aux paramètres du modèle les valeurs adéquates pour la seconde phase, celle qui nous intéresse : le suivi. Les étapes principales que nous allons détailler sont données figure 3.11. Cette apprentissage est la base même de la robustesse des modèles HMM. En effe, à partir du moment où il est bien adapté, il permet de suivre n'importe quel signal si on en a déjà quelques exemples. On a ainsi juste à décrire le signal par un modèle HMM.

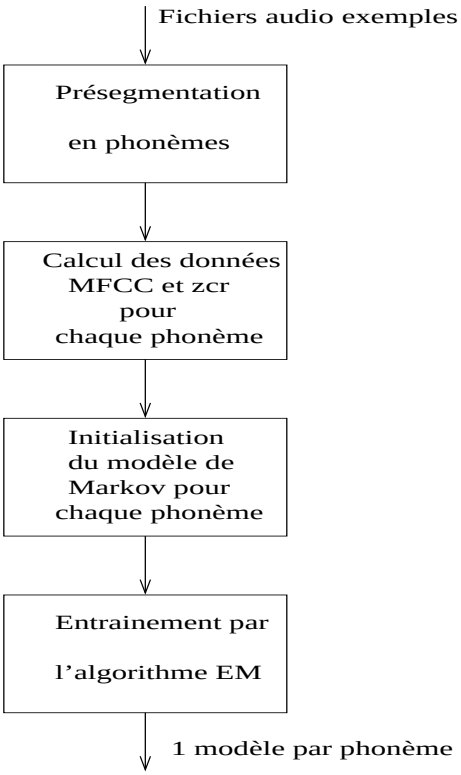


FIG. 3.11 – Etapes de la génération des modèles bas-niveau

3.4.1 Présegmentation

Pour entraîner les modèles de phonèmes, il faut disposer d'exemples segmentés, c'est à dire de fichiers audio du locuteur dont la voix va être suivie comportant assez de mots pour contenir tous les phonèmes qu'il pourrait utiliser. Plus les exemples seront nombreux, plus le modèle sera robuste.

Il existe divers logiciels libres (comme Mbrolign) pour segmenter un fichier audio en phonèmes mais ils font beaucoup d'erreurs qu'il faut corriger "à la main". Le logiciel Mbrolign ne fonctionne que pour une fréquence d'échantillonnage de 16 kHz. C'est pourquoi par la suite nous avons utilisé des fichiers wav échantillonnés à 16kHz pour les tests. Les paramètres utilisés (MFCC et zero-crossing rates - cf le paragraphe précédent-) sont ensuite calculés à partir de cette segmentation considérée comme la référence par la suite.

Le nombre d'exemples pour chaque phonème pour l'une des voix

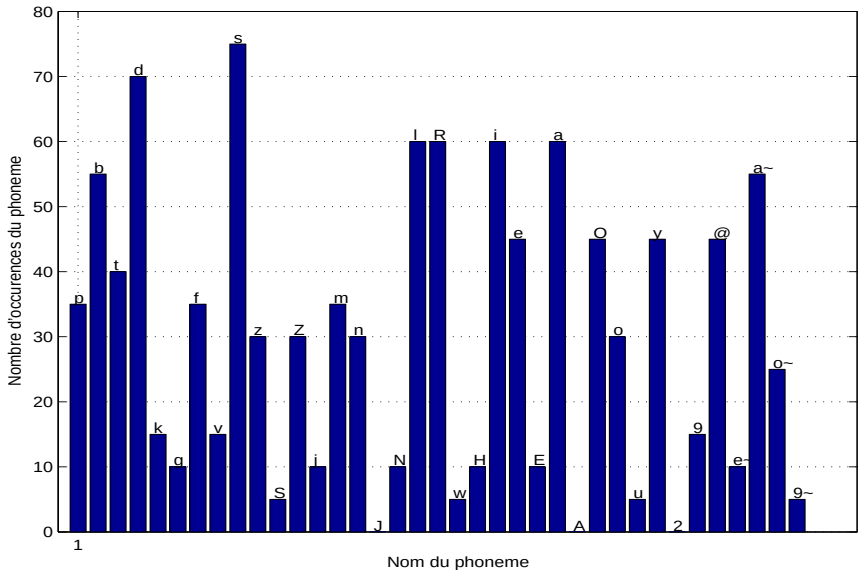


FIG. 3.12 – Nombre d'exemples des phonèmes

sur lesquelles nous avons travaillé est donné sur la figure 3.12. L'intérêt de disposer d'une base de données suffisamment fournie (tous les phonèmes) permet de segmenter automatiquement des exemples nouveaux par la suite en utilisant le modèle HMM lui-même.

3.4.2 Apprentissage Bas niveau

Il s'agit d'affiner les modèles en entraînant leurs paramètres. Il est possible d'aboutir à un modèle $\lambda = (A, B, \Pi)$ qui maximise la probabilité $p(o|\lambda)$ qui sert à définir la distance appelée "vraisemblance" (likelihood). La vraisemblance est définie par son logarithme :

$$f(o, path, \lambda) = log(P(o|\lambda)) \tag{3.8}$$

Pour ce faire, une procédure itérative est utilisée, dans notre cas, il s'agit de l'algorithme EM (Expectation-Maximization) connu aussi sous le nom d'algorithme Baum-Welch. Une autre possibilité consiste à utiliser l'algorithme de Viterbi qui se contente de calculer la vraisemblance uniquement pour le chemin le plus probable alors que l'algorithme EM prend en compte tous les chemins possibles.

L'initialisation

Dans un premier temps, les centres des gaussiennes sont initialisés par l'algorithme K-means. Chaque extrait de signal, correspondant à un phonème est découpé en 2 ou 3 segments de même taille, selon le nombre d'états donnés.

Le vecteur de probabilité d'état est initialisé avec une probabilité très forte pour le premier état mais des probabilités non-nulles pour les états suivants. Il est constaté qu'après une itération, ce vecteur est réestimé avec une probabilité maximale (égale à 1) pour le premier état.

La matrice A des transitions entre états est initialisée au hasard, de forme triangulaire supérieure. La somme sur chaque ligne doit toujours être égale à 1.

L'entraînement bas-niveau

L'algorithme EM est un algorithme de ré-estimation itératif. Il cherche à maximiser les probabilités de génération. Pour cela, sont associés aux états, aux transitions et aux observations, le nombre de fois où ils sont utilisés pour tous les exemples à disposition et tous les chemins susceptibles de générer les séquences d'observation, pondéré par la probabilité du chemin. Les deux étapes de l'algorithme sont les suivantes :

1^{re} étape : "expectations".

Pendant la première étape de l'algorithme EM, des statistiques sur les transitions (vecteur π : probabilité des états à t=1 et les transitions ultérieures -matrice a-), ainsi que les probabilités d'observations sont calculées :

$$\pi_i = \frac{\text{nombre de passages par l'état i à t=1}}{\text{nombre d' itérations total}} \tag{3.9}$$

$$a_{ij} = \frac{\text{nombre de transitions de l'état i vers l' état j}}{\text{nombre de transitions sortant de l' état i}} \tag{3.10}$$

$$b_j(k) = \frac{\text{nombre de passages dans l'état j observant un vecteur v}}{\text{nombre de passages dans l' état j}} \tag{3.11}$$

Ces statistiques peuvent être calculées grâce à des variables intermédiaires γ et ξ :

$$\text{(nombre de passages par l' état i à t=1)} = P(q_1 = i \mid O, \lambda) = \gamma_1(i) \quad (3.12)$$

$$\text{(nombre de passages dans l' état i)} = \sum_{t=1}^{T} P(q_t = i \mid O, \lambda) = \sum_{t=1}^{T} \gamma_t(i) \quad (3.13)$$

(nombre de passages par l' état j en observant le vecteur $o_t = v$) =

$$\sum_{t=1}^{T} P(q_t = i, o_t = v \mid O, \lambda) = \sum_{t=1 \ avec \ o_t=v}^{T} \gamma_t(i) \quad (3.14)$$

(nombre de transitions de l'état i vers l'état j) =

$$\sum_{t=1}^{T} P(q_t = i, q_{t+1} = j \mid O, \lambda) = \sum_{t=1}^{T} \xi_t(i, j) \quad (3.15)$$

Remarque : comme on entraîne sur K exemples audio, il faut sommer les espérances sur les différents exemples (cela est fait à la seconde étape de l'algorithme). Pour un souci de clarté, nous omettons cette somme. Les formules pour calculer les variables γ et ξ, pour un modèle de multi-gaussiennes et pour un seul exemple sont données par :

$$\begin{aligned} \gamma_t(j,k) &= P(q_t = j, G_t = k \mid O, \lambda) \\ &= \gamma_t(j) P(G_t = k \mid q_t = j, O, \lambda) \\ &= \gamma_t(j) \left(\frac{c_{jk} \, \mathcal{N}(o_t; \mu_{jk}, U_{jk})}{\sum_{k=1}^{M} c_{jk} \, \mathcal{N}(o_t; \mu_{jk}, U_{jk})} \right) \end{aligned} \quad (3.16)$$

$$\text{où } \gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{\sum_i \alpha_t(i)\beta_t(i)} \quad (3.17)$$

avec

$$\alpha(j,t) = P\Big(q(t){=}j \mid O{=}v(1:t)\Big) = P\Big(q(t){=}j, O{=}v(1:t) \mid \lambda\Big) \quad (3.18)$$

$$\beta(j,t) = P\Big(O{=}v(t+1:T) \mid q_t{=}j\Big) \quad (3.19)$$

Les variables α et β sont calculées par l'algorithme forward-backward, détaillé ci-dessous. Il permet de considérablement réduire le nombre d'opérations pour calculer la vraisemblance.

2^{me} étape : étape "maximization".

Cette étape ré-affecte les paramètres du modèle et calcule la nouvelle vraisemblance. Tant que la vraisemblance varie plus que le seuil fixé (10^{-4} en général), ces 2 étapes sont répétées. Le choix de ce seuil n'est pas évident puisqu'il ne doit pas être trop petit ni trop grand. Il faut éviter le surapprentissage d'un exemple, c'est-à-dire que le modèle devient trop précis sur un exemple particulier et ne peut pas suivre un autre exemple des mêmes phonèmes. De même, il faut qu'il soit suffisamment petit pour ne pas donner un modèle trop permissif, qui risquerait de reconnaître de mauvais phonèmes. Les formules de ré-estimation des paramètres des multi-gaussiennes sont, en partant de (3.16) :

$$c_{ij} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)}{\sum_{t=1}^{T} \gamma_t(j)} \; 1 \leq j \leq N, 1 \leq k \leq M \tag{3.20}$$

$$\mu_j = \frac{\sum_{t=1}^{T} o_t \gamma_t(j,k)}{\sum_{t=1}^{T} \gamma_t(j)} \; 1 \leq j \leq n, 1 \leq k \leq m \tag{3.21}$$

$$U_j = \frac{\sum_{t=1}^{T} (o_t - \mu_j)(o_t - \mu_j)' \gamma_t(j,k)}{\sum_{t=1}^{T} \gamma_t(j)} \; 1 \leq j \leq N, 1 \leq k \leq M \tag{3.22}$$

L'algorithme forward-backward

Cet algorithme est très important, il est central dans l'utilisation pratique des HMM. Il consiste à calculer les α et β d'une manière très astucieuse ce qui réduit considérablement le nombre d'opérations comme nous allons le voir. Le but est de calculer efficacement $P(O|\lambda)$ soit : étant donnés les paramètres d'un modèle λ, quelle est la probabilité d'observer la séquence O ? Cette probabilité peut être obtenue en sommant sur tous les chemins possibles Q la probabilité $P(O, Q|\lambda)$:

$$P(O|\lambda) = \sum_{Q} P(O, Q|\lambda)$$

$$= \sum_{q_T} \cdots \sum_{q_2} \sum_{q_1} b_{q_T}(o_T) a_{q_{T-1} q_T} b_{q_{T-1}}(o_{T-1}) \cdots b_{q_2}(o_2) a_{q_1 q_2} b_{q_1}(o_1) \pi_{q_1} \tag{3.23}$$

La complexité de ce calcul est donc de : $n^{2^{|O|}} * |O|$ où n est le nombre d'états des chemins considérés et $|O|$ le nombre de vecteurs d'observations.

L'algorithme forward

L'idée consiste à sommer toutes les informations au temps $t = 1$ dès qu'elles sont connues, faire de même à $t = 2$ et ainsi de suite. Nous défissons à $t = 1$:

$$alpha_1(j) = p(o_1, q_1 = j|\lambda) = b_j(o_1)\pi_j \tag{3.24}$$

A $t = 2$, nous avons :

$$\alpha_2(j) = p(o_1, o_2, q_2 = j|\lambda) = b_j(o_2)\sum_{i=1}^{N} a_{ij}\alpha_1(i) \tag{3.25}$$

A chaque nouveau temps jusqu'à $t = T$, nous avons :

$$\alpha_t(j) = p(o_1, \ldots, o_t, q_t = j|\lambda) = b_j(o_t)\sum_{i=1}^{N} a_{ij}\alpha_{t-1}(i) \tag{3.26}$$

Au final, au temps $t = T$, nous obtenons :

$$p(O|\lambda) = p(o_1, \ldots, o_t, q_t = j|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{3.27}$$

Le caractère 'forward' apparait clairement dans les équations ci-dessus, l'itération se déplaçant dans les sens des temps croissants. La même chose est réalisée en sens inverse avec lálgorithme dit 'backward'. La complexité de l'algorithme forward est fortement avantageuse en coût de calculs puisquélle est de l'ordre de $|O|n^2$

L'algorithme backward

L'équation (3.23) peut être tronquée par une récursion qui se déplace en arrière dans le temps :

$$p(O|\lambda) = \sum_{q_1} \pi_{q_1} b_{q_1}(o_1) \sum_{q_2} a_{q_1 q_2} b_{q_2}(o_2) \ldots \sum_{q_T} a_{q_{T-1} q_T} b_{q_T}(o_T) \tag{3.28}$$

La variable β s'écrit dans ce cas :

$$\beta_t(i) = p(o_{t+1}, o_{t+2}, \ldots, o_T|q_t = i, \lambda) \tag{3.29}$$

L'équation 3.28 est alors calculée ainsi :

1. Initialisation
$$\beta_1(i) = 1,\ 1 \leq i \leq N \tag{3.30}$$
2. Propagation
$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}\beta_{t+1}(i)) \tag{3.31}$$

3. Terminaison

$$p(O|\lambda) = \sum_{i=1}^{N} \pi_i b_i(o_1 \beta_1(i) \tag{3.32}$$

Cet algorithme est utilisé pour la segmentation et l'entraînement mais en général il ne l'est pas pour la reconnaissance à cause de son caractère anti-causal. Ensuite la variable 'forward-backward' γ est définie ainsi :

$$\gamma_t(j,k) = P(q_t = i|O,\lambda) = \frac{P(O, q_t = i|\lambda)}{P(O|\lambda)} \tag{3.33}$$

L'expression de γ en fonction de α et β a été donnée (équation (3.17)).

3.4.3 Apprentissage Haut niveau

L'apprentissage haut-niveau fonctionne de la même manière que l'apprentissage bas-niveau. Plusieurs types d'entraînement haut-niveau sont possibles. Un premier consiste à ne ré-estimer que les probabilités de transition haut-niveau, c'est à dire les transitions entre phonèmes et à laisser invariantes celles internes à chaque modèle bas-niveau. Les autres grandeurs utilisées $(mu, U, mixmat, \Pi)$ étant laissées invariantes. Un deuxième type d'entraînement plus conséquent consiste à pouvoir tout ré-estimer. Cela signifie, entre autres, que l'on tient mieux compte dans ce cas des contextes des phonèmes, i.e. chaque modèle de phonèmes peut être modifié par ses plus proches voisins.

Remarque : nous n'avons pas parlé ici des g-states car ils servent uniquement au suivi. De plus, comme dit précédemment, nous ne pouvons pas les entraîner en haut-niveau de manière satisfaisante (problème des exemples erronés).

3.5 Le suivi

Le suivi est le but ultime de notre recherche. Le système que nous cherchons à développer doit répondre à certains critères de temps. Il doit répondre aux contraintes du temps réel tout en restant assez robuste. Contrairement à la reconnaissance de mots isolés ou inclus dans des phrases, nous cherchons à suivre le plus finement et le plus rapidement possible les phonèmes d'un flot de parole. Nous devons donc avoir une vitesse de calcul élevée et

une réactivité du modèle la plus grande possible. Pour répondre a cette contrainte du temps réel, il faut utiliser les algorithmes de calcul les plus adaptés et trouver quels sont les paramètres qui amélioreront la réactivité. Nous avons trouvé dans la littérature plusieurs façons de réaliser le suivi. En voici deux qui sont fréquemment utilisées :

3.5.1 L'algorithme Probabilité "Avant" (Forward Probability)

Cet algorithme est la première partie de l'algorithme Forward-Backward décrit plus haut. On défini :

$$\alpha_n(j) = p(q_j^n, X_1^n) \tag{3.34}$$

représentant la probabilité que le modèle HMM ait généré la séquence partielle X_1^n en se trouvant dans l'état q_j à l'instant n. Etant donné que (sans hypothèses particulières) :

$$\begin{aligned} p(q_j^n, X_1^n) &= \sum_{k=1}^{J} p(q_k^{n-1}, q_j^n, X_1^{n-1}, x_n) \\ &= \sum_{k=1}^{J} p(q_k^{n-1}, X_1^{n-1}) p(q_j^n, x_n | q_k^{n-1}, X_1^{n-1}) \end{aligned} \tag{3.35}$$

avec J le nombre d'états total, cette probabilité "avant" peut être calculée par la récurrence "avant" :

$$\alpha_n(j) = \sum_k \alpha_{n-1}(k) p(q_j^n, x_n | q_k^{n-1}, X_1^{n-1}) \tag{3.36}$$

où la somme porte sur l'ensemble des prédécesseurs possibles q_k de l'état q_j. L'initialisation de cette récurrence est donnée par :

$$\alpha_1(j) = \Pi_j \tag{3.37}$$

où Π représente la distribution initiale des états du modèle HMM. Dans cet algorithme, on prend en compte tous les chemins possibles pour arriver à un état pour calculer la probabilité de cet état. Ceci demande énormément de calcul et surtout la multiplication de probabilités qui peuvent entrainer des problèmes numériques (underflow). Une autre solution, parfois considérée comme plus simple, consiste à ne prendre en compte que le meilleur chemin à travers le modèle. Cette approche s'appelle l'approche Viterbi.

3.5.2 L'algorithme Viterbi

En plus d'éviter les problèmes numériques, cette méthode permet de dévoiler la meilleure séquence d'états à travers le modèle et, par conséquent, la segmentation optimale des observations selon les états du modèle. L'algorithme Viterbi résulte d'une simplification de la récurence avant. On remplace toutes les sommes par une fonction de maximum. Selon l'approximation Viterbi, la récurence 3.36 devient :

$$p(q_j^n, X_1^n) = \max_k \Big[p(q_k^{n-1}, X_1^{n-1}) p(q_j^n, x_n | q_k^{n-1}, X_1^{n-1}) \Big] \tag{3.38}$$

Où $p(q_j^n, X_1^n)$ représente la probabilité du meilleur chemin partiel allant de l'état initial q_1 à l'état q_j du modèle HMM en ayant émis les n premiers vecteurs X_1^n de la séquence X. On peut donc aussi écrire :

$$p(q_j^n, X_1^n) = \max_k \Big[p(q_k^{n-1}, X_1^{n-1}) p(q_j | q_k) \Big] p(x_n | q_j) \tag{3.39}$$

et donc :

$$\alpha_n(j) = \max_k \Big[\alpha_(n-1)(k).a_{kj} \Big] p(x_n | q_j) \tag{3.40}$$

avec a_{kj} la probabilité de transition de l'état k à l'état j

3.5.3 Le suivi à Posteriori

Les deux algorithmes précédemment cités sont tous les deux utiles pour trouver la probabilté d'être dans un état pour une observation donnée et donc pour rechercher quel est l'état qui convient le mieux à une observation connaissant les états précédents. Nous préfèrerons l'algorithme Viterbi qui est moins coûteux en calcul et permet d'avoir directement le chemin entier de plus grande probabilité à chaque itération.

Le chemin suivi à postériori est trouvé grâce à l'algorithme Viterbi appliqué à tout le signal à suivre. Il est donc nécessaire de connaître tout ce signal. Ce n'est pas un suivi en temps réel qui est réalisé mais un suivi qui nous donne le chemin de plus grande probabilité pour un exemple donné étant donné un modèle HMM pré-entrainé. C'est en quelque sorte le chemin auquel on cherche à se rapprocher quand on suit un locuteur en temps réel. La segmentation en phonème réalisée est bien sûr meilleure que lors du calcul du chemin temps réel. Ce chemin à postériori est un indicateur du chemin utopique que l'on cherche à atteindre. Si ce chemin n'est pas bon, le chemin calculé en temps réel ne sera pas très bon non plus.

Nos simulations nous permettent de calculer ce chemin en même temps que celui temps réel. On obtient ainsi une référence à laquelle on veut s'approcher. Cette référence va être un des outils utilisé pour affiner les paramètres du modèle : nombre de coefficients, type des coefficients...

3.5.4 Le suivi en Temps Réel

Lors du suivi en temps réel, le suivi que l'on veut affiner, on ne connaît que le signal passé. On ne peut donc utiliser que les vecteurs de coefficients passés. On utilise l'algorithme Viterbi pas à pas. Pour chaque fenêtre d'analyse, l'état le plus probable est détecté puis on recommence. On applique l'algorithme Viterbi à l'état q_j pour trouver le numéro de l'état suivant, l'état qui a la plus grande probabilté d'après le passé. On calcule d'abord la vraisemblance de l'observation : obslik

$$obslik_n(j) = P(x_n|q_j)$$

puis on trouve α_n à partir de α_{n-1} et de $obslik_n$

$$\alpha_n(j) = \max_k \Big[\alpha_(n-1)(k).a_{kj}\Big] obslik_n(j)$$

Et enfin, on normalise le vecteur alpha(n) pour avoir de vrais probabilités et éviter les erreurs dues au conditionnement numérique :

$$\alpha_n(j) = \frac{\alpha_n(j)}{\sum_{k=1}^{J} \alpha_n(k)}$$

On obtient donc un chemin pas à pas qui tend à s'approcher du chemin à postériori présenté précédemment. Nous cherchons à nous en approcher le plus possible. Les tests ont été réalisés dans cette optique là : après avoir maximisé les performances du chemin à postériori, nous avons cherché à maximiser les performances du chemin temps réel. Voici le type de figure obtenue (figure 3.13) :

Le schéma du haut représente les deux chemins, à postériori et temps réel, entrelacés. Les courbes représentent l'évolution des états les plus probables en fonction du signal émis. Le chemin en temps réel étant en bleu (ou trait pointillé) et le chemin à postériori en rouge (ou trait plein). Nous pouvons bien voir le chemin temps réel qui s'approche du chemin à postériori mais s'en écarte parfois localement. Le schéma du bas représente le signal émis et la segmentation finale obtenue grâce au chemin à postériori. Les figures de test seront toutes agencées de cette même façon.

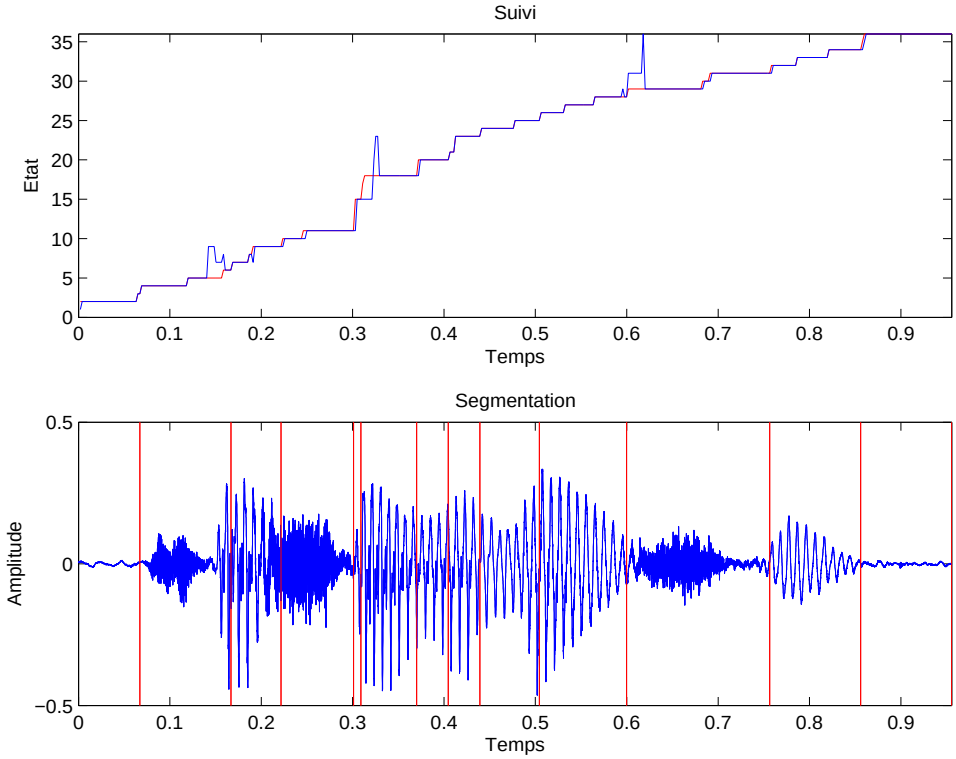


FIG. 3.13 – Suivi et segmentation de la phrase "Je suis Robinson"

Nous avons effectué de nombreux tests de simulation tout au long de l'étude. Etant donné le nombre assez conséquent de paramètres pouvant varier, les tests ont été orientésse sont axés sur plusieurs axes. Les objectifs principaux étant de réussir à affiner au mieux les coefficients décrivant le signal (MFCC et ZCR) et de créer un modèle HMM le plus adapté.

Chapitre 4

Tests et Résultats

Les tests ont été réalisés grâce à Matlab. Nous avons implémenté un programme de création du modèle HMM ainsi qu'un système de simulation du suivi temps-réel. L'enregistrement des voix à suivre en temps réel a été réalisé en plusieurs exemplaires grâce au concours des trois acteurs de la pièce "Retour Définitif et Durable de l'Etre Aimé" écrite par Olivier Cadiot. Ces acteurs (2 hommes et une femme) ont lu différents textes de la pièce, normalement d'abord, puis ensuite plus ou moins vite et plus ou moins lié. Nous avons utilisé ces enregistrements comme base d'entrainement du modèle bas niveau (après segmentation en phonèmes), comme base d'entrainement du modèle haut niveau ainsi que comme données de tests.

4.1 Tests sur les paramètres utilisés

Le bon choix du jeu de coefficients utilisé est crucial pour obtenir un bon suivi de la voix. Il est important aussi d'avoir un bon nombre d'exemples d'entrainement pour chaque phonème (Bas niveau) mais aussi pour chaque phrase (Haut niveau). Dans le cas contraire, le modèle se focalise sur la prosodie et le rythme d'un exemple donné et n'est plus robuste au suivi de tout type d'exemples de la même phrase. Sauf mention contraire dans les tests réalisés, nous avons utilisé les paramètres suivants :

- Fréquence d'échantillonnage 16kHz. Cette fréquence d'échantillonnage a été dictée par le logiciel de segmentation MBROLIGN.
- Tests en monolocuteur. Les exemples d'entrainement et les exemples de tests sont des enregistrements du même locuteur. Les tests por-

tant sur la capacité d'un même système à suivre plusieurs locuteurs différents ont été commencés. Nous en parlerons plus loin.
- 2 mixtures de Gaussiennes. Nous permettons ainsi la possibilité d'avoir deux prononciations différentes d'un même phonème, une pour chaque mixture de gaussiennes. Ce nombre pourrait être augmenté avec le nombre d'exemples disponibles (cf suite).
- Des fenêtres d'analyse de 256 échantillons. Nous prenons ainsi des fenêtres d'analyse de 16ms.
- Suivi sur une phrase non utilisée pour l'apprentissage. Il est bien sûr important de prendre une phrase encore inutilisée pour réaliser le suivi. Dans le cas contraire, le modèle aura déjà été entrainé sur cette phrase et les résultats seraient biaisés. Dans le cadre de ces tests, nous avons pris la phrase "Je suis Robinson" énoncée plusieurs fois par Laurent Poitrenaux.
- Entrainement Haut Niveau des transitions inter-phonèmes avec 4 phrases. L'entrainement Haut niveau n'a été réalisé que sur les transitions entre phonèmes. On garde ainsi une base de données constante de modèles HMM de phonèmes pour toutes les phrases. On n'ajoute que les transitions entre les phonèmes pour créer le modèle entier.

Il est, en particulier, très important de prendre une phrase non utilisée pour l'entraînement lors du suivi. Voici les différents tests effectués en utilisant le modèle et les paramètres sus-cités :

4.1.1 Quels coefficients utiliser pour bien décrire les phonèmes ?

Les différents tests que nous avons réalisés ont surtout porté sur le nombre de MFCC pris en compte. Plusieurs jeux de coefficients ont été testés pour montrer leur pertinence.

Tout d'abord, en n'utilisant que les deux ZCR (cf figure 4.1), nous avons obtenu un suivi correct mais un alignement médiocre. C'est à dire que le système arrive à suivre la phrase mais pas en temps réel, plutôt en temps différé. Les erreurs de réactivité sont énormes dans ce cas et la segmentation très mauvaise. Par exemple, au phonème 'b' de l'alphabet phonétique Sampa, le système associe un 'bin'. Ceci prouve bien la pertinence de l'utilisation de ces paramètres ZCR. Leur capacité à décrire le caractère voisé du signal donne déjà une bonne description temporelle du signal. Cependant elle n'est pas assez précise et ne permet pas de séparer tous les phonèmes.

L'ajout de coefficients MFCC permet cette séparation (cf figure 4.3).

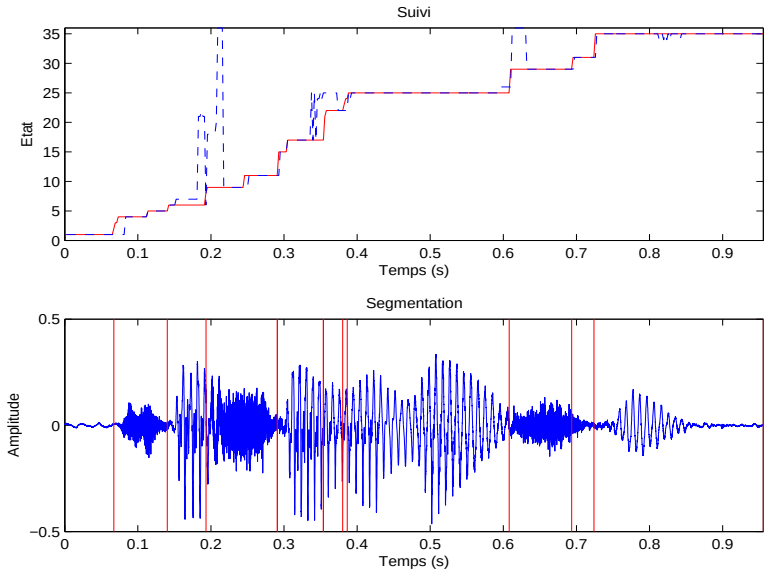


FIG. 4.1 – Suivi avec 2 ZCR pour la phrase "Je suis Robinson"

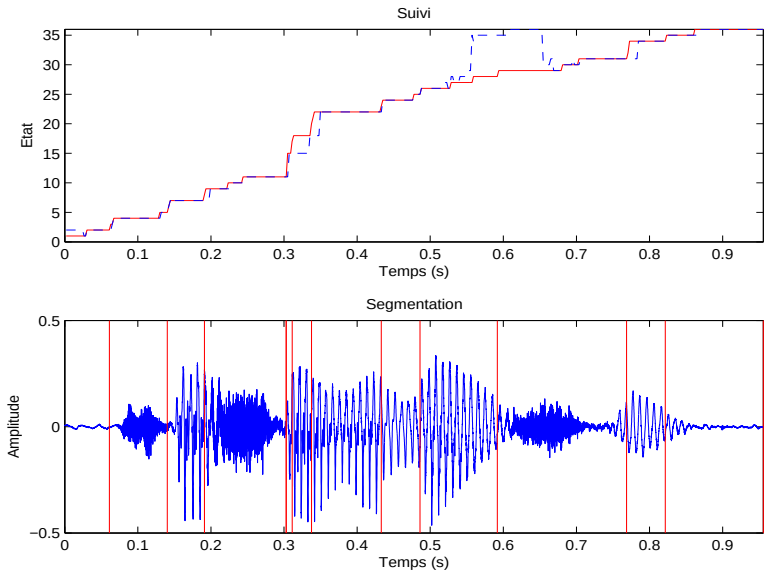


FIG. 4.2 – Suivi avec 2 ZCR et 8 MFCC pour la phrase "Je suis Robinson"

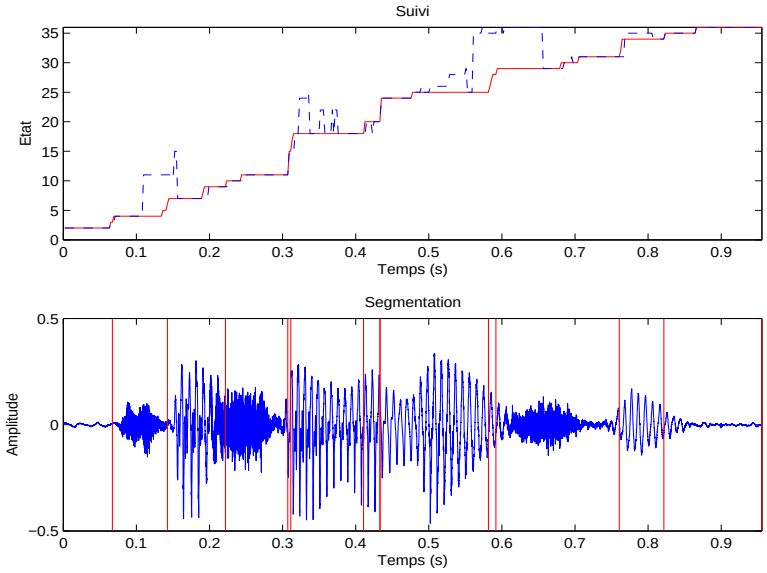


FIG. 4.3 – Suivi avec 2 ZCR et 12MFCC pour la phrase "Je suis Robinson"

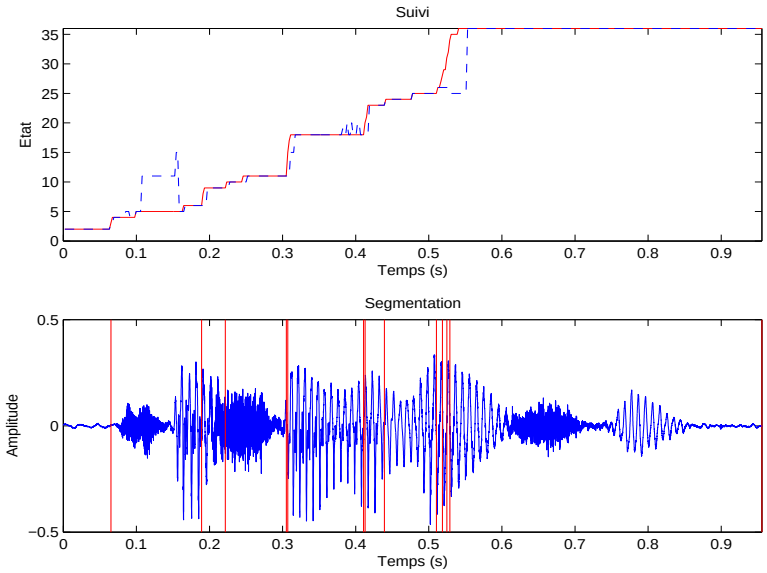


FIG. 4.4 – Suivi avec 2 ZCR, 12MFCC et 12 Delta-MFCC pour la phrase "Je suis Robinson"

Ces paramètres décrivant pleinement l'évolution des formants dans la voix rendent l'analyse plus fine et donc le suivi meilleur. Nous avons testé le système avec plusieurs nombres de MFCC : 5, 8, et 12 coefficients... Il apparaît qu'après tous ces tests nous n'avons pas besoin de beaucoup de MFCC. Seulement 8 suffiraient ce qui semble peu mais peut être expliqué. Tout d'abord, les variations de ces coefficients sont beaucoup plus faibles dans les coefficients d'ordre élevé. Leur présence ne serait donc pas nécessaire. Elle apporterait une précision faible. Mais de plus, la précision nécessaire à un bon suivi n'est pas la précision maximum. Il faut avoir une marge de manoeuvre assez large pour réussir à suivre un voix dans toutes les situations possibles. Cette précision d'analyse serait atteinte avec huit coefficients MFCC et 2 coefficients ZCR.

Enfin, l'ajout des coefficients Delta-MFCC devrait rendre la séparation plus fine (cf figure 4.4). Ces paramètres décrivent en effet l'évolutivité du langage et prennent donc en compte la notion de temps. Cependant, les résultats que nous obtenons montrent que dans le cadre du suivi temps réel, ces coefficients rendent la description trop fine et donc le suivi moins bon.

Sur les figure 4.1 à 4.4, sont représentés les suivis pour 2, 10, 14 et 26 coefficients. Elles montrent clairement que la segmentation et le suivi sont meilleurs avec 10 coefficients. Cependant, il reste un problème de suivi en temps réel, le suivi qui nous intéresse, pour le "s" final de la phrase testée. Il est, en effet, assimilé au début à un silence. Ceci provient du fort poid de suivi des ZCR qui sont très proches en valeur de celle calculée pour un silence. Nous verrons plus loin dans le chapitre 5, Limites et Evolutions, comment nous pourrions nous affranchir de cette erreur en ajoutant un nouveau paramètre. Nous avons comparé la segmentation avec une segmentation de référence réalisée à l'oreille et visuellement. L'erreur moyenne faite avec un suivi à posteriori est inférieure à 25ms. Avec le suivi en temps réel, les résultats sont biaisés par le problème précédent de détection de silence. Cependant, si nous réussissions à nous en affranchir, la figure 4.2 montre bien que les deux suivis sont très proches et prouve qu'alors la précision du suivi temps réel serait inférieure aussi à 25ms.

De plus, nous avons aussi testé des cas de suivi avec moins de coefficients MFCC et les Delta-MFCC (par exemple 2 ZCR, 5 MFCC et 5 Delta-MFCC). Les résultats obtenus prouvent bien que l'utilisation des Delta-MFCC fausse le suivi et la segmentation. Nous avons aussi testé le suivi sur d'autres phrases de la pièce pour valider nos résultats pour tous les phonèmes. Les résultats ont été positifs. Tous ces résultats ont été réalisés avec les coefficients MFCC normalisés. Cependant, il n'était pas sûr que cette normalisation soit justifiée. C'est pourquoi nous avons testé différents

suivis avec des coefficients normalisés et des coefficients non normalisés.

4.1.2 Discussion sur la "normalisation" des coefficients MFCC

A la fin du calcul des MFCC, nous avons effectué une "normalisation" des vecteurs d'observations. Cette étape consiste à centrer et à univarier les vecteurs. La nécessité de ce traitement peut être discutée. Imaginons qu'un coefficient soit biaisé et diverge anormalement. Il entraînera alors tous les autres dans son biais si l'on centre le vecteur entier et qu'on l'univarie. Ceci peut poser un problème pour le suivi car alors tous les coefficients seraient mauvais. Le système ne pourrait plus se caler sur les coefficients non biaisés. Cependant les différents tests (cf figure 4.5 et 4.6) que nous avons réalisés montrent sans erreur possible que cette "normalisation" améliore le suivi et améliore surtout la réactivité du suivi. La rapidité de détection est bien plus grande quand les coefficients utilisés sont normalisés. En effet, la normalisation réalisée replace tous les vecteurs coefficients sur la même échelle et donc privilégie la prise en compte de leurs variations de forme dans le temps plutôt que leur amplitude relative.

En annexe, plusieurs exemples de suivis sont représentés en fonction du nombre et de la nature des coefficients pour vérifier l'impact de la normalisation sur la précision du suivi.

4.1.3 Nombre d'exemples d'entrainement, limitations et obligations

Il est nécessaire, lors d'un entrainement de HMM avec multi-gaussiennes, d'avoir au moins autant d'exemples d'entrainement que de mixtures de gaussiennes. En effet, ces mixtures permettent de prendre en compte plusieurs façons de prononcer le même exemple et il faut connaître ces manières de prononcer. Cependant, il est toujours mieux d'avoir au moins le double d'exemples ou beaucoup plus pour bien affiner les centres de chaque gaussienne. Dans nos tests, nous avons pris 2 gaussiennes mais il est possible d'en prendre beaucoup plus pour élargir la reconnaissance et le suivi. Nous avons été obligé de nous limiter à ce nombre à cause de la taille de la base de données que nous avons segmentée. En effet, certains phonèmes n'ont que cinq exemples.

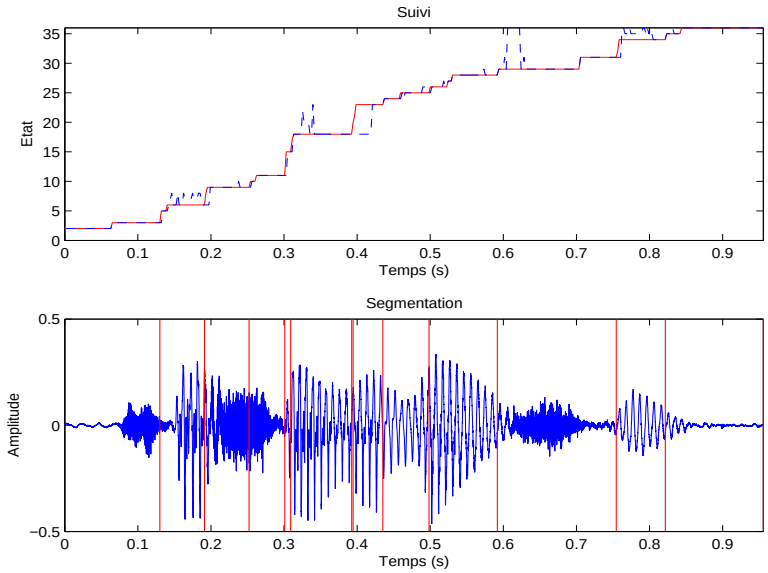


FIG. 4.5 – Suivi avec 10 coefficients (8 MFCC et 2 ZCR) non normalisés pour la phrase "Je suis Robinson"

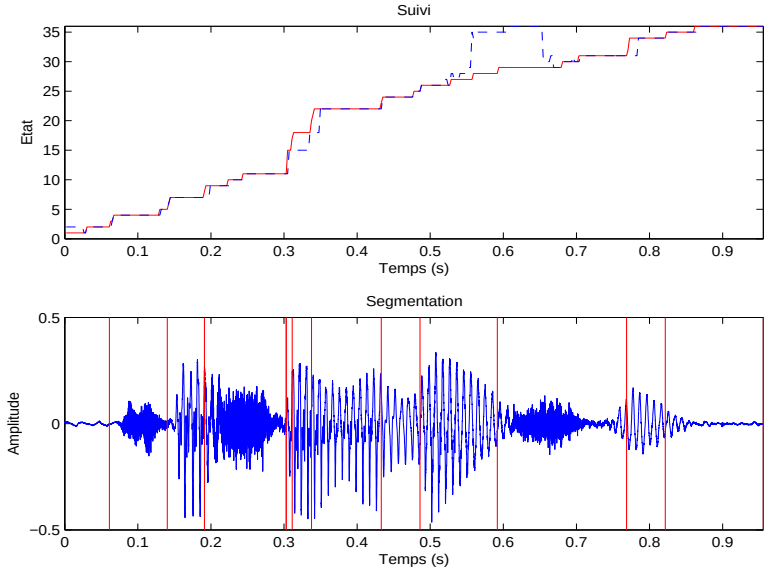


FIG. 4.6 – Suivi avec 10 coefficients (8 MFCC et 2 ZCR) normalisés pour la phrase "Je suis Robinson"

4.1.4 Tests sur la taille de recouvrement entre les fenêtres d'analyse (Overlap)

La taille du recouvrement entre les fenêtres d'analyse (Overlap) a été aussi un sujet de test. En effet, plus ce recouvrement est grand, plus la finesse d'analyse est grande. Ici on devrait pouvoir augmenter le recouvrement comme on veut car la taille de la fenêtre reste constante et la précision que l'on améliore et temporelle et sans rapport avec les coefficients utilisés. Nous avons testé le système avec des décalages de 32, 64 et 128 échantillons avec toujours une fenêtre d'analyse de 256 échantillons. Comme prévu, ils donnent de bons résultats pour le plus grand recouvrement. C'est à dire pour un décalage de 32 échantillons. La contrainte du temps réel lors de l'implémentation en langage C et jMax limitera ce recouvrement. Dans le cas présent, un décalage de 32 échantillons représente 2 ms ce qui est un peu petit pour réaliser les calculs nécessaires.

4.2 Apport de l'apprentissage

4.2.1 Bas-niveau

L'apprentissage bas-niveau a une importance primordiale puisqu'il est la base sur laquelle le suivi va s'appuyer ensuite. Nous avons parlé de la présegmentation, elle doit être la plus précise possible (et donc corrigée "à la main") pour que chaque extrait de signal audio représente le plus exactement possible le phonème auquel il correspond. Les points suivants permettent de comparer les résultats selon l'emploi qui est fait de l'apprentissage bas-niveau. Ces tests ont été réalisés sur la phrase : "Je suis Robinson, j'y suis." (dite par l'actrice Valérie Dashwood). Aucun entraînement haut-niveau n'est réalisé, les données ne sont pas normalisées. La figure et le tableau 4.7 sont la référence à laquelle seront comparés les résultats des tests suivants puisqu'il s'agit de la segmentation réalisée à la main. Le tableau associe aux phonèmes de la phrase suivie leurs indexes temporels respectifs en secondes et les droites verticales sur le signal montrent la segmentation du signal audio. Sur la figure 4.8 et celles qui suivront, le suivi réalisé par un algorithme Viterbi temps réel est en bleu (et en pointillé) et le suivi par l'algorithme Viterbi a posteriori est en rouge (et en trait plein).

- La figure et le tableau 4.8 montrent les résultats d'un suivi d'une phrase non-utilisée pour le bas-niveau.
- La figure et le tableau 4.9 montrent les résultats d'un suivi d'une phrase non-utilisée pour le bas-niveau mais 5 autres exemples de

cette phrase sont utilisés pour le bas-niveau en plus d'exemples d'autres phrases.
– La figure et le tableau 4.10 montrent les résultats d'un suivi d'une phrase utilisée pour le bas-niveau.

Analyse

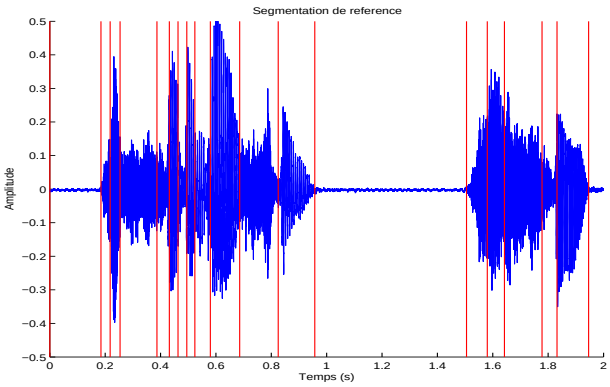
– Il apparaît que les pauses et les fricatives sont assez bien détectées quelque soit le cas considéré. Il semble donc que l'emploi des deux paramètres de taux de passages par zéro soient très efficaces. En effet, ils mesurent le caractère bruité des phonèmes, ils discriminent donc bien les pauses et les fricatives des autres phonèmes.
– Certains phonèmes se retrouvent superposés. Par exemple, le "e " est superposé au "s" suivant, dans tous les cas sauf le dernier suivi, où la phrase à suivre est utilisée dans l'apprentissage bas-niveau. Ces erreurs peuvent s'expliquer par le fait que les phonèmes sont dépendants du contexte dans lequel ils sont prononcés, c'est-à-dire les phonèmes précédent et suivant premiers voisins. Le triphone "b e∼ s" n'existe pas dans les exemples utilisés pour l'apprentissage bas-niveau (chacun de ces trois phonèmes sont appris en bas-niveau avec des contextes différents), ce qui expliquerait l'erreur constatée.
– Le suivi en temps réel approche précisèment le suivi a posteriori dans le dernier cas. Les pics du trajet temps réel montrent qu'avec l'algorithme Viterbi temps réel, le chemin le plus probable au temps t n'est pas forcément le même que celui a posteriori. Le suivi tente alors d'entamer un chemin qui par la suite va revenir à un état antérieur qui s'avère plus probable et le chemin va se racorder au chemin le plus probable a posteriori.

Remarque : Il faut noter le problème récurrent sur le "H" qui est difficile à segmenter à cause de sa durée et du bruit de la fricative "s" qui précède. Une solution consiste à définir un phonème hybride qui remplace "H" et "i" lorsqu'ils sont consécutifs.

4.2.2 Haut niveau

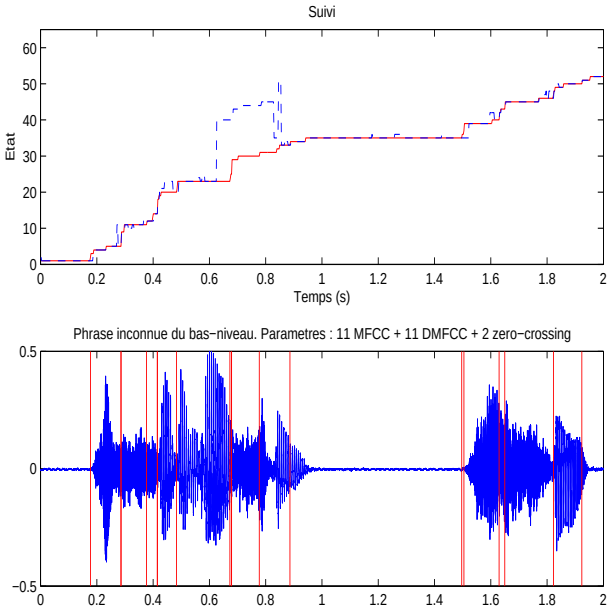
L'apprentissage haut-niveau se fait sur des exemples de la phrase à suivre, mais évidemment pas sur la phrase à suivre elle-même. Trois situations sont à envisager :

– Aucun apprentissage haut-niveau n'est réalisé : figure 4.11



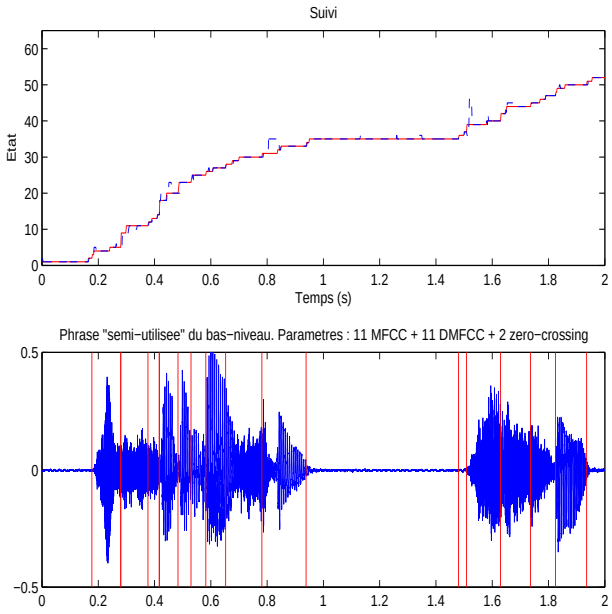
0.000	pause
0.185	Z
0.218	@
0.254	s
0.387	H
0.432	i
0.463	R
0.495	O
0.524	b
0.580	e ∼
0.686	s
0.825	o ∼
0.957	pause
1.505	Z
1.580	i
1.642	s
1.778	H
1.832	i
1.946	pause

FIG. 4.7 – Référence



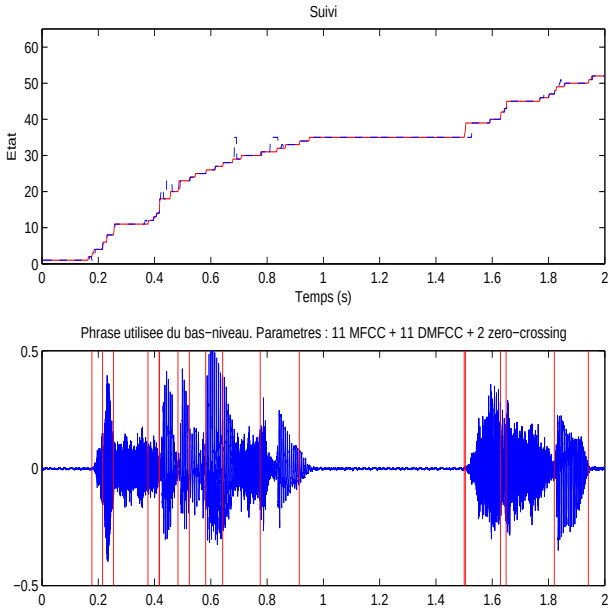
0.000	pause
0.177	Z
0.286	@
0.286	s
0.377	H
0.415	i
0.415	R
0.483	O
0.673	b
0.679	e∼
0.679	s
0.777	o∼
0.886	pause
1.496	Z
1.504	i
1.629	s
1.649	H
1.823	i
1.923	pause

FIG. 4.8 – suivi phrase non-utilisée pour le bas-niveau



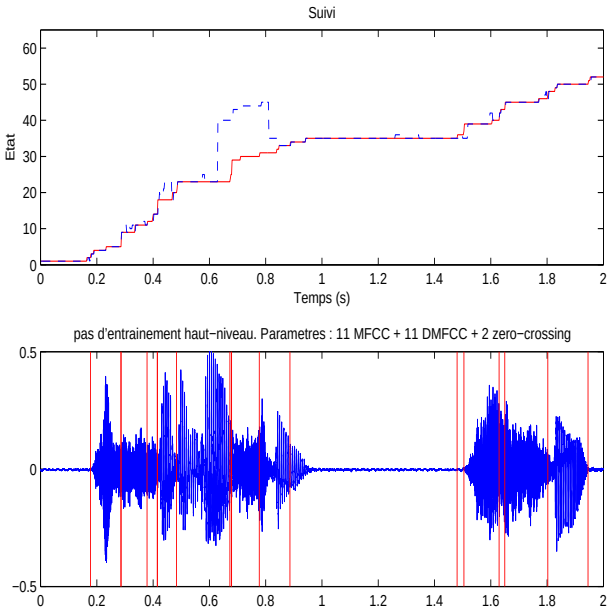
0.000	pause
0.177	Z
0.280	@
0.280	s
0.377	H
0.417	i
0.417	R
0.483	O
0.530	b
0.582	e ~
0.653	s
0.781	o ~
0.938	pause
1.480	Z
1.508	i
1.629	s
1.736	H
1.825	i
1.935	pause

FIG. 4.9 – suivi phrase "semi-utilisée" pour le bas-niveau



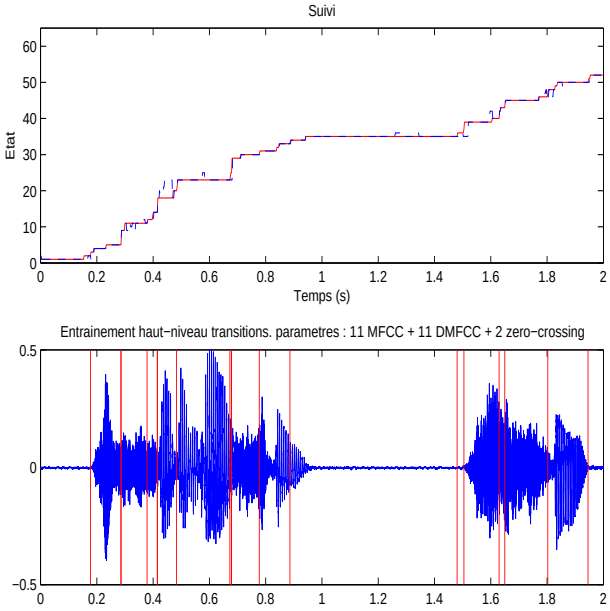
0.000	pause
0.177	Z
0.216	@
0.254	s
0.377	H
0.417	i
0.417	R
0.483	O
0.526	b
0.582	e ~
0.653	s
0.777	o ~
0.912	pause
1.500	Z
1.504	i
1.639	s
1.647	H
1.811	i
1.943	pause

FIG. 4.10 – suivi phrase utilisée pour le bas-niveau



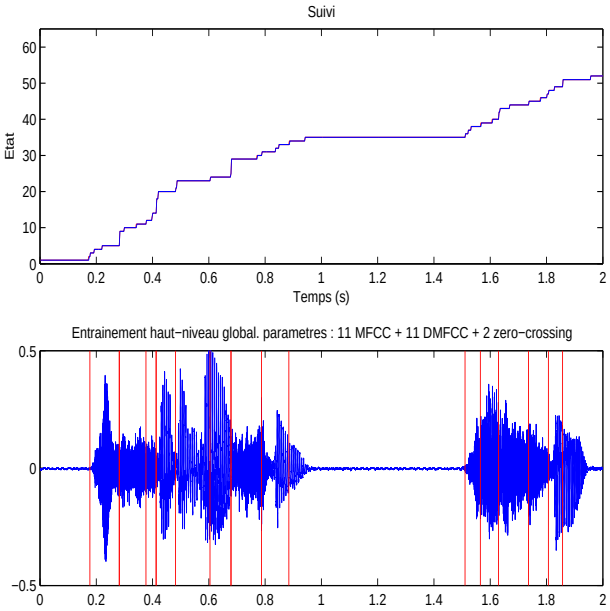
0.000	pause
0.177	Z
0.286	@
0.286	s
0.377	H
0.413	i
0.413	R
0.483	O
0.671	b
0.679	e ∼
0.679	s
0.785	o ∼
0.886	pause
1.519	Z
1.565	i
1.629	s
1.732	H
1.803	i
1.857	pause

FIG. 4.11 – suivi sans apprentissage haut-niveau.



0.000	pause
0.177	Z
0.286	@
0.286	s
0.379	H
0.415	i
0.415	R
0.483	O
0.673	b
0.679	e ∼
0.679	s
0.777	o ∼
0.886	pause
1.480	Z
1.504	i
1.629	s
1.649	H
1.803	i
1.946	pause

FIG. 4.12 – suivi avec apprentissage haut-niveau uniquement sur les transitions



0.000	pause
0.187	Z
0.260	@
0.260	s
0.425	H
0.425	i
0.453	R
0.498	O
0.530	b
0.590	e
0.645	s
0.771	o
0.908	pause
1.515	Z
1.549	i
1.621	s
1.647	H
1.815	i
1.915	pause

FIG. 4.13 – suivi avec apprentissage haut-niveau global

- Apprentissage haut-niveau uniquement sur les transitions entre phonèmes : figure 4.12
- Apprentissage haut-niveau global sur les matrices de transitions des phonèmes (le bas-niveau est affecté) ainsi que sur tous les autres paramètres du modèle : figure 4.13

Analyse

- Il est important de remarquer que ces tests ont été réalisés sur la même phrase que les tests précédents mais que cette phrase n'est absolument pas utilisée pour l'apprentissage bas-niveau (ni pour l'apprentissaqe haut-niveau).
- La figure 4.11 est exactement la même que la figure 4.8, elles sont réalisées dans les mêmes conditions.
- L'apprentissage haut-niveau améliore le suivi temps réel. Le chemin temporel ne comporte presque plus de pics intempestifs. Dans le dernier cas (4.13), les chemins temps réel et a posteriori sont parfaitement les mêmes ce qui est normal. En effet, à chaque itération, les probabilités sont affinées jusqu'à ce qu'il n'y ait plus qu'un seul chemin envisageable.

4.3 Tests avec erreurs dans la voix suivie :

Plusieurs tests d'erreur ont été réalisés. Pour ce faire, nous avons introduit volontairement des erreurs diverses et variées dans les voix à suivre et lancé la simulation.

4.3.1 Silences

Le bon suivi du silence est un point important en suivi de voix parlée. En effet, si le système se déclenchait à chaque petit bruit parasite cela serait très gênant. Nous avons réalisé des tests avec des silences allongés (10 secondes par exemple), des silences raccourcis et même des silences enlevés. Les résultats ont toujours été très positifs. Le système est extrêmement robuste pour la détection des silences.

4.3.2 Bégaiements

Un autre type d'erreur peut être le bégaiement du locuteur. Les tests ont été réalisés en dupliquant par exemple le début d'un mot. Le locuteur dit alors : "Jeje susuis rorobinson...". Le système avance dans le modèle pour finalement revenir au début du mot dans le modèle.

4.3.3 Erreurs

Des erreurs étrangères ont été finalement ajoutées. Des morceaux de signaux venant d'autres exemples de phrases, des morceaux de signaux venant de plus loin dans la phrase, des respirations répétées ajoutés un peu partout dans la phrase à suivre permettent de tester la capacité du système à sauter les erreurs.

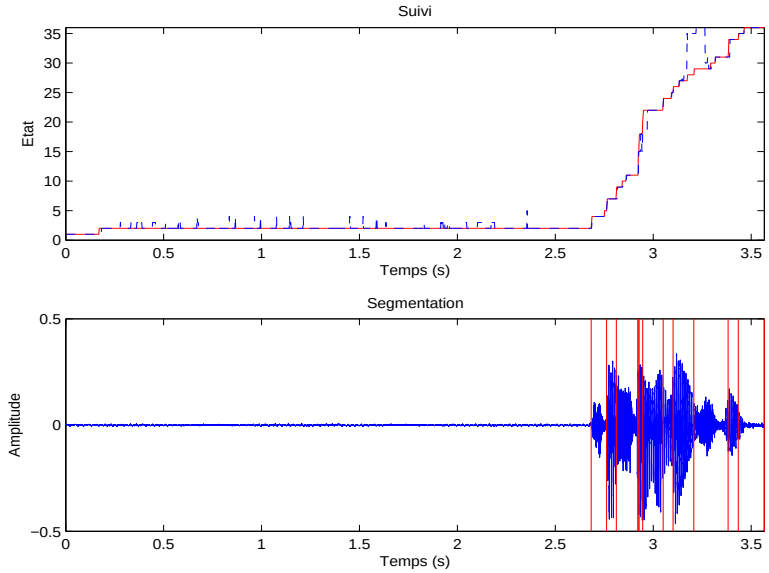


FIG. 4.14 – Suivi avec 10 coefficients et un long silence au début pour la phrase "Je suis Robinson". Malgré quelques soubressauts du suivi en temps réel, la segmentation est bonne. Le suivi est très robuste pour les silences

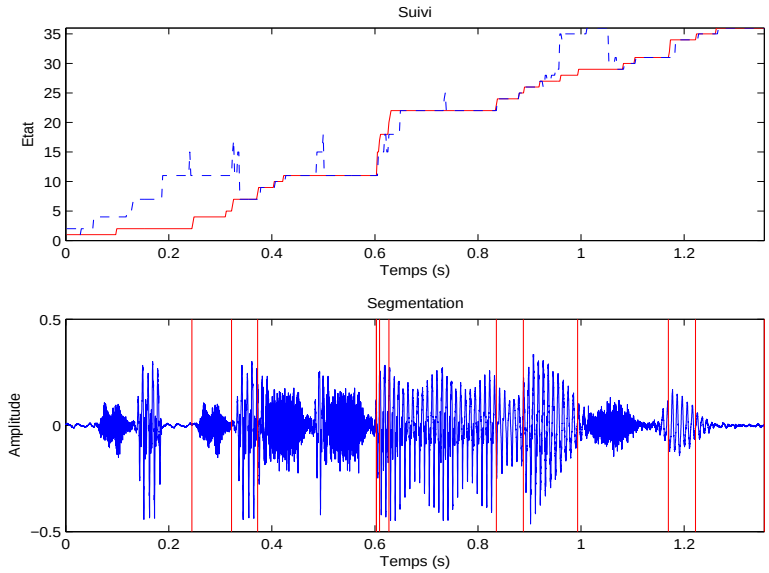


FIG. 4.15 – Suivi avec 10 coefficients et des bégaiements pour la phrase "Je-je su-suis Ro-robinson". La segmentation est bonne. Le suivi temps réel a des problèmes mais se recale finalement. Mais bon un acteur Beg...

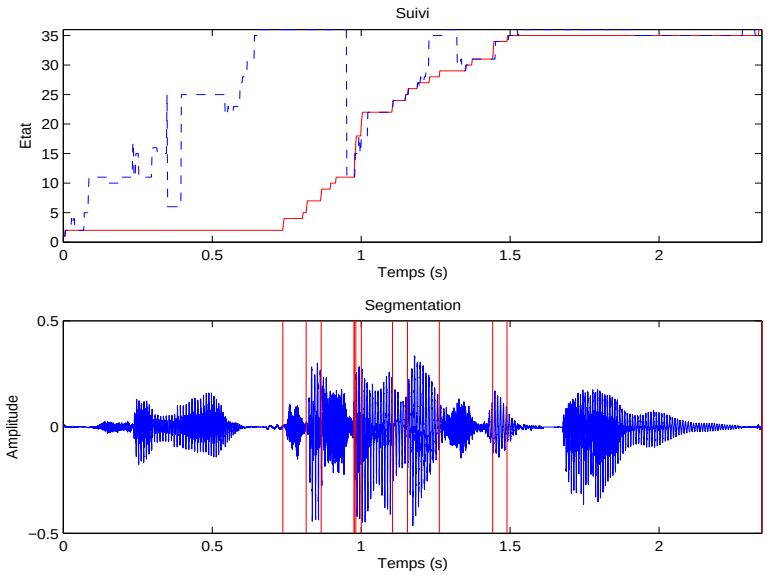


FIG. 4.16 – Suivi avec 10 coefficients et des erreurs ajoutées pour la phrase "Je suis Robinson". La segmentation du suivi a posteriori est bonne mais le suivi en temps réel avance alors qu'il devait attendre le signal du milieu.

Ces tests montrent bien que le système prend en compte les erreurs possibles mais sa façon de le faire n'est pas robuste. Pour le bégaiement, par exemple, le chemin ne recommence pas exactement au début du mot. Les erreurs, ajout de signal n'ayant aucun rapport, ne sont pas sautées. Dans la figure 4.16, le premier bloc de signal ne devrait pas faire avancer le suivi. Seul le silence de taille variable est très bien suivi. Pour pallier à ces problèmes, nous avons implanté le système des états fantômes de Diemo Schwarz et Nicolas Orio (cf [Orio et Déchelle. 2001][3]). Cependant, nous n'avons pas eu le temps de le tester dans le cadre de ce stage, le temps imparti étant trop court. C'est une des évolutions qui, nous le pensons, améliorerait le suivi.

Chapitre 5

Limites et Evolution du système

Les tests que nous avons réalisés ont donné de bons résultats dans le cadre de notre étude. Cependant, le système étudié reste dépendant de la pré-segmentation de la phrase à suivre. Cette pré-segmentation est longue et fastidieuse. C'est pourquoi il faudrait l'avoir faite une fois pour toute. Il serait bien aussi de généraliser le système à plusieurs locuteurs. Cette nouveauté nécessite une étude poussée des coefficients utilisés pour les rendre indépendants du locuteur.

5.1 Contexte des phonèmes

Nous avons vu dans les tests et les résultats des tests que le problème principal d'un bon suivi réside dans le fait d'avoir une bonne base de donnée d'entrainement du modèle HMM. Nos modèles ont été entrainés sur des exemples de phrases tirés de la pièce de théâtre "Retour Définitif et Durable de l'Etre Aimé" et les suivis ont porté sur les mêmes phrases que ces phrases d'entrainement. Le but d'un bon suivi est de pouvoir être adaptable à tout type de phrase sans entrainement préalable. Il est donc nécessaire de prendre en base de données d'entraînement beaucoup plus de phrases.

5.1.1 Repertorier tous les contextes des phonèmes ?

Chaque phonème peut être rencontré dans une phrase dans différents contextes en fonction du phonème suivant et du phonème précédent. Ces contextes sont les triphones dont nous avons parlé précédemment. Pour réussir à entrainer un modèle bas niveau de qualité, il faudrait donc réussir à répertorier tous les contextes des phonèmes. Dans la littérature, les linguistes en répertorient plus de 10000. Ce qui est énorme. Réussir à créer ce bas niveau permettrait à l'utilisateur du suivi de n'avoir qu'à entrainer le modèle haut niveau sur les transition pour adapter l'enchainement des modèles bas niveau à la phrase à suivre.

5.1.2 Utiliser tous les contextes pour l'entrainement ?

Réussir à créer ce bas niveau permettrait à l'utilisateur du suivi de n'avoir qu'à entraîner le modèle haut niveau sur les transitions pour adapter l'enchainement des modèles bas niveau à la phrase à suivre. Le nombre de contextes différents, ou de triphones différents est énorme et cela demanderait un travail titanesque de segmentation et de classement ainsi qu'un temps de calcul énorme. Cependant, cela serait fait une fois pour toute pour une voix donnée et n'aurait plus besoin d'être recommencé.

5.1.3 Evolutivité du modèle : auto-apprentissage

Il serait utile de mettre aussi en place un système d'auto-apprentissage sur les voix suivies. Il suffirait de mettre en mémoire les paramètres suivis pour ensuite les utiliser dans la base d'entraînement et refaire l'entraînement. Cette idée admet l'hypothèse que le suivi est très bon et que donc la segmentation peut être utilisée comme modèle pour suivre ultérieurement une autre phrase. Cependant, cette hypothèse est un peu paradoxale : pourquoi chercher à affiner le modèle s'il est déjà très bon ? En fait, ce système permettrait d'affiner les modèles bas niveau des phonèmes n'ayant pas beaucoup d'exemples d'entrainement. Il serait judicieux de réaliser l'auto-apprentissage de façon partielle. En effet, le temps d'entrainement de tout le modèle croissant avec le carré du nombre de données d'entrainement, il deviendrait trop long assez rapidement. Il faudrait donc implémenter un système qui tout en gardant son modèle pré-entrainé prendrait en compte les nouvelles données pour affiner le modèle. Cela pourrait se faire avec un système de poids en prenant comme données d'entraînement les centres des gaussiennes du modèle

avec un poid fort et les nouvelles données avec un poid faible. Ce système est encore à étudier.

5.2 Mono/multilocuteur

L'intérêt de disposer d'un système multi-locuteur est très grand, cela signifie que l'outil de suivi est unique et sert pour plusieurs voix différentes (que ce soit des voix d'hommes, de femmes ou d'enfants). L'apprentissage sur chaque voix ne serait plus nécessaire.

5.2.1 Théorie

Normalement, l'utilisation des coefficients MFCC pour la reconnaissance et le suivi de la parole permet de se détacher au mieux du timbre de la voix et donc de suivre plusieurs locuteurs différents. Les quelques tests que nous avons réalisés n'ont pas été positifs. Cependant nous n'avions pas fait d'études préalables de la validité du modèle, des coefficients, et de la base de données d'entraînement. Ces tests ont porté sur des enregistrements de nos voix. Puis nous avons simulé le suivi sur nos voix en utilisant les modèles bas niveau des acteurs initiaux et en entrainant les haut niveaux avec des exemples de nos voix. Apparemment les coefficients que nous utilisons ne sont pas directement très adaptés au suivi en multi-locuteur.

5.2.2 Application

Pour suivre la voix en multilocuteur, il faudrait d'abord réaliser la base de données en multilocuteur. les tests que nous avons effectués en plaçant dans la base de données d'entraînement une voix d'homme et une voix de femme n'ont pas donné de résultats très probants. Le suivi d'une troisième voix n'était vraiment pas bon. Il faudrait prendre en compte plus d'exemples de voix différentes et les intégrer dans la base de données. Ainsi, le modèle pourrait suivre un éventail plus large de voix.

5.3 Amélioration des coefficients de description

Les coefficients décrivant les phonèmes que nous avons utilisés ne sont bien sûr pas optimaux en règle générale mais le sont-il jamais ? Ils sont bons dans le cadre de l'utilisation que nous cherchons à faire de l'outil de suivi de voix parlée. Cependant ils peuvent être encore améliorés.

5.3.1 Seuillage des ZCR

Les paramètres de nombre de passages par zéro ZCR sont biaisés. En effet, il arrive fréquemment que le signal tourne rapidement autour de zéro pour finalement redescendre. C'est pourquoi, nous avons pensé introduire un seuillage sur le calcul de ces paramètres ZCR.

Dans notre idée, le signal ne devrait plus passer de positif à négatif pour que l'on dise qu'il y a un passage par zéro. Il faudrait qu'il passe d'un seuil négatif à un seuil positif. Ainsi on élimine toutes les petites variations de bruit ou de parasites qui biaisent le calcul du ZCR. Cela revient finalement à effectuer un filtrage passe-bas du signal pour les calculs des ZCR.

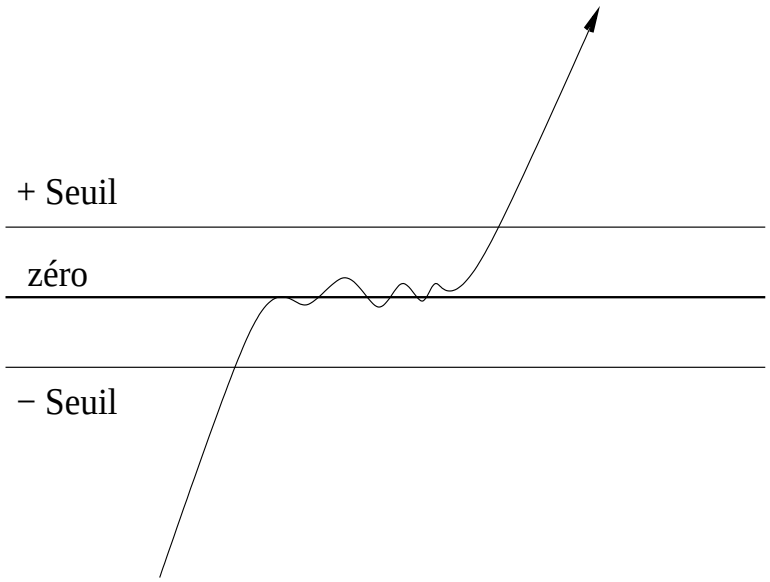


FIG. 5.1 – ZCR filtrés. Ici on ne prend en compte qu'un passage par zéro

Cette façon de faire devrait affiner le calcul des ZCR et donc aider à

réaliser une séparation meilleure des phonèmes. En particulier, nous avons vu que souvent, le "s" est classé comme un silence si le silence est proche. Changer la façon de calculer les ZCR aiderait à éviter ces erreurs. Cependant, pour calculer ces nouveaux ZCR, il est nécessaire de prendre en compte tout le signal passé pour chercher les maximums locaux passés. Cela ralentirait le calcul temps réel des coefficients.

5.3.2 Ajout d'un paramètre de silence

C'est pour cette raison que nous avons pensé à un autre paramètre. Il est difficile d'adapter le système quand on le change d'ambiance sonore rien qu'à cause des changements dans le niveau de bruit. Pour cela, il serait bien d'introduire un paramètre ayant une valeur minimum lorsque l'on est dans un silence et une valeur maximum lorsque l'on est dans du signal. Ce paramètre serait donc juste un seuil. Il pourrait être calculé juste sur la fin du signal pour ne pas limiter la réactivité du suivi.

L'introduction de ce paramètre permettrait donc aussi de résoudre le problème précédent du "s" classé comme un silence. Il suffirait de régler le seuil critique de signal pour être dans un silence en fonction du niveau sonore ambiant de l'environnement. Ce réglage pourrait même être automatique.

Il serait bon, en conclusion, de recentrer les objectifs du système. En effet, suivant le but du dispositif, il peut être bon de développer telle ou telle caractéristique de précision, de rapidité ou de portabilité. Le développement d'un aspect limitant les autres. Notre contrainte pour la pièce de théâtre "Retour Définitif et Durable de l'Etre Aimé" était une segmentation en temps réel de la voix parlée connaissant le texte dit. Dans ce contexte là, il faut affiner la précision de segmentation au risque de ne pas avoir de multilocutarité et de prise en main directe. Même si les phrases à suivre ont du être présegmentées, c'était le prix à payer pour avoir le suivi le plus précis possible. La continuité de cette étude serait le travail sur les thèmes cités dans ce chapitre : multilocutarité et portabilité. L'étude de ces sujets permettrait de réaliser un système de suivi le plus général possible avec possibilité de le spécialiser en multilocutarité, en précision pour la segmentation automatique.

Chapitre 6

Application en temps réel

6.1 Le spectacle Retour définitif et durable de l'être aimé

L'outil de suivi de voix parlée que nous avons développé pendant ce stage doit répondre aux souhaits exprimés par Olivier Cadiot et Gilles Grand, respectivement réalisateur et compositeur de ce spectacle. En effet, cette pièce utilise un grand nombre d'effets sonores.

Sur une scène dépouillée de tout décor, trois acteurs emmènent le spectateur dans un univers truffé de sons et d'histoires étranges. Les ingénieurs du son et les acteurs rassemblent leur virtuosité pour créer en temps réel un paysage sonore qui tient lieu de décor. Dans ce cadre, Gilles Grand souhaiterait utiliser un effet particulier sur un passage où chacun des trois acteurs récite un texte à son tour : le but serait que Valérie ait la voix de Laurent, Laurent celle de Philippe et Philippe celle de Valérie. Cet effet nécessiterait donc un suivi temps réel de la voix parlée et une resynthèse temps réel utilisant la voix du voisin. Notre outil serait donc utilisé dans cette optique (la resynthèse temps réel étant prise en charge par l'équipe Analyse-Synthèse). Grâce à un apprentissage préalable sur de multiples enregistrements des phrases dites par chacun des acteurs, l'outil est entraîné. Sur scène, lors du suivi en temps réel de la voix de l'acteur, l'index temporel du phonème qui est prononcé est récupéré et fourni à l'outil de re-synthèse temps réel. Le temps de détection du phonème par rapport au moment où celui-ci est prononcé, reste inférieur au seuil de détection de deux sons.

Dans les tests effectués pour le spectacle, les programmes d'apprentissage et de suivi avec "ghost states" n'ont pas été entièrement validés, faute

de temps. Mais ceux-ci devraient permettre ultérieurement de suivre la voix de l'acteur malgré des divergences éventuelles par rapport aux phrases d'entraînement qui respectent strictement le texte de l'oeuvre. On note donc que cette robustesse de l'outil permet une plus grande liberté d'expressivité pour l'acteur.

6.2 Vers un patch Jmax

Le but ultime des simulations sous Matlab était de mettre au point le système de suivi pour ensuite le transcrire en C/C^{++} pour en faire un patch de Jmax. Ce patch permettra de bénéficier d'une facilité accrue du système de suivi en situation de spectacle.

6.2.1 Le format SDIF Sound Description Interchange Format

Pour transmettre les résultats de nos programmes Matlab à la plateforme C, nous avons défini un format SDIF spécifique. Ce type de données standardisées a été développé par l'équipe Analyse-synthèse et notamment par notre tuteur de stage Diemo Schwarz. Le format qui nous intéresse dans notre cas est double.

- Un premier format SDIF concerne les données issues des apprentissages bas-niveau. Plusieurs "frames" comportant chacun plusieurs matrices sont définies pour transmettre le nom des phonèmes, les informations sur le nombre de paramètres des modèles associés, les matrices de transition, les données des PDF (Probability Density Function) - moyennes, covariances, inverses -.
- Un deuxième format SDIF concerne les données issues des apprentissages haut-niveau, à savoir la matrice de transition globale et les phonèmes de la phrase à suivre.

6.2.2 Le patch Jmax

Le patch réalisé par Diemo Schwarz a l'interface représentée figure **??**. A l'instar du suivi de partition, une représentation graphique très démonstrative permet de suivre en temps réel le suivi des phonèmes du texte. Chaque petit rectangle surmonté de son phonème associé s'allume

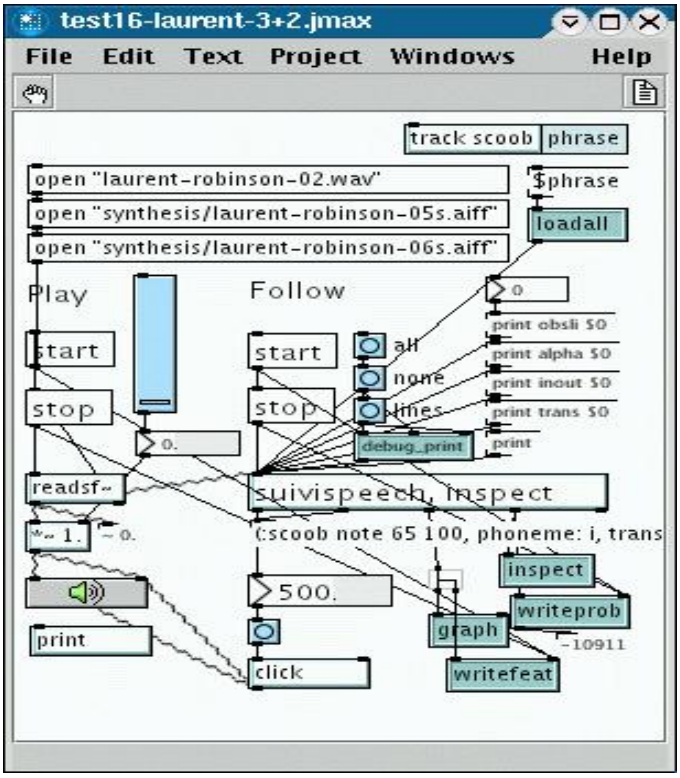


FIG. 6.1 – Interface du patch Jmax

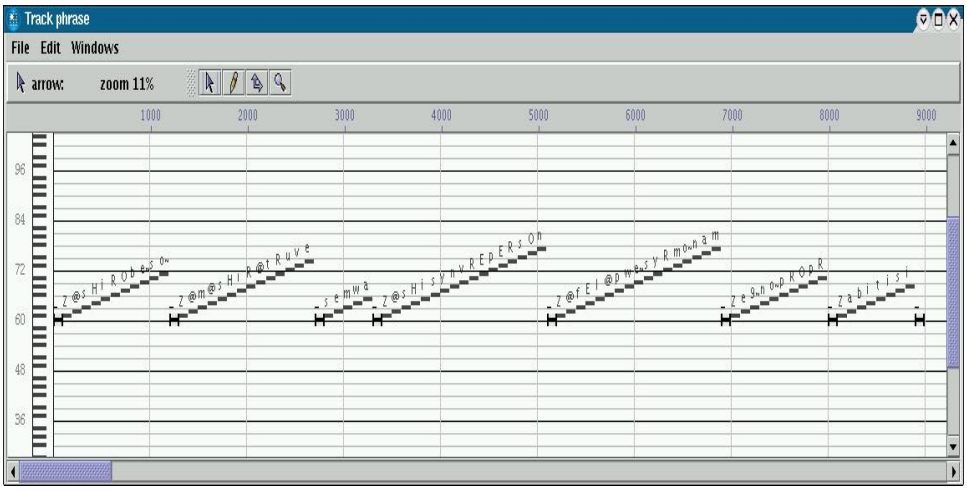


FIG. 6.2 – Suivi

et émet un clic lorsque son phonème est détecté.Voici une vue de cette représentation :

Remarque : Pour travailler à partir des mêmes données, il était important d'avoir un programme commum pour calculer ces données. Un mex-file a donc été écrit pour faire appel à la fonction C qui calcule les MFCC et les zero-crossing rates.

6.3 Idées d'applications

L'outil que nous avons créé pendant ce stage est spécialement adapté à des applications qui nécessitent un suivi à l'échelle temporelle du phonème, et non à celle du mot (pour laquelle des outils spécifiques existent déjà). Par ailleurs, il faut bien garder à l'esprit que l'utilisation du suivi nécessitera une phase d'entraînement préalable, et donc une connaissance du texte qui devra être suivi. Par ailleurs, à l'heure actuelle, seule une mise en œuvre mono-locuteur est admise ; cependant dans une version ultérieure de l'outil, une mise en œuvre multi-locuteur pourra être prévue. En tenant compte de ces contraintes, voici quelques idées qui nous sont venues.

6.3.1 Apprentissage des langues

Guide de prononciation

Il s'agirait là de développer un laboratoire de langues "intelligent" orienté sur le travail de l'accent et de la prononciation. Le principe serait le suivant : l'utilisateur est chargé de prononcer des phrases proposées par le logiciel. Ces phrases sont suivies par l'outil. La liste des phonèmes attendus est parcourue peu à peu jusqu'à ce qu'un phonème manque : cela signifie qu'il y a eu défaut de prononciation. l'outil indique alors à quel phonème précis l'erreur est faite. La démarche que nous avons suivie pour développer l'outil permet d'intégrer plusieurs types de prononciations correctes (pour différents accents d'une langue par exemple). En effet, pour chaque phonème, plusieurs prononciations sont admise grâce à l'utilisation d'états contenant plusieurs classes, représentées par des multi-gaussiennes. Cependant, cette application nécessiterait une version multi-locuteur de notre outil.

Apprendre à lire une langue

Cette application concerne les personnes souhaitant apprendre à lire une langue qu'ils parlent déjà mais qu'ils ne lisent pas (enfants ou adultes illettrés). Le principe est le suivant : le professeur lit la phrase inscrite sur l'écran de l'ordinateur. Simultanément, le phonème prononcé (ou la syllabe) est mis en gras dans la phrase. L'élève peut ainsi associer les syllabes entendues avec les syllabes écrites. Cette mise en œuvre de loutil nécessiterait une étape dadaptation à la voix du professeur afin de pouvoir effectuer la phase dapprentissage nécessaire au suivi.

6.3.2 Karaoké adaptatif

Ce karaoké du futur permettrait de contrôler le tempo et l'harmonisation de la chanson en temps réel (dans une limite musicale acceptable). En fonction de la vitesse de diction du chanteur, la chanson serait subrepticement ralentie ou accélérée afin de sadapter aux petites erreurs de "flow" du chanteur. Par exemple, si le chanteur attendait avant de continuer à chanter les paroles, la musique attendrait aussi et ne changerait d'harmonie que sur une parole donnée. Ainsi, le chanteur serait plus libre de s'exprimer, le système le suivrait. L'action de réguler le tempo pourrait servir au début de la chanson, lorsque le chanteur s'adapte à la pulsation rythmique. Le tempo serait alors plus ou moins commandé par le débit des phonèmes prononcés par le chanteur. Cette application nécessiterait une version multi-locuteur de notre outil.

6.3.3 Commande d'effets audiovisuels pour le spectacle vivant

C'est dans le cadre du spectacle vivant que la nécessité de l'outil de suivi de voix parlée a été ressentie. La fonction requise était de récupérer chaque phonème en temps réel afin de le resynthétiser immédiatement avec une voix différente. Cependant, l'obtention de cette suite de phonèmes pourrait constituer une commande "atemporelle" d'évènements sonores et visuels, entièrement corrélée avec le rythme de diction de l'acteur en scène. Lavantage de ce type de commande est quelle permet à lacteur une grande souplesse dexpression tout en lui offrant une haute précision pour les déclenchements deffets audiovisuels. Pour ce contexte, nous avons recensé plusieurs utilisations.

Contrôle de l'éclairage ou deffets sonores

Ce mode de commande est intéressant si le metteur en scène souhaite ajouter des effets de courte durée. Dans la pièce "Retour Définitif et Durable de l'Etre Aimé", certains effets sonores interviennent seulement sur la durée d'une phrase ou même sur un mot en particulier, et l'on voit l'ingénieur du son jongler avec ses potentiomètres pour lancer l'effet au moment exact. Avec l'outil, il lui suffirait de déclencher le suivi quelques instants avant et l'effet serait lancé au phonème près. Il pourrait en être de même pour des effets de lumières, ou tout autre type de commande nécessitant une synchronisation très précise avec l'acteur.

Accompagnement sonore dun texte

En poussant encore plus loin ce type de commande par suivi de voix parlée, une nouvelle forme de diction pourrait être imaginée : la "diction harmonisée". A chaque phonème par exemple, on associe un échantillon sonore ou bien une hauteur de note jouée par un instrument donné. Lorsque le texte est récité, les sons associés aux phonèmes se synchronisent parfaitement avec le rythme de diction. Cette application pourrait constituer une nouvelle forme de contrainte d'écriture textuelle et musicale. On pourrait jouer sur l'utilisation des rimes qui commanderaient alors la répétition du même module sonore.

Aujourd'hui, ces différentes application ne sont pas encore réalisables. Il faudrait d'abord valider le multilocuteur et réussir à ne plus dépendre de l'entrainement haut niveau. La puissance de cet outil réside dans le fait quil lie la souplesse d'expressivité avec la précision temporelle de déclenchement dévénements à l'échelle du phonème.

Chapitre 7

Conclusion

Au cours de notre stage, nous avons élaboré un système de suivi de parole en temps réel. Il s'agissait de développer un outil principalement destiné au spectacle même si les applications envisageables sont multiples. Ce stage a été l'occasion pour nous de mettre en pratique et de d'enrichir des connaissances touchant à des disciplines diverses comme le traitement du signal, la reconnaissance de la parole, la phonétique... Les étapes de notre travail, présentées dans un ordre qui n'est pas "chronologique" en raison des retours en arrière, des remises en question nécessaires à chaque nouvelle étape, ont été les suivantes :

- définir un modèle de Markov approprié (un bas- et un haut-niveau),
- bien contrôler la nature et le nombre de données pour caractériser complètement une voix (MFCC, delta-MFCC, taux de passages par zéro,)
- travailler les paramètres du modèle (nombre d'états, caractéristiques des mélanges de gaussiennes...),
- disposer d'une base de données suffisante(exemples pré-segmentés),
- réaliser les scripts d'apprentissage et de suivi, le suivi étant a posteriori dans un premier temps,
- modifier l'algorithme forward dans un deuxième temps pour la contrainte temps réel,
- développer un système de gestions d'erreurs éventuelles vis à vis du texte appris (états fantômes).

Si l'utilisation des modèles de Markov est classique en reconnaissance de parole, quelques apports personnels ont permis d'améliorer les résultats. Le taux de passages par zéro en fin de fenêtre d'analyse, par exemple a

permis d'améliorer la réactivité du suivi. L'idée d'entraîner les états fantômes avec toutes les classes autres que celle du phonème associé à l'état fantôme considéré est originale, inspirée de l'apprentissage des états fantômes pour le suivi de partition (cf [Orio et Schwarz. 2001][4]).

Les nouvelles directions à emprunter pour achever l'outil de suivi de parole sont multiples, nous avons donné quelques pistes dans le chapitre V. Nous les rappelons ici :

- disposer d'une base de données de phonèmes prenant en compte leur contexte (notion d'allophones),
- entraîner les états fantômes sur des séries d'exemples comportant des erreurs répertoriées (bégaiement, saut de phonème, saut de mot, substitution de mot...)

Les perspectives d'applications du suivi de parole au service du spectacle sont très importantes. Le suivi permet de communiquer des tops à une machine qui peut déclencher toutes sortes d'évènements calés sur le déroulement en temps réel de la pièce. Ces évènements peuvent être une séquence sonore, une image, un effet lumineux, une effet pyrotechnique, une substitution de voix d'un acteur comme pour la pièce d'Olivier Cadiot... Le suivi de parole est donc un formidable outil qui peut élargir considérablement le champ des possibles dans les applications des nouvelles technologies au spectacle vivant.

Une évolution intéressante de ce système est le suivi de la position et du geste artistique. Dans ce cas, les coefficients d'analyse seraient réadaptés mais l'architecture globale du système reste utilisable, les HMM seraient toujours intéressants à utiliser. Il faudrait alors chercher quels seraient les paramètres les plus adaptés à la description de l'expression corporelle.

Chapitre 8

Annexes

8.1 Exemples de suivi

8.1.1 Comparaison entre les coefficients normalisés et les coeffients non normalisés

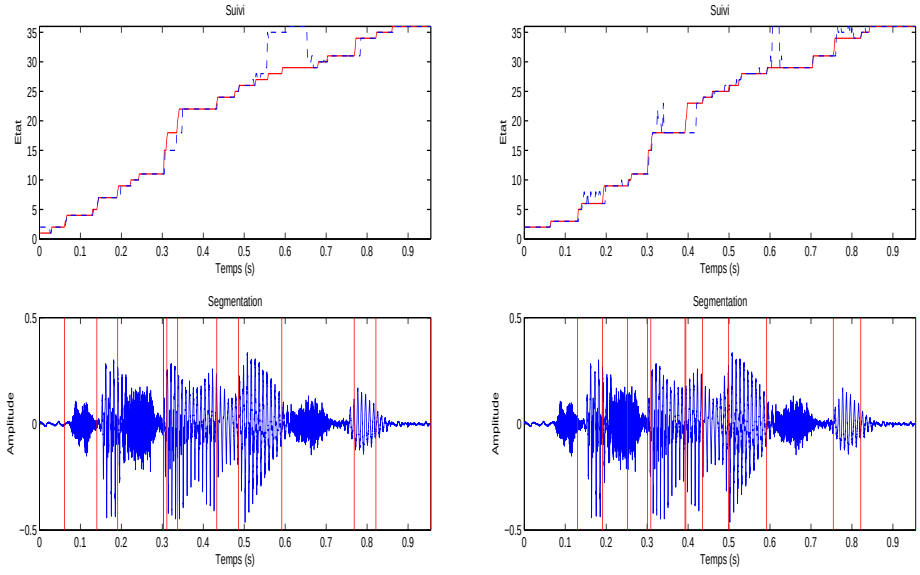


FIG. 8.1 – Suivi avec 2 ZCR et 8 MFCC normalisés (à droite) et non normalisés (à gauche)

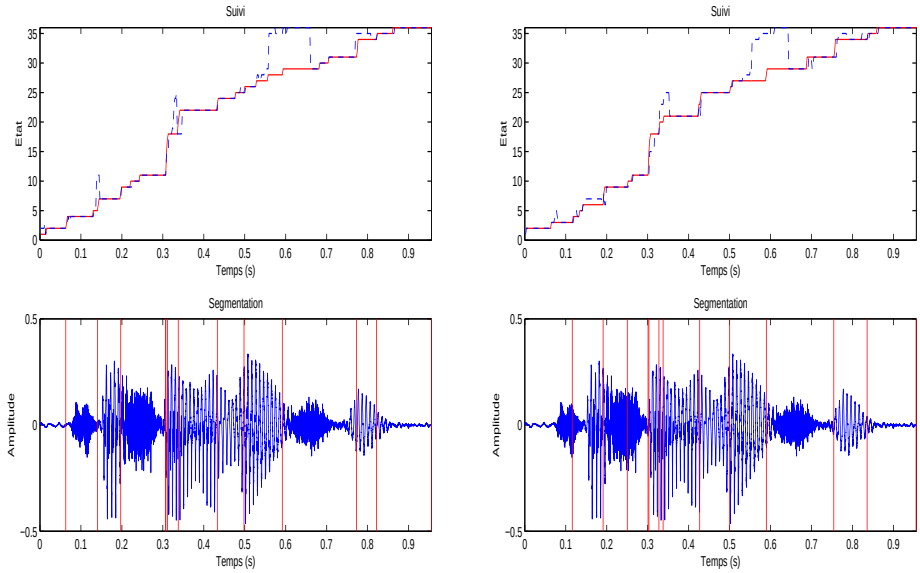


FIG. 8.2 – Suivi avec 2 ZCR et 10 MFCC normalisés (à droite) et non normalisés (à gauche)

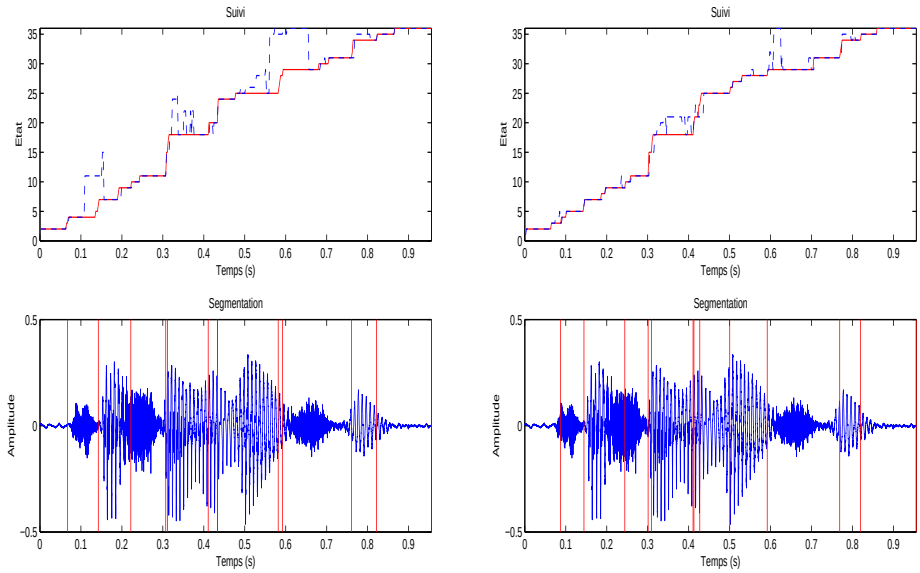


FIG. 8.3 – Suivi avec 2 ZCR et 12 MFCC normalisés (à droite) et non normalisés (à gauche)

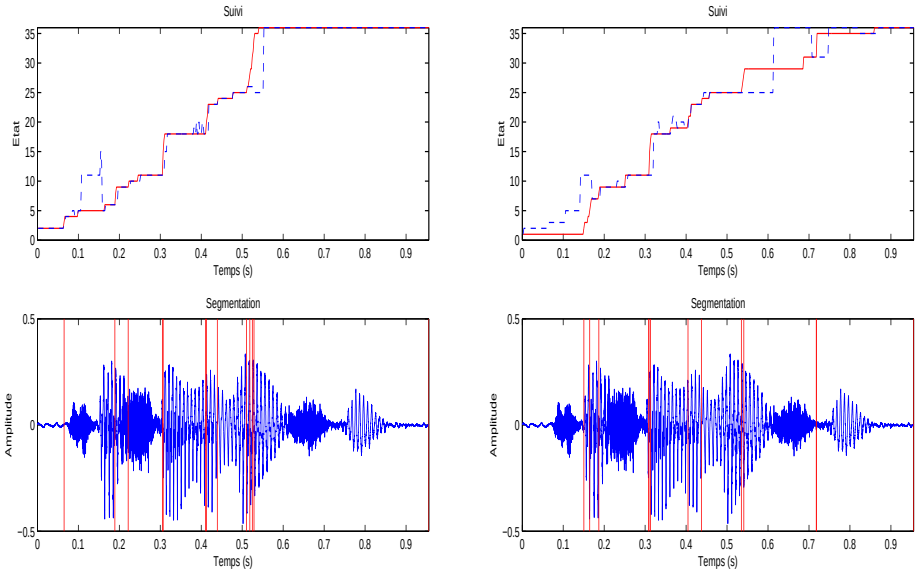


FIG. 8.4 – Suivi avec 2 ZCR, 12 MFCC et 12 Delta-MFCC normalisés (à droite) et non normalisés (à gauche)

Ces différents tests montrent bien que non seulement le suivi est meilleur avec des données normalisées (uni-variées et centrées) mais aussi que la segmentation est améliorée dans ce cas.

8.1.2 Suivi d'une phrase longue

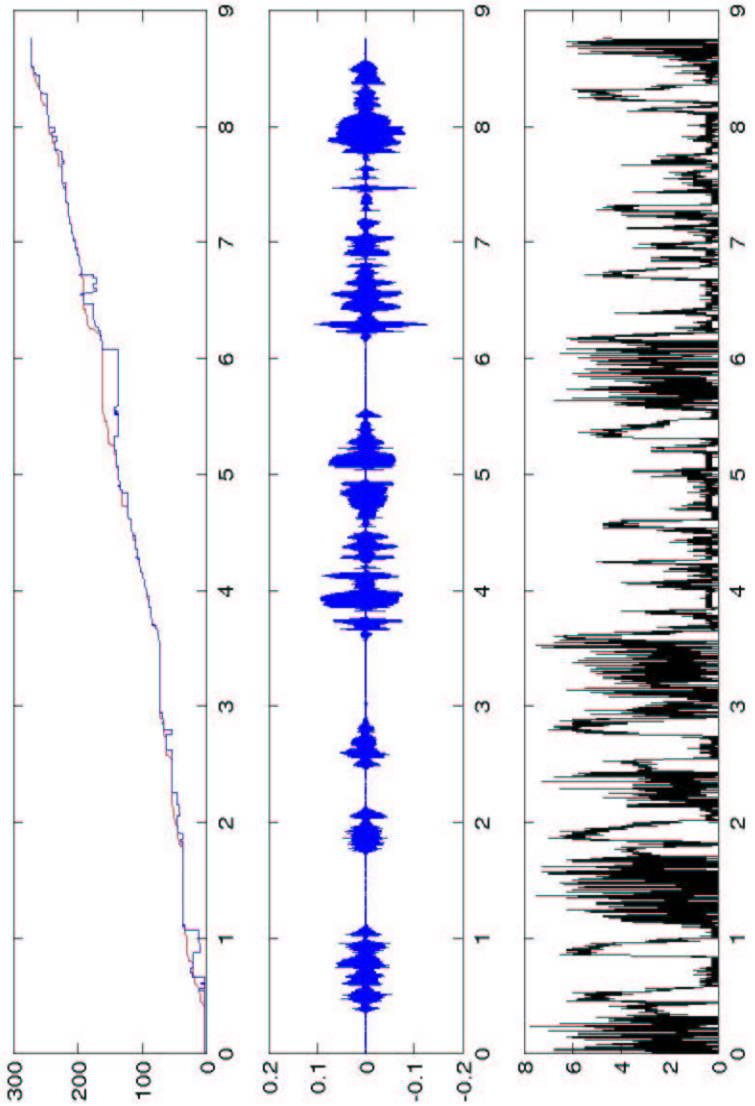


FIG. 8.5 – "... Je suis Robinson, j'y suis j'y reste, c'est le temps exact de vie qui dure le temps où je suis, c'est la durée de ce que je dis qui découpe le moment où je suis là..."

8.2 L'algorithme des K-Moyennes (K-Means)

L'algorithme des k-moyennes permet de séparer en classes un ensemble de données. Cette séparation en classe est basée sur la proximité en distance Euclidienne. Chaque classe sera définie par son centre C_j dans l"ensemble décrit et sa variance. On définit d'abord le nombre de classes que l'on veut et on entre notre jeu de données dans l'algorithme. Ce jeu de données est en général un ensemble de vecteurs de même type. La variance d'une classe est relative à la distance moyenne qu'à un vecteur de cette classe avec le centre de cette classe.

Voici le détail de l'algorithme avec K classes

1. Initialisation : On choisit arbitrairement ou au hasard les vecteurs centres des classes.
$$C_1 \dots C_K = I_1 \dots I_K$$
2. Recherche des voisins les plus proches : Pour chaque vecteur à classer, on recherche son centre de classe le plus proche en distance Euclidienne. On lui alloue alors cette classe.
$$S_k = X_n | d(X_n, C_k) < d(X_n, C_j), \forall j \neq k$$
3. Remise à jour des centres : après avoir associé àtous les vecteurs un centre de classe, on remet à jour les centres des classes en moyennant tous les vecteurs d'une même classe. On obtient un nouveau jeu de centres de classe.
$$C_k = moyenne X_j | X_j \in S_k$$
4. Itération : On itère le processus jusqu'à ce que la remise à jour n'améliore plus les vecteurs centres de classes.

8.3 Texte de la pièce "Retour Définitif et Durable de l'Etre Aimé" de Olivier Cadiot

Ircam 31 mars 2003

No 1 : Valérie Dashwood

No 2 : Philippe Duquesne

No 3 : Laurent Poitrenaux

* suivi de la voix, c'est le numéro de l'acteur qui parle.

(*) analyse de cette voix, c'est le numéro de la voix de l'acteur qui est diffusée

A/ Robinson

2 (3) Je suis Robinson, c'est moi, je suis accoudé à la fenêtre, c'est maintenant, la neige est bleue à cause du néon bleu, mes deux coudes sont posés sur le froid du fer forgé du balcon, il y a trois griffons entrelacés à un souvenir de plante en métal,

3 (1) je suis Robinson, je me suis retrouvé, c'est moi, je suis une vraie personne, je fais le point sur mon âme, j'ai un nom propre, j'habite ici, c'est chez moi, je suis dans une ville, je suis heureux, je suis central, le balcon est à exactement 0°, il neige, la neige est bleue, c'est ici,

1 (2) je suis Robinson, j'y suis j'y reste, c'est le temps exact de vie qui dure le temps où je suis, c'est la durée de ce que je dis qui découpe le moment où je suis là, c'est moi, l'intérieur de moi correspond exactement à l'extérieur, je suis là où je suis, c'est celui qui dit qui l'est, j'existe.

B/ Page 2

2 Regardez-moi dans les yeux quand je vous parle, vous êtes fuyant, vous avez les yeux morts,

2 Ne dites rien, ma nièce m'a tout raconté sur vous,

2 (1) votre sexualité misérable, votre petite libido à trois balles, vos nostalgies médiocres, vous êtes un médiocre, un fruit sec, sans avenir, zéro,

vous étes très très déséquilibré, absolument, ne me regardez pas comme ça avec ces yeux de chien, pourquoi restez-vous comme ça sans bouger ? je vois bien que vous n'êtes pas mort, ne me faites jamais ça à moi, attention,

2 il y a un truc qu'il ne faut jamais me faire à moi, hein, ne me faites jamais le coup de "Je suis mort".

2 Vous n'avez pas l'air bien, il faut parler, arrêtez de gigoter comme ça, c'est fou d'être nerveux à ce point, il faut se soigner,

2 (3) Vous savez les gens diront toujours que vous faites semblant, d'un type qui fait la manche, on dit toujours qu'il fait semblant, le hic, c'est qu'il fait peut-être semblant mais, pendant ce temps, il le fait quand même, vous me suivez,

2 Vous n'avez rien écouté, hein ? si, alors qu'est-ce que je viens de dire ? ça parlait de quoi ? prouvez-le, quoi ? je n'entends rien, quoi ?

2 Ne bougez plus, voi-là.

C/ Photon

2 En tout cas on peut créer un jumeau à distance, je l'ai lu, ça marche, ça vous intéresse ? ne bougez pas, je vais vous lire l'article, c'est assez délicat à comprendre, vous m'arrêtez si vous avez des questions,

2 (3 ou 1) imaginez l'envoi d'un message secret dont le porteur serait un grain de lumière, un photon nommé C, Chris. Attention. L'idée de l'expérience est que A, Alice, puisse lire le message porté par C, Chris, sans avoir à le recevoir directement.

(3) C'est là où ça se corse. On lui donne un partenaire B, Bob, tous les deux ont la propriété d'être corrélés, en termes de mécanique quantique, ça veut dire que, quelle que soit la distance qui les sépare, le fait de mesurer les caractéristiques de l'un

(1) permet aussitôt de déterminer celles de l'autre. Vous suivez ?

(3) Bob reçoit le message de C, Chris, sans le comprendre, il est perturbé par cette opération et du même coup A, Alice, avec qui on a vu qu'il était étrangement lié, est aussi touchée. Attention. Tellement touchée qu'elle peut reconstituer un jumeau de C, Chris, à partir des informations fournies par B, Bob. Ça marche.

D/ Grosoiseaux

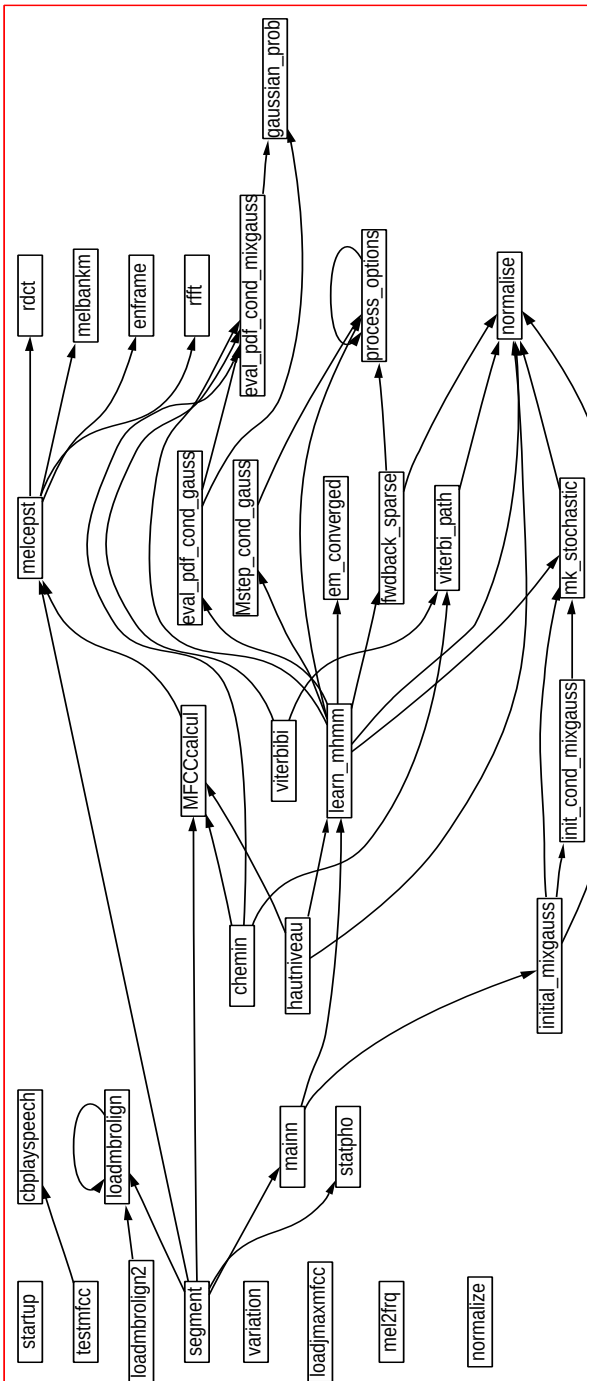
Les neuf gros oiseaux bruns zonent dans les champs de ping-pong. J'ai zappé la gâche alors je cherche des pâtes à la gnôle et neuf agneaux au camping de la montagne. C'est la teuf du boeuf, Bob. Absolument répète la japonaise, fascinée, vrrr. Pareil pour la photo, c'est pareil, ffft-fft.

Je glisse, je bouge sans cesse, ils croient que je danse, j'esquive, jeu de jambes infime, ffft-fft, déplacement, droite, hop, glisser, hop, c'est mon sport, en avant, je m'appuie sur les basses profondes au fond de la musique. Je glisse.

On dirait du Schumann derrière ? des embryons de piano, des débuts de romance sans paroles remixée avec des bruits de tronçonneuse ou le son sourd d'une clouteuse qui traverse une planche de chêne toutes les deux secondes, la musique avance, petits modules tendres en travaux, cinq bluettes sur dynamite obligée, une mélodie sort, ça se danse, sonate pour Travaux publics, je glisse, je suis dans le rythme, j'esquive chaque phrase en me couvrant, corps ramassé, respiration rapide, ouf-ouf, esquive, hop. Je glisse.

Ça c'est la belle-sœur par alliance de la femme de Rauschenberg. J'invente mes blagues, il n'y a plus que ça qui m'amuse, je me suis toujours demandé qui inventait les blagues, il faut bien quelqu'un au départ hein. Je glisse.

8.4 Arborescence des fichiers Matlab



Bibliographie

[1] [Roweis]. Hidden Markov Models. Sam Roweis.

[2] [Orio et Schwarz. 2001]. Alignment of Monophonic and Polyphonic Music to a Score. 2001

[3] [Orio et Déchelle. 2001]. Score Following Using Spectral Analysis and Hidden Markov Models. Nicolas Orio et Diemo Schwarz. 2001.

[4] [Orio et Schwarz. 2001]. Alignement of Monophonic and Polyphonic Music to a Score. N.Orio, N. and D.Schwarz. Proceedings of ICMC, 2001.

[5] [Rabiner et Juang. 1993]. Fundamentals of Speech Recognition. L.R.Rabiner et B.Juang. Englewood Cliffs, NJ : Prentice-Hall. 1993.

[6] [Rabiner et Al. 1989]. A Tutorial on Hidden Markov Models and Slected Applications in Speech Recognition. Lawrence R.Rabiner. Proceedings of the IEEE , vol.77, No.2, February 1989.

[7] [Cano, Loscos et Bonada. 1999]. Score-Performance Matching using HMMs. Pedro Cano, Alex Loscos et Jordi Bonada. Proceedings of the ICMC, 1999.

[8] [Loscos, Cano et Bonada. 1999]. Low-Delay Singing Voice Alignment to Text. Alex Loscos, Pedro Cano et Jordi Bonada. Proceedings of the ICMC, 1999.

[9] [Hasegawa-Johnson 2000]. Lecture Notes in Speech Production, Speech Coding, and Speech Recognition. Mark Hagesawa-Johnson. University of Illinois at Urbana-Champaign, February 2000.

[10] [Boite, Bourlard, Dutoit, Hancq et Leich. 1999]. Traitement de la parole. René Boite, Hervé Bourlard, Thierry Dutoit, Joël Hancq et Henri Leich. Presses polytechniques et Universitaires Romandes, collection électricité, décembre 1999.