REALISATION D'UN SYSTEME (PARADIS) POUR LA SYNTHESE DE LA PAROLE ARABE A PARTIR DU TEXTE

Noureddine Chenfour
Université Mohammed Ben Abdellah,
Faculté des Sciences Dhar El-Mehrez Fès,
Département de Mathématique et Informatique.

Résumé

L'étude que nous présentons dans cet article s'inscrit dans le cadre de la réalisation d'un système de synthèse de la parole à partir du texte pour la langue arabe.

Nous examinerons l'architecture générale de notre système PARADIS basé sur la concaténation des di-syllabes avec TD-PSOLA comme technique de synthèse, et la transcription graphème - phonème suivant le principe des règles compilées.

Après avoir examiné les problèmes de transcription phonétique, nous présentons l'intérêt du choix de la di-syllabe comme unité de concaténation pour le synthétiseur et son apport au niveau de la qualité de synthèse. En effet, la di-syllabe permet de réduire les problèmes de discontinuité temporelle lors de la concaténation. L'une des composantes fondamentales de notre système de synthèse de la parole arabe PARADIS est le synthétiseur TD-PSOLA. Il est connu par sa capacité d'action directe sur le signal de parole et par le concept de séparation entre l'algorithme de codage et la technique de synthèse. Il garantit ainsi une grande flexibilité au niveau suprasegmental et offre de très bonnes possibilités de variations prosodiques, ce qui apporte plus de naturel à la parole synthétique.

Mots Clés

Synthèse de la parole arabe - di-syllabe - TD-PSOLA - signaux à court-terme - marques de pitch - transcription graphème-phonème.

الملخص

إن الدراسة التي نتقدم بها في هذه المقالة تدخل في إطار إنجاز نظام تحويل المكتوب إلى كلام منطوق.

سنقوم إذن بفحص المكونـــات الأساسية لنظامـــنا PARADIS الذي يعتمد على وصل "الديسلابـــات" (di-syllabes) من جهة، وعلى تقنية TD-PSOLA من جهة أخرى. أما فيما يخص التحويل من الحروف إلى الكتابة الصوتية فهي تعتمد على مبدأ القواعد «المجمعـــة» (règles compilées).

بعد فحص المشـــاكل المتعلقة بالكتابة الصوتية، نقوم بشرح الأسبـــاب الرئيسية لاختيارنا لـــ« الديسِلاب» كوحدة صوتية داخل النظام ومدى تحسينه لنوعية الكلام المنطوق الناتج عن عملية الإلصاق.

يعتمد نظامنا أساسا على تقنية TD-PSOLA التي تمكن من التفريق بين عملية التشفير وتقنية التأليف والعمل المباشر على الإشارة الصوتية وتقريب الكلام المولد من الطبيعي.

الكلمات المفاتيح

علاج الكلام المنطوق – الديسلاب – تقنية TD-PSOLA – إشارات قصيرة المدى – علامات النغمة – الكتابة الصوتية.

Abstract

The study that we present in this paper concerns the realization of a text speech synthesis system for the Arabic language.

We will examine the general architecture of our PARADIS system that is based on the concatenation of di-syllables and TD-PSOLA as a synthesis method, the grapheme to phoneme transcription being based on the principle of compiled rules.

After having examined phonetic transcription problems, we present the interest of the choice of the di-syllable as a concatenation unit for the synthesizer and its contribution to improve the voice quality produced by the synthesizer. Indeed, di-syllables reduce temporal discontinuity problems during the concatenation. One of the fundamental components of our TTS system is TD-PSOLA synthesizer. It is known by its capacity of direct action on the speech signal and the concept of separation between the coding algorithm and the synthesis technique. TD-PSOLA has significantly improved the synthetic speech quality as it allows, with a great simplicity, the variation of the fundamental frequency of the synthesized speech signal.

Keywords

Synthesis of the Arabic language - di-syllable - TD-PSOLA - short-term signals-pitch marks - grapheme to phoneme transcription.

1. Introduction

Un système de synthèse de la parole par concaténation d'unités acoustiques est un système à deux composantes, l'une statique et l'autre dynamique. La composante statique consiste en un dictionnaire d'unités acoustiques de concaténation. La composante dynamique est le système de traduction du texte d'entrée en un message vocal équivalent. La synthèse passe par un ensemble d'étapes dont l'exploitation des unités du dictionnaire et leur adaptation prosodique constituent les contraintes de base pour former des messages vocaux de haute qualité.

La technique de synthèse TD-PSOLA (Time Domain Pitch Synchronous OverLap and Add) qui devient un standard pour les synthétiseurs TTS (Text To Speech) offre des possibilités intéressantes de concaténation et de variations prosodiques. La caractéristique la plus remarquable de l'algorithme TD-PSOLA est qu'il opère directement sur le signal temporel.

Cependant, un système de synthèse intégrant TD-PSOLA comme synthétiseur, doit disposer d'un dictionnaire d'unités acoustiques préparé d'une façon rigoureuse. Les unités doivent alors être choisies de manière à couvrir tout texte. En outre, elles doivent être sélectionnées et étiquetées de manière à favoriser l'opération de concaténation tout en introduisant les modifications prosodiques désirées. Pour l'arabe, la di-syllabe permet d'améliorer amplement la qualité de synthèse et de réduire les problèmes de discontinuité temporelle lors de la concaténation.

Notre système PARADIS (Psola ARAbic DI-syllable concatenation based System) constitue le résultat final d'un ensemble de travaux qui ont débuté il y a plus de 20 années (Mouradi, 1985). Le système PARADIS accepte en entrée un texte arabe voyellé. Celui-ci est d'abord traité par un module de transcription graphèmes-phonèmes pour générer le texte phonétique correspondant. Le module de transcription est engendré automatiquement par compilation d'un fichier de règles à l'aide d'un Langage de SPEcification formelle des Règles de Transcription LSPERT que nous avons réalisé pendant une toute première étape de notre recherche en 1994. Le texte phonétique est ensuite découpé en di-syllabes puis traité par un module prosodique permettant d'y insérer des marqueurs de variation de pitch et de durée. Ce dernier module vient d'être intégré dans notre système (Chenfour, 2001 ; Benabbou & al., 2002). A partir du texte découpé en di-syllabes, un signal acoustique est généré par le module de synthèse TD-PSOLA que nous avons développé en 1997 (Chenfour, 1997). L'extraction des segments acoustiques correspondants est réalisée par un module d'accès direct aux unités du dictionnaire acoustique. Un dictionnaire de di-syllabes préliminaire a été élaboré en 1997 (Chenfour, 1997). Nous avons repris la génération de ce dictionnaire avec un locuteur expérimenté et des méthodes automatiques plus performantes (Chenfour & al., 2000).

2. Architecture du Système PARADIS

2.1. Schéma général

Le système PARADIS se présente avec une architecture à 6 phases (Figure 1). La première phase de transcription graphèmes-phonèmes consiste à lire un texte arabe voyellé et le transcrire en un texte phonétique correspondant. Le module de trans-

cription a été généré automatiquement par notre compilateur de règles LSPERT (Langage de SPEcification des Règles de Transcription) à partir d'une spécification formelle des règles de transcription. Le texte phonétique est ensuite traité par un module prosodique permettant d'y insérer des marqueurs de variation de pitch et de durée. Une troisième phase consiste à découper le texte phonétique en di-syllabes. Un module d'accès direct basé sur une technique de Hash-code constitue la quatrième phase permettant l'extraction rapide des unités acoustiques à partir du dictionnaire des di-syllabes. Après décodage des données extraites par un module EPEC (Chenfour, 2000), le synthétiseur TD-PSOLA permet enfin de générer le signal vocal correspondant au texte d'entrée.

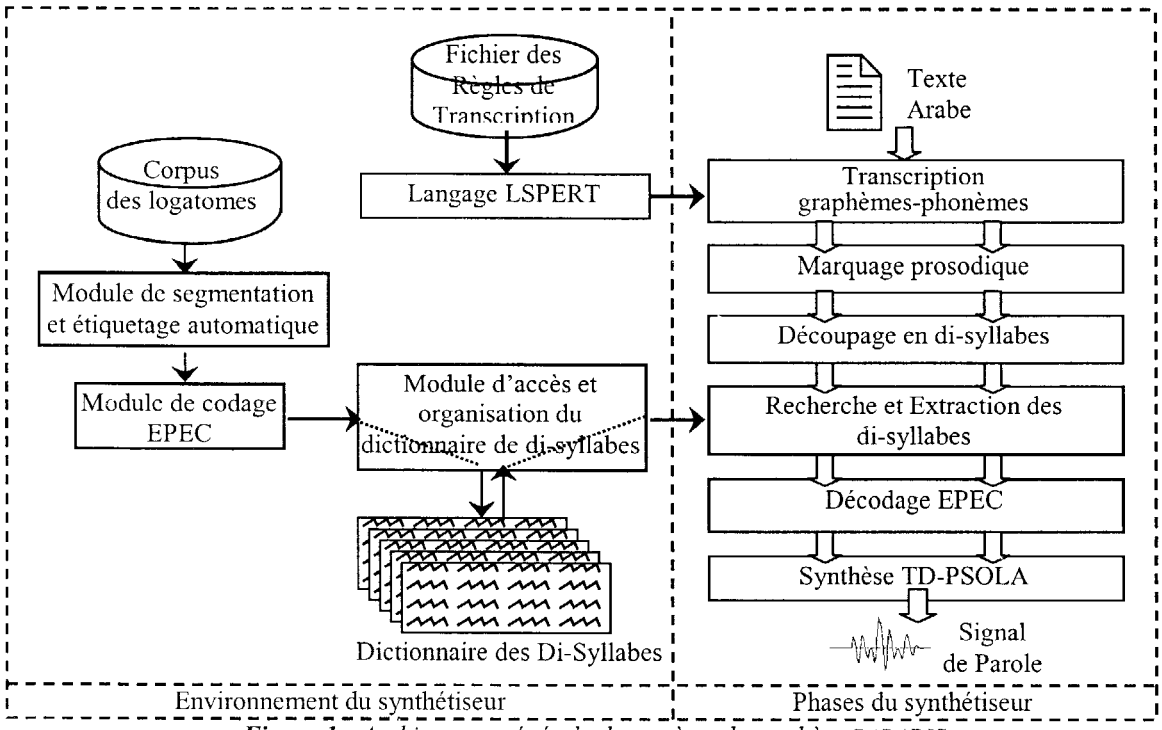


***Figure 1** : Architecture générale du système de synthèse PARADIS*

2.2. Phases du système

Notre système de synthèse est composé de deux grandes parties, à savoir la partie de traitement linguistique et la partie de traitement acoustique. L'objet de la première partie est la préparation du texte en entrée dans une forme phonétique, dotée de marqueurs prosodiques, exploitable par le synthétiseur. La deuxième partie consiste à traduire la chaîne phonétique en un message vocal de bonne qualité. Les deux modules du système sont composés d'un ensemble de six phases.

2.2.1. Transcription graphème - phonème

Il s'agit de la première phase du système PARADIS. C'est une phase nécessaire, pendant laquelle une transformation du texte en entrée en son équivalent phonétique est accomplie. Cette opération qui semble évidente à l'aide d'un locuteur humain nécessite en réalité l'application d'un nombre important de règles.

2.2.2. Marquage prosodique

Le marquage prosodique constitue la deuxième phase du système de synthèse. Son objectif est de doter le message phonétique ainsi généré lors de la phase précédente d'indications prosodiques caractérisant l'élocution de l'énoncé (durées des syllabes, positions et durées des pauses, évolution de la courbe mélodique). La représentation phonético-prosodique engendrée est ensuite utilisée par le module de synthèse sonore qui assure la génération du message vocal correspondant.

2.2.3. Découpage en di-syllabes

Le système PARADIS étant un système de synthèse par concaténation de di-syllabes, il est alors nécessaire de décomposer la chaîne phonético-prosodique obtenue en une suite de di-syllabes munies de facteurs prosodiques adéquats. C'est alors la nouvelle chaîne obtenue qui sera délivrée au synthétiseur.

2.2.4. Extraction des di-syllabes à partir du dictionnaire

Cette étape consiste à choisir dans le dictionnaire ou répertoire de di-syllabes, les unités qui seront effectivement utilisées pour synthétiser la succession des sons désirés. La recherche des di-syllabes est effectuée avec une grande rapidité grâce à l'organisation directe et particulière du dictionnaire. Chaque unité extraite sera traitée par l'intermédiaire du synthétiseur TD-PSOLA, tout en lui appliquant les modifications prosodiques recommandées par le module de traitement linguistique.

La concaténation des différents signaux des di-syllabes consécutives est accompagnée par un lissage au niveau des frontières de concaténation. En effet, les différentes unités acoustiques ne possèdent pas exactement les mêmes caractéristiques acoustiques à leurs frontières. L'opération de lissage est aussi assurée par le synthétiseur TD-PSOLA.

2.2.5. Décodage EPEC

La 5ème phase du système PARADIS est optionnelle. Elle dépend de la nature codée/ non codée des segments de signal répertoriés dans le dictionnaire acoustique. Si le codeur EPEC a été utilisé pour le stockage d'une di-syllabe dans le dictionnaire, le décodage de celle-ci est alors nécessaire après son extraction.

Le principe de notre codeur EPEC (Extandable Prediction period Extraction Coding - codage par extraction de la période de prédiction extensible) repose sur la séparation du signal de parole en deux composantes : composante de bruit et composante voisée. La composante de bruit est conservée intégralement sans codage. Cependant, la composante voisée présente une grande richesse que nous avons exploitée pour faire le codage. En effet, il a été remarqué (Taori & al., 1995) que sur une longueur de 4 à 5 périodes d'un signal voisé, il y a une grande ressemblance entre les différentes périodes de pitch. Cette ressemblance peut être exploitée pour représenter un nombre de périodes de pitch donné par une seule que nous avons appelé *période de prédiction*. Ceci constitue la phase de codage de l'algorithme EPEC. La phase de décodage consiste à régénérer la redondance qui a été supprimée lors de la phase de codage. Cette opération est basée sur le même principe OLA (OverLap & Add) de l'algorithme TD-PSOLA présenté au paragraphe 4.2.

Signalons que l'algorithme de codage/décodage EPEC ne requiert pas une grande complexité de calcul et peut être implémenté en temps réel sans nécessiter de DSP.

2.2.6. Synthèse TD-PSOLA

L'une des composantes fondamentales de notre système de synthèse de la parole arabe PARADIS est le synthétiseur TD-PSOLA. Il est caractérisé par sa capacité d'action directe sur le signal de parole ainsi que son indépendance vis à vis de tout algorithme de codage. Il garantit ainsi une grande flexibilité au niveau suprasegmental et offre de très bonnes possibilités de variations prosodiques, ce qui apporte plus de naturel à la parole synthétique. Le synthétiseur peut aussi reproduire, avec une complexité de calcul minimale, un signal de parole qui approche au maximum le signal d'origine.

Nous présentons dans la section 4.1 de cet article les détails de l'implémentation de l'algorithme TD-PSOLA, ainsi que les contraintes que nous avons respectées pour bénéficier de ses larges possibilités. Nous présentons ainsi un algorithme optimisé que nous avons développé et intégré dans notre système de synthèse PARADIS.

2.3. Environnement du Système

L'environnement de notre système est constitué d'une collection d'outils permettant son alimentation par un ensemble de données et des informations nécessaires pour sa mise en œuvre ainsi que sa mise à jour éventuelle. Nous présentons ci-dessous les éléments de base de l'environnement de notre système PARADIS.

2.3.1. Langage LSPERT

La transcription phonétique était réalisée classiquement par un ensemble de tests arborescents relatifs au contexte et représentant le formalisme classique d'une règle. On se retrouve alors devant un chevauchement considérable des règles. Pour pallier à ces problèmes et offrir un moyen permettant la mise à jour simple et rapide du module de transcription graphème-phonème, nous avons réalisé un langage de spécification des règles LSPERT. Le phonétiseur est alors généré automatiquement par compilation d'un programme de règles LSPERT.

2.3.2. Module de segmentation et étiquetage automatique

La génération du dictionnaire des di-syllabes a été accomplie de manière complètement automatique, ce qui nous permettra l'élaboration rapide de plusieurs dictionnaires correspondant à différents locuteurs. L'opération est divisée en deux parties principales : la segmentation et l'étiquetage.

2.3.3. Dictionnaire des di-syllabes

Il s'agit de la base de données du système PARADIS. En l'absence de ce composant, le système devient inactif et sans aucune utilité (Chenfour, 2001).

2.3.4. Module d'accès et organisation du dictionnaire des di-syllabes

En vue d'une meilleure exploitation du dictionnaire acoustique et afin de fournir rapidement les unités demandées par le synthétiseur, nous avons opté pour une organisation particulière du dictionnaire des di-syllabes. Tous les accès sont alors rendus

directs grâce à une technique de Hash-Code basée sur un code di-syllabique approprié (voir paragraphe 7).

3. Module de traitement linguistique

Le module de traitement linguistique ayant une relation directe avec le texte en entrée, il dépendra alors étroitement de la langue étudiée. Nous sommes donc amenés à citer les différents problèmes qu'on peut rencontrer dans cette phase de transcription, ainsi que la manière de résolution qui peut être classique ou basée sur la notion de règles compilées. Cette dernière vision, étant de nature intelligente et permettant l'évolutivité du système de transcription, sera retenue avantageusement.

Le choix de l'unité de concaténation ainsi que la décomposition du texte en des unités de structure bien choisie fera suite au module linguistique qui se termine en fin par l'étude prosodique du texte. En effet, la reproduction d'une unité acoustique doit respecter les spécifications faites lors de cette première étape linguistique.

3.1. Transcription graphèmes-phonèmes d'un texte arabe

Le premier problème rencontré dans la synthèse de la parole à partir du texte est celui de la transcription graphèmes-phonèmes. Cette opération naturellement réalisée par un locuteur humain, cache derrière elle l'application d'un ensemble de règles qui diffèrent selon la langue étudiée. La langue arabe, qui possède ses propres symboles graphiques présente plusieurs discordances entre ce qui s'écrit et ce qui se prononce.

En fait, certaines lettres ou groupes de lettres changent de prononciation suivant leur contexte et nécessitent ainsi l'établissement d'un ensemble de règles pour la transformation automatique des graphèmes en phonèmes. Le module TGP (Transcription Graphèmes-Phonèmes) doit alors cerner tous les problèmes et il doit employer l'approche optimale capable de lui fournir toutes les possibilités d'implantation et d'adaptation des règles de transcription. Avant de discuter le choix de la méthode de spécification des règles, nous présentons brièvement une liste de problèmes que le module TGP doit résoudre. Pour une simplicité d'écriture et de manipulation directe par clavier des codes phonétiques, nous avons adopté un codage approprié que nous présentons dans le tableau suivant :

ي	و	هـ	ن	م	ل	ك	ق	ف	غ	ع	ظ	ط	ض	ص	ش	س	ز	ر	ذ	د	خ	ح	ج	ث	ت	ب	ء
y	w	h	n	m	l	k	q	f	g	c	V	T	D	S	^	s	z	r	v	d	x	H	j	~	t	b	e

• *Problème du [Al] (ال) lunaire ou solaire :*

Le groupe de lettres (ال) n'est pas toujours prononcé [eal]. En effet le [l] peut être prononcé ou non suivant la nature de la consonne suivante. On divise alors l'ensemble des consonnes arabes en deux parties : une première partie des consonnes appelées *consonnes lunaires* et dont la présence après le [eal] exige la prononciation du [l].

Exemple :

القمر → [*eal*qamar]. Qui veut dire la lune. (D'où le terme lunaire).

Cette première partie est constituée de 14 consonnes :

{ي، و، ه، م، ك، ق، ف، غ، ع، خ، ح، ج، ب، ء}

Avec ء un représentant de l'ensemble : {آ، ئ، إ، ؤ، أ}

La deuxième partie regroupe le reste des consonnes qui sont appelées *consonnes solaires*. La présence de l'une de ces consonnes après le [eal] empêche la prononciation du [l], et la consonne sera forcément géminée.

Exemple :

الشّمس → [*ea*^^ams]. Qui veut dire le soleil. (D'où le terme solaire).

Cette deuxième partie est constituée de 14 consonnes :

{ن، ل، ظ، ط، ض، ص، ش، س، ز، ر، ذ، د، ث، ت}

• *Elision du Alif [A] (ا) :*

Nous avons vu que le (ا) et bien prononcé un [ea] ou (A), mais dans certains contextes celui-ci n'est pas prononcé, comme dans les exemples suivants :

ذَهَبوا → [vahabU] .

بالدّار → [biddAri].

• *Problème des liaisons :*

En arabe, une voyelle longue en fin d'un mot suivi par un deuxième mot qui commence avec un [Al], sera transformée en voyelle courte et le [A] du deuxième mot sera éliminé. Enfin, les deux mots seront transformés en un seul, comme dans les exemples :

في البَيت → [filbayti]. Qui traduit en fait : فلبَيت

في الدّار → [fiddAri]. Qui traduit : فدّار

Remarquons avec le deuxième exemple qu'il y a un problème de liaison et un problème du [Al] solaire ce qui a entraîné la suppression de 4 graphèmes pour la transcription (ي ال).

• *Changement de graphèmes avant une pause :*

En arabe, lorsqu'il s'agit d'une pause, celle-ci est marquée par la transformation du dernier signe diacritique du dernier mot en un *soukoun* (ْ). Si la dernière consonne du dernier mot du groupe de souffle est un [t] (ة), celle-ci sera transformée en un [h](ـه), comme dans les exemples :

ذَهبتُ إلى الجامعة → [vahabtu eilaljAmicah].

Où le groupe [ti] est transformé en [h].

سَعيدٌ → [sacId].

Avec suppression du groupe [uN].

• *Problèmes des irrégularités :*

Un ensemble de mots n'est pas conforme aux règles de prononciation de certaines voyelles telles que la voyelle [a]. Celle ci sera prononcée inhabituellement longue.

Exemples :

هَذَا qui se lit : [h*A*vA] alors que sa traduction directe est [h*a*vA].

هَؤُلاءِ qui se lit : [h*A*eulAei].

ذَلِكَ qui se lit : [v*A*lika].

Ces mots peuvent être mis dans un dictionnaire des mots irréguliers. Cependant, l'handicap est que ces mots acceptent plusieurs possibilités d'affixation.

Exemples :

فَبِذَلِكَ → [fabiv*A*lika].

كَهَؤُلاءِ → [kah*A*eulAei].

فَبِذَلِكُم → [fabiv*A*likum].

Ces mots seront alors vus comme des groupes de lettres avec une prononciation particulière, indépendante du contexte.

• *Prononciation des nombres :*

Les nombres changent de prononciation suivant la position grammaticale dans la phrase, et suivant le genre (masculin ou féminin). Le changement se traduit par une transformation des voyelles ou de tout un groupe de graphèmes correspondants.

Exemple 1. Changement suivant la position grammaticale :

رَأَيتُ 5 رِجالٍ → [xamsat**a**]. (j'ai vu 5 hommes)

إِنَّهُم 5 رِجالٍ → [xamsat**u**]. (ils sont 5 hommes)

3.2. Approche de transcription classique

La transcription phonétique était réalisée classiquement par un ensemble de tests arborescents relatifs au contexte et représentant la formulation classique des règles. Cet ensemble de tests se transforme vite en un chevauchement considérable des règles. La mise à jour ou même la lisibilité des règles est alors une opération très délicate, et tout changement ou insertion d'une nouvelle règle peut se répercuter sur le reste des règles. Nous avons donc utilisé un formalisme bien adapté pour l'organisation des règles de manière à offrir à tout concepteur de règles une grande facilité à formuler, supprimer ou rectifier les règles. La spécification des règles de transcription doit être indépendante de la manière de leur intégration dans le système. Il s'agit du modèle avec compilation des règles.

3.3. Réalisation d'un compilateur de règles pour le module TGP

3.3.1. Principe de base

Toutes les règles de transcription sont spécifiées indépendamment les unes des autres, à l'aide d'une notation simple et régulière. Ces règles seront ensuite transformées à l'aide d'un compilateur de règles en leurs équivalentes classiques. Un moteur d'inférence permettra enfin, lors de l'analyse du texte, de choisir et exécuter les règles suivant la vérification de leurs conditions de déclenchement et suivant un ordre de

particularité bien précis. La génération de l'équivalent classique étant une opération automatique, et la spécification d'une règle étant facile et indépendante, ceci permettra une grande souplesse de mise à jour des règles de transcription. La description des différentes transformations linguistiques pourra être facilement envisageable.

3.3.2. Format des règles

Une règle de transcription est une règle qui substitue une chaîne de caractères à une autre, lorsque cette dernière se situe dans un contexte particulier traité par la règle. Un contexte peut être divisé en deux parties : un contexte gauche et un contexte droit. Le contexte gauche (respectivement droit) représente un ensemble de caractères se trouvant à gauche (respectivement à droite) de la chaîne à transcrire. Inspirés par le formalisme couramment utilisé en linguistique (Chomsky & Halle, 1968), nous avons choisi la forme d'un système de réécriture pour une règle de transcription :

$pg \rightarrow pd \,/\, cg + cd$;

Cette règle signifie que la chaîne *pg* (partie gauche) sera transformée en *pd* (partie droite) si les contextes gauche et droit (*cg* et *cd*) sont vérifiés. Notons la souplesse et la possibilité de configuration d'une telle représentation jusqu'au format absolu : $pg \rightarrow pd$; représentant ainsi une règle qui se déclenche sans contraintes. Cette représentation nous a incité au développement d'un outil complet pour la génération automatique du module TGP : Le Langage de SPEcification des Règles de Transcription LSPERT.

Le schéma global de la compilation et de l'évaluation des règles est présenté dans la figure 2.

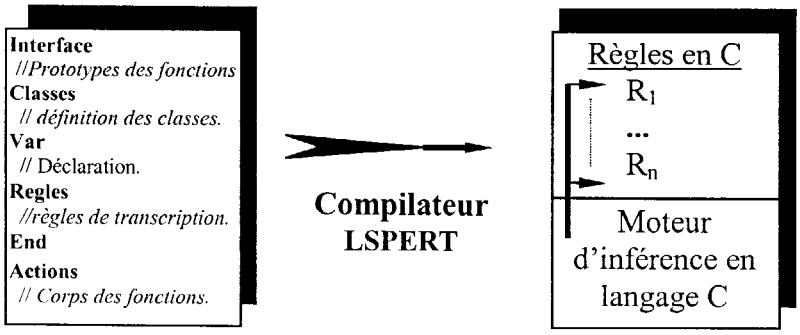


Figure 2 : *Schéma du compilateur de règles LSPERT.*

3.3.3. Préparation des Règles

Notre schéma de transcription se base sur 7 types de Règles qui ont été étudiées en vue d'être formulées avec LSPERT :

- Mots irréguliers,
- symboles mathématiques,
- symboles de ponctuation,
- les abréviations,
- la transcription directe,
- les nombres,
- et les transcriptions dépendant du contexte.

3.3.4. Vitesse de transcription

Nous avons remarqué que la manière avec laquelle il était possible de spécifier les règles indépendamment les unes des autres avait un inconvénient. Avant de trouver la règle déclenchable, le moteur d'exécution des règles devrait parcourir toutes les règles qui figurent avant celle-ci, ce qui retarde le processus de transcription. Une solution convenable consiste alors à classifier les règles suivant le graphème ou le groupe de graphèmes à transcrire. Avec une telle classification, on aboutit à une optimisation du nombre de règles invoquées. Il suffit alors d'ajouter quelques tests supplémentaires qui permettent de situer la chaîne à transcrire dans la classe adéquate.

4. Le synthétiseur TD-PSOLA

Plusieurs modèles de synthèse de la parole ont été proposés et développés dans le but de représenter le signal de parole avec un ensemble réduit de paramètres et une fonction mathématique pouvant le reproduire à partir de ces paramètres. Cette modélisation a l'avantage de réduire le débit binaire du signal de parole dans le but d'une transmission rapide ou un stockage optimal du signal acoustique. Nous pouvons citer le cas du modèle de prédiction linéaire LPC qui a été adopté pendant plusieurs années (Saito et Itakura, 1966), (Atal et Schroeder, 1967). Cependant, avec ce modèle la parole synthétique n'a pas atteint le naturel souhaité. Nous pouvons aussi signaler les travaux basés sur les synthétiseurs à formants (Klatt, 1990), qui malheureusement nécessitent un effort considérable pour l'établissement des règles de transition.

La méthode PSOLA (Pitch Synchronous OverLap & Add), qui est une méthode non paramétrique de synthèse de la parole (Charpentier et Stella, 1986), a cependant donné de meilleurs résultats. Sa variante dans le domaine temporel TD-PSOLA (Time Domain PSOLA) est relativement flexible et moins complexe (Moulines et Charpentier, 1990), (Dutoit, 1993), (Chenfour, 1997).

D'une part, la méthode TD-PSOLA consiste à faire une séparation entre l'algorithme de codage et la méthode de synthèse, la rendant ainsi indépendante de tout algorithme de codage et considérant l'entrée du synthétiseur TD-PSOLA sous forme d'échantillons du signal non codés (Figure 3).

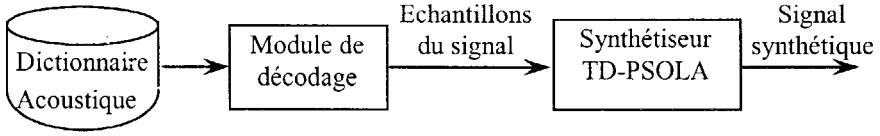


Figure 3 : *Séparation du module de Codage/Décodage et du module de synthèse TD-PSOLA*

D'autre part, elle est connue par sa capacité d'action directe sur le signal de parole tout en apportant une grande souplesse de modification de la fréquence fondamentale et de la durée des différentes portions du signal.

Un autre avantage de TD-PSOLA par rapport aux autres techniques existantes est qu'elle opère dans le domaine temporel, d'où sa simplicité et son temps de réponse nettement rapide.

Dans ce paragraphe, nous rappelons en premier lieu les principes de synthèse TD-PSOLA. Nous présentons ensuite notre adaptation de l'algorithme dans le but d'une

meilleure exploitation de toutes les possibilités de la technique TD-PSOLA, notre objectif étant la réalisation d'un système de synthèse de la parole arabe temps réel et de haute qualité.

4.1. Principe de la technique de synthèse TD-PSOLA

Dans un système de synthèse de la parole à partir du texte, il est nécessaire de modifier les paramètres prosodiques du signal de parole d'analyse pour produire les spécifications mélodiques extraites à partir du texte. Ces facteurs mélodiques sont traduits par une variation de la durée et de la fréquence fondamentale.

La solution proposée avec TD-PSOLA est de considérer le signal de parole comme une séquence de fenêtres de signaux à court-terme qui se recouvrent, centrées sur les marques de pitch successives du signal.

Ces signaux à court-terme peuvent être extraits à partir du signal par une analyse automatique qui constitue la première phase du synthétiseur TD-PSOLA. Elle consiste à appliquer au signal d'origine une suite de fenêtres de *Hanning* qui se recouvrent et permettent l'extraction des signaux à court-terme.

Les périodes de pitch sont représentées par les distances entre les signaux à court-terme. Une re-superposition des signaux à court-terme permettra d'obtenir le signal d'origine.

L'abaissement de la fréquence fondamentale (qui correspond à une augmentation du pitch) sera exprimé par un écartement entre les fenêtres (ou les signaux à court-terme). L'élévation consistera à rétrécir la distance entre les fenêtres.

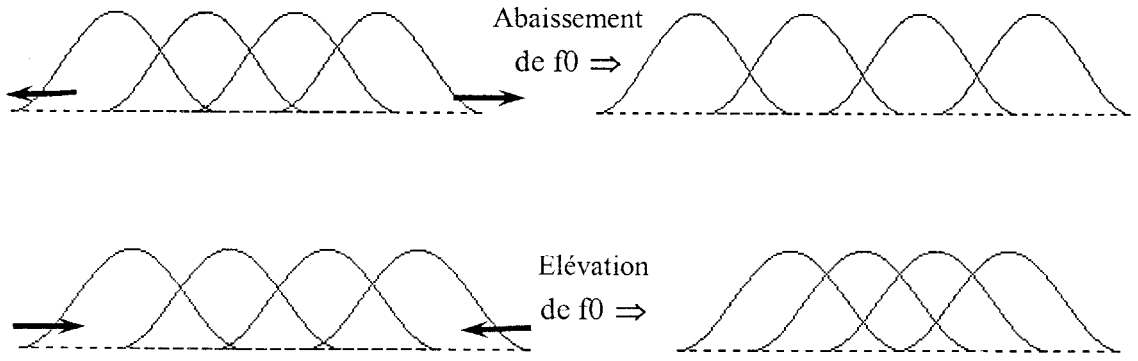


Figure 4 : *Procédure de modification de la fréquence fondamentale*

La modification de la durée globale d'une portion de signal est l'opération qui consiste à dupliquer ou éliminer quelques signaux à court-terme de façon à obtenir la durée désirée du signal synthétique.

Remarque :

En fait, les modifications concernent uniquement la partie voisée qui est considérée comme la partie qui transporte la majorité des manifestations prosodiques. Quant aux sons non voisés, ils sont repris intégralement sans changement.

4.2. Algorithme de synthèse TD-PSOLA

Après avoir présenté le principe de la technique de synthèse TD-PSOLA, nous pouvons décrire l'algorithme correspondant. Celui-ci requiert trois étapes successives : analyse pitch-synchrone et extraction des signaux à court-terme d'analyse, modifications prosodiques apportées aux signaux à court-terme d'analyse et production des signaux à court-terme de synthèse, enfin le calcul du signal synthétique par recouvrement - addition des signaux à court-terme de synthèse.

1- L'analyse pitch-synchrone du signal de parole d'origine extrait du dictionnaire. Elle consiste à le décomposer en une séquence de signaux à court-terme S_i. Ceux ci sont obtenus en multipliant le signal d'origine par une séquence de fenêtres d'analyse h_i :

$$S_i(n) = h_i(n - M_i) \times S(n) \qquad (1)$$

Avec M_i sont les marques de pitch. Elles sont distribuées de manière synchrone avec les périodes de pitch sur les portions voisées du signal, et arbitrairement sur les portions non voisées. Chaque fenêtre d'analyse h_i est centrée sur la marque de pitch M_i correspondante, et a une longueur choisie de manière à couvrir au minimum deux périodes de pitch.

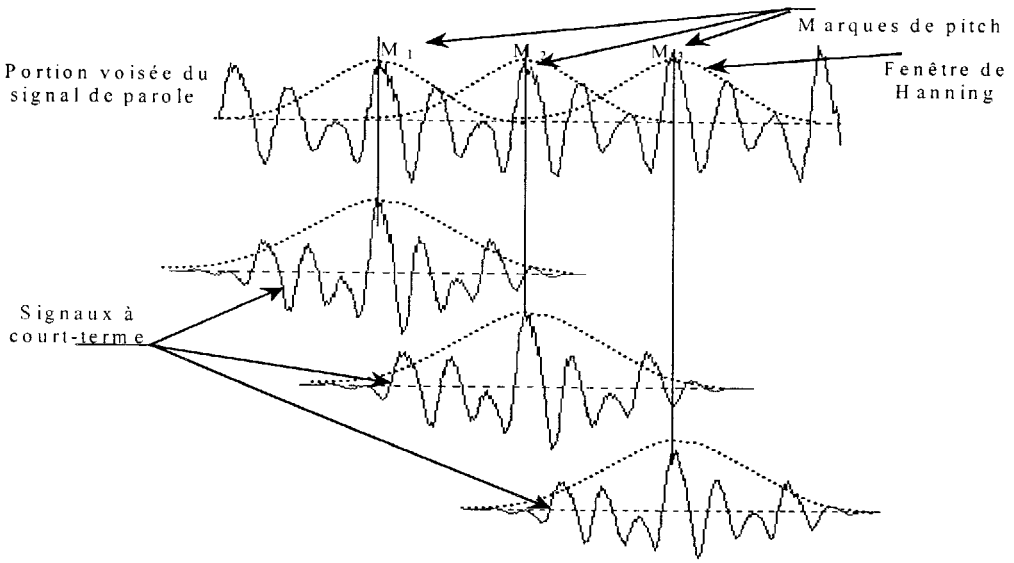


Figure 5 : *Extraction des signaux à court-terme à partir d'un signal de parole par une analyse TD-PSOLA.*

2- Modifications prosodiques apportées aux signaux à court-terme d'analyse. La séquence des signaux à court-terme dégagés est reprise pour produire une nouvelle séquence de signaux synthétisés notés $\hat{S}_k(n)$, synchronisés avec un nouvel ensemble de

marques de pitch de synthèse notés $M_k^{'}$. Les nouvelles marques de pitch sont déterminées en fonction des spécifications prosodiques. Ainsi, elles seront plus ou moins écartées si une modification de pitch est demandée. En général, $M_k^{'} = FMp \times M_i$; FMp étant le facteur de modification de pitch précisé par le module prosodique. Par ailleurs, il n'y a pas une correspondance exacte entre les marques de pitch de synthèse et celles d'analyse car on peut être amené à éliminer ou à dupliquer quelques marques. Ceci est effectué puisque le nombre de marques de pitch détermine la durée du signal synthétique, qui est aussi spécifiée par le module prosodique. Une application est alors définie entre les marques de pitch de synthèse et les marques de pitch d'analyse, en accord avec les facteurs de modifications de pitch et de durée soit : $M_k^{'} \rightarrow M_{\varphi(k)} = M_i$. L'application fait correspondre en même temps les signaux à court-terme de synthèse $(\hat{S}_k(n))$ aux signaux à court-terme d'analyse $(S_i(n))$. Les marques de pitch de synthèse vont indiquer les distances entre les différents signaux à court-terme de synthèse. Ces derniers sont alors obtenus à l'aide de la relation suivante :

$$\begin{aligned} \hat{S}_k(n) &= S_{\varphi(k)}(n + M_{\varphi(k)} - M_k^{'}) \\ &= h_{\varphi(k)}(n - M_k^{'}) \times S(n + M_{\varphi(k)} - M_k^{'}) \end{aligned} \qquad (2)$$

3- Synthèse du signal modifié par recouvrement-addition des signaux à court-terme. Il s'agit de la dernière étape de l'algorithme qui consiste en la synthèse du signal final. Le signal de synthèse est calculé par recouvrement-addition (OverLap & Add) des différents signaux à court-terme de synthèse (Figure 6), à l'aide de l'équation :

$$\hat{S}(n) = \sum_{k=-\infty}^{+\infty} \hat{S}_k(n)$$

Cette équation peut conduire à de mauvaises variations d'énergie qui résultent de la phase de modification de pitch. L'équation est corrigée avec la multiplication par des facteurs de normalisation α_k et la division par un dénominateur obtenu par recouvrement-addition des carrées des fenêtres de synthèse correspondantes.

On obtient l'équation :

$$\hat{S}(n) = \frac{\sum_k \alpha_k \times \hat{S}_k(n) \times h_k(n - M_k^{'})}{\sum_k h_k^2(n - M_k^{'})} \qquad (3)$$

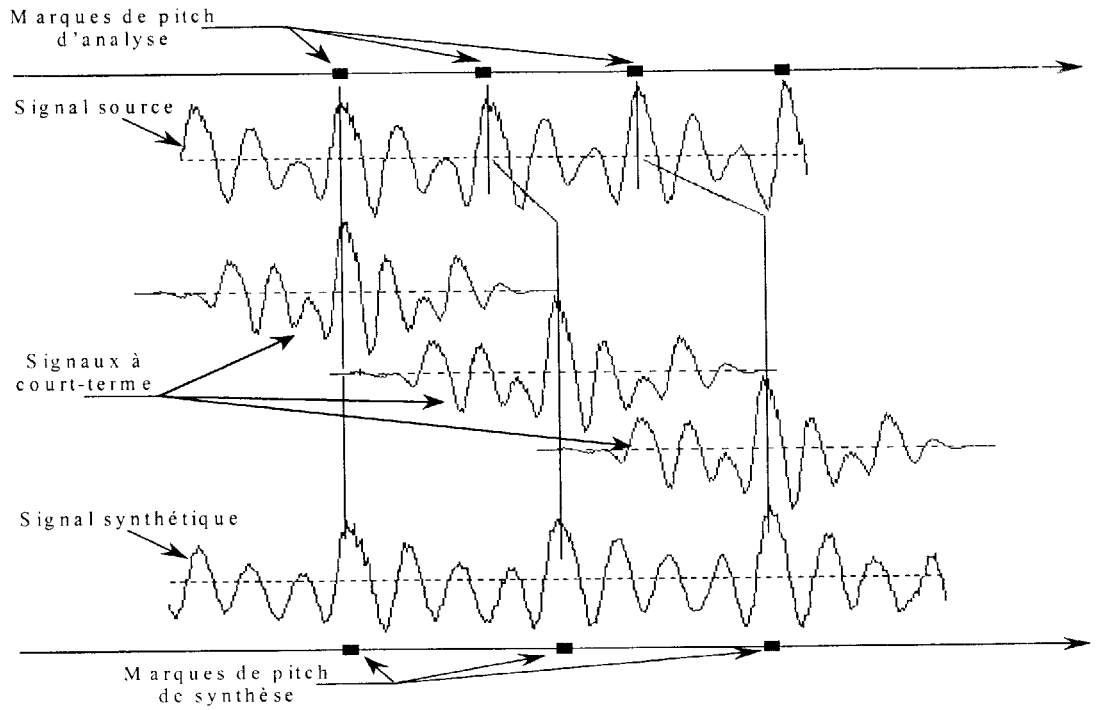


Figure 6 : *Synthèse TD-PSOLA avec modification de la fréquence fondamentale.*

4.3. Contraintes d'implantation

4.3.1. Marquage du pitch

Le marquage de pitch constitue une contrainte fondamentale pour l'application de l'algorithme TD-PSOLA. Le signal doit ainsi être muni d'une suite de marques de pitch distribuées de façon synchrone de la fréquence fondamentale sur les portions voisées du signal. Si on ne respecte pas cette exigence, la procédure de synthèse avec une petite variation prosodique peut produire un signal synthétique de qualité médiocre. Sur les portions non voisées, nous avons utilisé un marquage uniforme toutes les 10 ms. Pour accélérer l'opération de marquage, nous avons procédé, pendant la phase d'analyse, à un marquage automatique de toutes les di-syllabes constituant le dictionnaire acoustique du synthétiseur. La procédure de marquage est basée sur la détection du voisement et du maximum local correspondant au début de chaque période de pitch. Le résultat de marquage est très satisfaisant.

4.3.2. Choix de la fenêtre d'analyse

Chaque fenêtre d'analyse doit être centrée sur la marque de pitch correspondante. La fenêtre doit garantir l'atténuation des lobes secondaires, car elles seront candidates à une sommation ultérieure, et elles portent des informations sur l'identité des fenêtres de signal voisines (Figure 7). Les conditions d'atténuation seront assurées par une fenêtre de Hanning définie de la manière suivante :

$$h(k) = \begin{cases} 0.5 + 0.5 \times \cos(2 \times \pi \times k / N), & pour \quad |k| \leq N/2 \\ 0 & ailleurs \end{cases}$$

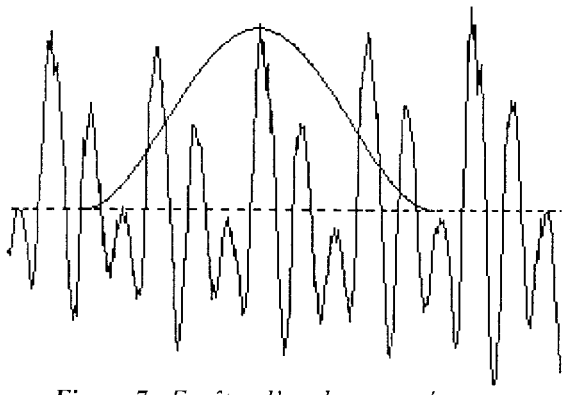


***Figure** 7 : Fenêtre d'analyse centrée sur une marque de pitch.*

Il est possible de considérer des fenêtres de longueur égale à deux fois la période de pitch locale. Nous avons utilisé une fenêtre de durée uniforme et plus longue (égale à 25.6 ms) pour augmenter la longueur de l'intervalle d'interpolation lors de la concaténation des segments du signal, ce qui a donné un meilleur résultat.

4.3.3. Facteurs de normalisation

Nous avons remarqué que l'utilisation de facteurs de normalisation $\alpha_i = 1$ ne provoque aucune dégradation de la qualité de la parole synthétique. Nous avons alors retenu cette valeur pour tous les α_i.

4.3.4. Fréquence d'échantillonnage

Le choix de la fréquence d'échantillonnage est un facteur très important et doit être fait en prenant en considération différents critères. Une fréquence d'échantillonnage très grande est équivalente à une augmentation du nombre d'échantillons à prendre en compte par le synthétiseur. Ainsi un segment d'analyse de 10 ms correspond à 50 échantillons avec une fréquence d'échantillonnage de 5 Khz, et à 400 échantillons avec une fréquence d'échantillonnage de 40 Khz. Le nombre d'opérations est alors plus grand avec une grande fréquence d'échantillonnage.

Sur le plan perceptif, il est démontré que l'oreille présente une sensibilité d'audition vis à vis des fréquences de parole émises, et un pic de sensibilité pour les fréquences voisines de 3000 Hz (Xavier, 1992). Notons aussi que les fréquences des 3 premiers formants sont situées entre 250 et 3800 Hz (Markel & Gray, 1976). Pour les fricatives, l'information se trouve au voisinage de 5000 Hz. Or avec une fréquence d'échantillonnage fe = 5 Khz, la plus haute fréquence qui peut être contenue dans le signal sera de fe/2 = 2500 Hz (théorème de Shannon[1]). Ceci conduit à une grande perte d'informations et une mauvaise qualité de parole. La fréquence d'échantillonnage que nous avons

[1] La perte d'information entre le signal continu et le signal discret correspondant est nulle si la fréquence d'échantillonnage *fe* est au moins égale au double de la plus haute fréquence *fm* contenue dans le signal : $fm \leq fe/2$

adoptée est de 10 Khz, ce qui représente une valeur minimale et suffisante pour avoir une parole presque sans perte d'informations.

4.3.5. Changement de repères

Vu que les signaux à court-terme sont nuls en dehors des fenêtres d'analyse, nous avons défini les valeurs d'un signal à court-terme, à l'aide d'un changement de repère vers le début de la fenêtre, pour les valeurs de n = 0 à 255. La valeur 128 remplace alors la marque de pitch locale M_i, car la fenêtre est centrée sur cette marque de pitch. La relation (1) devient :

$$\begin{cases} S_i(n) = S(n) \times h_i(n-128), & pour \quad n = 0 \quad à \quad 255 \\ 0 & ailleurs \end{cases}$$

S (n) étant à chaque période d'analyse, un signal de 256 échantillons tirés à partir du signal d'origine; *S (128)* est l'échantillon qui correspond à la marque de pitch courante M_i. La période de pitch locale sera notée $P_i = M_i - M_{i-1}$.

Vu le changement de repère qui touche aussi le signal de synthèse, la valeur *128* remplace la marque de pitch image M_k' de M_i. La relation (2) devient une simple opération de copie :

$$\begin{cases} \hat{S}_k(n) = S_i(n), & pour \quad n = 0 \quad à \quad 255 \\ 0 & ailleurs \end{cases}$$

Avec un pitch de synthèse $P'_k = M'_k - M'_{k-1}$.
La relation (3) devient :

$$\hat{S}(n) = \frac{\sum_{k=0}^{\infty} \alpha_k \times \hat{S}_k(\rho_k(n)) \times h_k(\rho_k(n) - 128)}{\sum_{k=0}^{\infty} h_k^2(\rho_k(n) - 128)}$$

ρ_k est la projection sur les fenêtres de chevauchement. C'est l'image de n sur les fenêtres qui se recouvrent sur l'échantillon n de la fenêtre courante. La relation ρ_k sera déterminée dans les sections suivantes.

4.3.6 .Signaux à court-terme d'addition

Avec une longueur des fenêtres d'analyse égale à 25.6 ms, le maximum de fenêtres qui peuvent se recouvrir à un instant donné est au nombre de quatre (Figure 8). Nous avons donc utilisé 4 signaux à court-terme d'analyse S_{-2}, S_{-1}, S_0 et S_1, et 4 signaux à court-terme de synthèse : $\hat{S}_{-2}, \hat{S}_{-1}, \hat{S}_0, \hat{S}_1$. L'indice 0 correspond à la fenêtre courante.

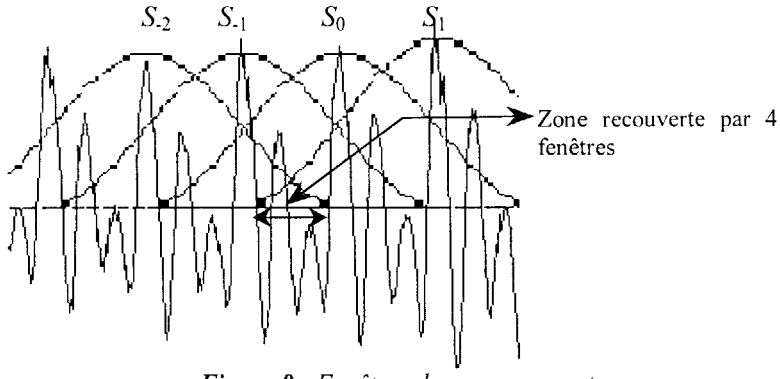


Figure 8 : *Fenêtres de recouvrement*

4.4. Compensation temporelle

La modification de la fréquence fondamentale, par suite des intervalles de temps entre les marques de pitch successives aura pour effet le changement de la durée globale du signal de synthèse. La correspondance entre marques de pitch d'analyse et marques de pitch de synthèse ne doit donc pas être une à une. On pourra être amené à dupliquer ou à éliminer quelques signaux à court-terme afin d'obtenir une durée du signal synthétique égale ou au moins proche de celle du signal d'analyse. Dans le cas d'une élévation de pitch, quelques signaux à court-terme doivent être éliminés, alors que dans le cas d'un abaissement on procède à des duplications des signaux à court-terme (voir figure 10). Il s'agit d'une compensation temporelle par élimination / duplication des signaux à court-terme de synthèse.

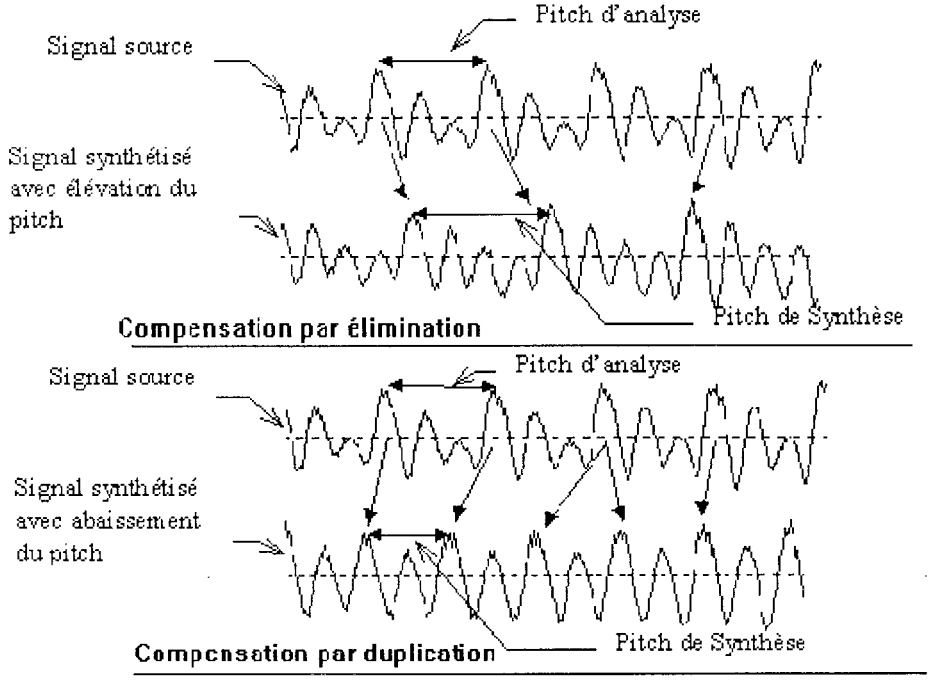


Figure 9 : *Procédure de compensation temporelle par élimination/duplication des signaux à court-terme*

4.4.1. Règle de duplication

Soient M_i les marques de pitch d'analyse, P_i sont les périodes de pitch correspondantes. M'_i les marques de pitch de synthèse, P'_i sont les périodes de pitch correspondantes.

L'algorithme de choix de la fenêtre à dupliquer se base sur la règle suivante :

• **Rd** : Si après la constitution d'un signal à court-terme de synthèse $\hat{S}_k$, la marque de pitch de synthèse M'_k est suffisamment loin de la marque de pitch d'analyse correspondante M_i, on duplique $\hat{S}_k$, c.à.d : $\hat{S}_{k+1} = \hat{S}_k$.

Nous avons considéré qu'une marque de pitch M'_k est suffisamment loin de la marque de pitch d'analyse correspondante M_i si : $M_i - M'_k \geq T_i$ avec $T_i = P_i / 2$(Figure 10). k n'est pas forcément égale à i vu les duplications / éliminations précédentes. On définit alors une relation entre les marques de pitch de synthèse et celles d'analyse φ : φ(k) = i.

Nous obtenons alors la règle équivalente :

• **Rd** : Si ∃ k avec φ(k) = i tel que : $M_i - M'_k \geq T_i$

alors $\hat{S}_{k+1} = \hat{S}_k$

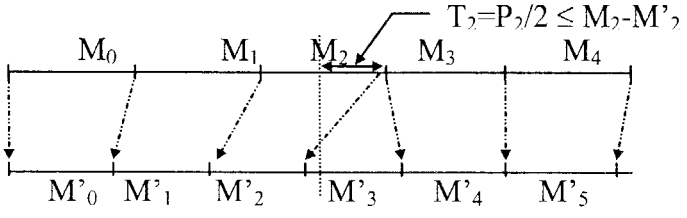
$$or \quad \forall i \quad M_i = \sum_{j=0}^{i} P_j . \quad \forall k \quad M'_k = \sum_{j=0}^{k} P'_j$$

La règle devient :

• **Rd** : Si ∃ k avec φ(k) = i, tel que : $\sum_{j=0}^{i} P_j - \sum_{j=0}^{k} P'_j \geq T_i$

alors : $\hat{S}_{k+1} = \hat{S}_k$,

$M'_{k+1} = M'_k + P'_k$



et φ(k + 1) = i.

Figure 10 : *Compensation temporelle par duplication*

4.4.2. Règle d'élimination

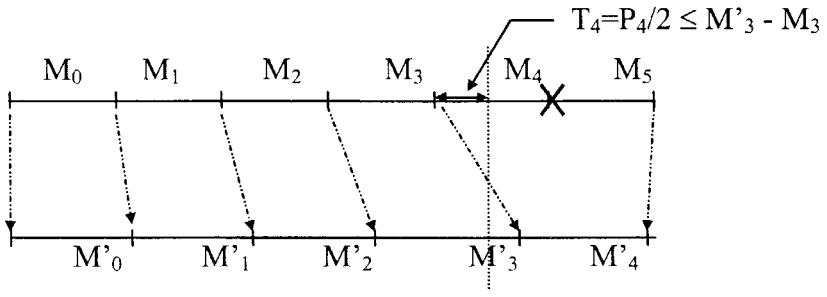
De la même manière, on établit la règle d'élimination suivante :

• **Re** : Si après la constitution d'un signal à court-terme de synthèse $\hat{S}_k$, la marque de pitch de synthèse M'_k est suffisamment loin de la marque de pitch d'analyse correspondante M_i, on élimine le signal d'analyse S_{i+1} centré sur la marque de pitch M_{i+1} (Figure 14).

On obtient la formule équivalente :

• **Re** : si ∃ k avec φ(k) = i, tel que : $\sum_{j=0}^{k} P'_j - \sum_{j=0}^{i} P_j \geq T_{i+1}$

alors : pas de synthèse pour M_{i+1} et $\varphi(k + 1) = i+2$.



avec $T_{i+1} = P_{i+1} / 2$.

Figure 11 : *Compensation temporelle par élimination.*

4.5. Variations prosodiques avec TD-PSOLA

La substance acoustique des paramètres prosodiques étant définie par la fréquence fondamentale, la durée et l'intensité, le modèle de synthèse TD-PSOLA reçoit alors en entrée une chaîne de di-syllabes munie des trois facteurs de variation prosodique : facteur de pitch FP, facteur de durée FD et facteur d'intensité FI. Chaque facteur est un pourcentage de variation muni d'un signe positif ou négatif pour indiquer une augmentation ou une diminution de valeur. Il sera appliqué à l'unité d'origine extraite à partir du dictionnaire TD-PSOLA pour produire les nouvelles spécifications prosodiques.

4.5.1. Modification du pitch

Pour chaque période d'analyse/synthèse, outre le signal d'origine on extrait la période locale d'analyse P_i à partir du dictionnaire acoustique. La période locale de synthèse correspondante sera : $P'_i = P_i + FP \times P_i /100$. Au fur et à mesure de la synthèse, la procédure de compensation temporelle sera appliquée en se basant sur les règles déjà présentées **Re** ou **Rd**, suivant la valeur de FP (inférieure ou supérieure à zéro) et si la condition de déclenchement de la règle est réalisée.

La formule $P'_i = P_i + FP \times P_i /100$, n'est appliquée que sur les portions voisées du signal. Si le segment d'analyse courant est non voisé, aucun changement de pitch n'est effectué ($P'_i = P_i = 100$). Avec cette contrainte, nous avons évité les distorsions qui peuvent résulter d'une modification de pitch appliquée à une portion de signal non voisée. Nous avons remarqué que la modification de pitch qui porte uniquement sur les portions voisées du signal est suffisante pour donner un effet sur le signal de synthèse.

4.5.2. Modification de durée

Le facteur de durée est un paramètre prosodique très important. En effet, la durée des phonèmes est très dépendante du contexte dans lequel ils apparaissent. D'autre part, le découpage du texte en groupes prosodiques va se traduire sur le plan acoustique par des allongements vocaliques et des pauses. Il sera donc primordial de pouvoir varier avec souplesse la durée du signal, selon les spécifications prosodiques qui accompagnent le texte. La technique de compensation temporelle offre une manière efficace pour atteindre cet objectif. Si elle a bien réussi avec une durée égale à la durée du signal d'origine, on pourrait très bien l'utiliser pour obtenir n'importe quelle durée désirée,

toujours par application de la procédure de duplication ou élimination de signaux à court-terme (Figure 12). La question qui sera posée et à laquelle on va essayer de répondre est : quelles sont les marques de pitch à dupliquer ou à éliminer ?

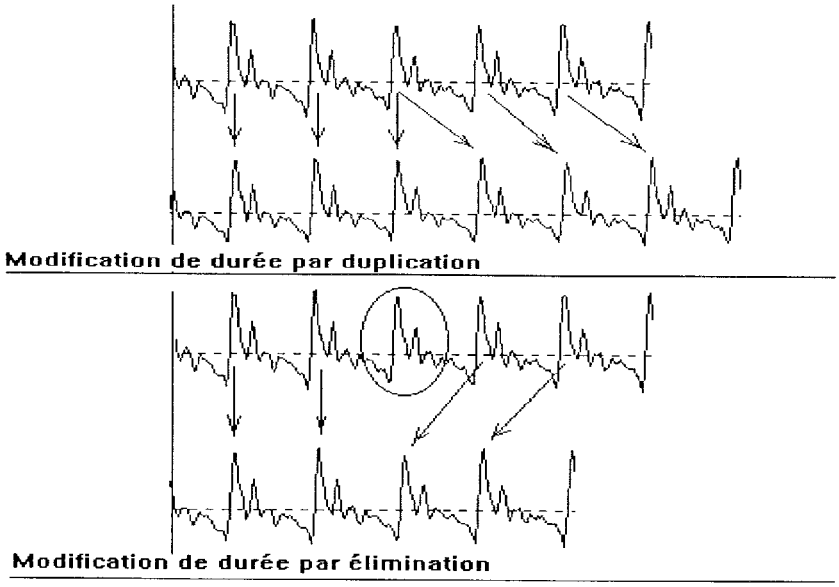


Figure 12 : *Procédure de modification de durée par duplication/élimination des signaux à court-terme.*

Nous avons déjà présenté deux règles pour le choix de la fenêtre à dupliquer ou à éliminer (Re et Rd). L'objectif était la construction d'un signal de synthèse de même durée que le signal d'analyse, ce qui correspond à un facteur FD = 0 (sans changement de durée). Les deux règles doivent alors être généralisées pour prendre en compte le cas de modification de durée, càd un facteur FD non nul.

On définit le facteur multiplicatif de la durée correspondant : FMd = 1 + FD / 100. Si la durée du signal d'analyse est D_a, le signal de synthèse doit avoir une durée $D_s = D_a \times FMd$.

Si FD > 0 alors FMd > 1 par suite $D_s > D_a$. et Si FD < 0 alors FMd < 1 par suite $D_s < D_a$.

Considérons une marque de pitch du signal d'analyse M_n ; la durée du signal jusqu'à M_n est :

$$D_a = \sum_{i=0}^{n-1} P_i$$

La durée du signal de synthèse correspondant est :

$$D_s = \sum_{i=0}^{n-1} (P_i \times FMd).$$

Pour assurer ce changement de durée, Il suffit donc de changer dans les deux règles chaque P_j d'analyse par ($P_j \times FMd$).

Nous aurons alors :

• **Rd** : Si $\exists$ k avec $\varphi(k) = i$, tel que :

$$\sum_{j=0}^{i} (P_j \times FMd) - \sum_{j=0}^{k} P_j' \geq T_i$$

avec $T_i = P_i / 2$

alors : $S_{k+1} = S_k$, $M'_{k+1} = M'_k + P'_k$et $\varphi(k+1) = i$.

• **Re** : Si $\exists$ k avec $\varphi(k) = i$, tel que :

$$\sum_{j=0}^{k} P'_j - \sum_{j=0}^{i} (P_j \times FMd) \geq T_{i+1}$$

alors : Pas de synthèse pour M_{i+1} et $\varphi(k+1) = i + 2$.

avec $T_{i+1}=P_{i+1} / 2$.

4.5.3. Modification d'intensité

On considère le facteur multiplicatif d'intensité FMi = 1 + FI / 100. L'opération de recouvrement-addition OLA du synthétiseur devient :

$$\hat{S}(n) = FMi \times \frac{\sum_{k=0}^{\infty} \alpha_k \times \hat{S}_k(n) \times h_k(n - M'_k)}{\sum_{k=0}^{\infty} h_k^2(n - M'_k)}$$

4.6. Concaténation des di-syllabes

4.6.1. Segments de concaténation

Il est alors actuellement recommandé de choisir des unités plus longues. De manière générale, les unités formées de deux voyelles séparées par un nombre quelconque de consonnes (VC...CV) constituent des segments idéals pour la concaténation. En effet, elles comportent le phénomène de coarticulation à plus long terme et réduisent le problème de discontinuité temporelle créé au cours de la synthèse car les régions de concaténation sont toujours des parties stables formées de voyelles. Cependant, elles engendrent une grande combinatoire. La di-syllabe arabe n'échappe à aucune de ces règles, avec la particularité qu'un mot arabe ne contient jamais plus de deux consonnes consécutives. La forme VCCV constitue alors la structure maximale de la di-syllabe arabe. Nous avons donc choisi la di-syllabe comme unité de base de notre système de synthèse arabe. Celle-ci a donné de meilleurs résultats et une synthèse de bonne qualité. Cependant, le nombre de di-syllabes s'avère beaucoup plus élevé que celui des di-phones. Nous avons pu réduire ce nombre à quelques 6800 unités (Chenfour, 2001).

La définition d'une di-syllabe (Chenfour, Mouradi, Benabbou, 1997) ressemble à la définition du diphone projetée sur l'échelle de la syllabe :

« Une di-syllabe est la portion du signal de parole située entre le noyau vocalique stable d'une syllabe et le noyau vocalique stable de la syllabe suivante ».

Le noyau vocalique stable d'une syllabe correspond à la partie stable de la voyelle qui constitue le noyau de la syllabe. Les mots arabes ne contiennent jamais plus de deux consonnes consécutives, nous obtenons alors six formes possibles d'une di-syllabe :

CV, VCV, VCCV, VC, VCC et V #.

C étant un représentant de la classe des consonnes en nombre de 28 et V représente la classe des voyelles courtes et longues en nombre de 6. Les di-syllabes peuvent être réparties en trois catégories :

- La structure CV rencontrée au début des mots seulement ou plus exactement au

début d'une phrase (après une pause).

- Les 2 structures VCV et VCCV qui occupent des positions médianes.
- Enfin, les structures VC, VCC et V# qui prennent les positions en fin des mots, et plus exactement à la fin des phrases ou avant une pause dans un message vocal.

Ces unités comportent d'une part la coarticulation de voyelle à voyelle à travers les consonnes, d'autre part, elles permettent une amélioration tangible de la procédure de concaténation qui n'agit plus que sur des voyelles, d'où la grande importance de telles unités.

Par ailleurs, la concaténation doit s'accompagner d'une interpolation efficace pour éliminer les distorsions et discontinuités spectrales qui résultent du fait que les 2 di-syllabes sujettes à la concaténation sont tirées de deux contextes différents. Le modèle TD-PSOLA se trouve alors bien adapté à cette opération. En effet, l'opération de recouvrement-addition (OLA) entre le dernier signal à court-terme de la pre-di-syllabe et le premier signal à court terme de la post-di-syllabe, qui sont tous deux des signaux voisés d'une même voyelle, sera considérée comme une interpolation linéaire entre les deux segments.

4.6.2. Génération automatique des voyelles longues

Les six voyelles de l'arabe sont composées de 3 brèves ([a], [u], [i]) et 3 longues ([A], [U], [I]). Les voyelles brèves et leurs correspondantes longues sont liées par un facteur multiplicatif de durée. Nous avons utilisé le principe de duplication des signaux à court-terme pour réaliser des allongements de la durée segmentale des voyelles brèves et produire leurs équivalentes longues. Ainsi, toutes les formes de di-syllabes basées sur les voyelles longues ont pu être générées automatiquement à partir des formes équivalentes basées sur les voyelles courtes. De cette manière, nous avons considéré les six formes possibles à base de 3 voyelles au lieu de 6. Ceci a réduit énormément la combinatoire des di-syllabes.

En fait, nous avons mené une étude sur 3 corpus de parole (un seul texte enregistré avec 3 locuteurs différents) dans le but d'établir le rapport de durée entre une voyelle courte et la voyelle longue correspondante.

Nous avons calculé les valeurs de durées séparément pour les 3 voyelles afin d'observer les facteurs d'allongement relativement à chaque voyelle. Nous avons obtenu le tableau des moyennes suivant :

Locuteur	Locuteur 1	Locuteur 2	Locuteur 3	Moyenne
Durée moyenne de la voyelle courte /a/	59,06	69,24	62,95	63,75
Durée moyenne de la voyelle longue /A/	107,46	140,54	128,95	125,65
Facteur d'allongement	1,82	2,03	2,05	1,97
Durée moyenne de la voyelle courte /u/	54,45	70,99	71,47	65,63
Durée moyenne de la voyelle longue /U/	117,49	142,44	142,91	134,28
Facteur d'allongement	2,15	2,00	2 ,00	2,05
Durée moyenne de la voyelle courte /i/	53,37	67,09	71,96	64,14
Durée moyenne de la voyelle longue /I/	111,43	136,36	142,30	130,03
Facteur d'allongement	2,09	2,03	1,98	2,03

***Tableau 1** : Rapport d'allongement entre les voyelles courtes et leurs équivalentes longues*

Nous remarquons que les valeurs de durée du locuteur 1 sont plus courtes que celles des 2 autres locuteurs à cause de son débit élevé. Cependant, les facteurs d'allongement sont stables et tournent autour de la valeur 2. Les valeurs respectives pour les 3 voyelles /a/, /u/ et /i/ sont : 1.97, 2.05 et 2.03. Puisque les valeurs sont très voisines, nous prenons leur valeur moyenne ≈ 2 qui va alors constituer le facteur d'allongement utilisé indépendamment de la voyelle.

4.6.3. Interpolation OLA

L'opération de recouvrement-addition (OLA) entre le dernier signal à court-terme de la pre-di-syllabe et le premier signal à court-terme de la post-di-syllabe, qui sont tous deux des signaux voisés, sera considérée comme une interpolation linéaire entre les deux segments. Le résultat obtenu avec des fenêtres de longueur deux fois le pitch local est déjà bon. Nous avons remarqué qu'avec des fenêtres de longueur fixée à 25.6 ms, nous pouvons obtenir une légère amélioration de la qualité d'interpolation. Ceci est expliqué par le fait que l'opération OLA peut porter sur quatre fenêtres et non uniquement sur deux. La zone d'interpolation est en effet plus large que lorsqu'on utilise des fenêtres de longueur deux fois la période de pitch.

5. Traitement Prosodique

La substance acoustique des paramètres prosodiques étant définie par la fréquence fondamentale, la durée et l'intensité, le modèle de synthèse TD-PSOLA reçoit alors en entrée une chaîne de di-syllabes munie des trois facteurs de variation prosodique : facteur de pitch *FP*, facteur de durée *FD* et facteur d'intensité *FI*. Chaque facteur est un pourcentage de variation muni d'un signe positif ou négatif pour indiquer une augmentation ou une diminution de la valeur. Il sera appliqué à l'unité d'origine extraite à partir du dictionnaire des di-syllabes pour produire les nouvelles spécifications prosodiques. Lors de la synthèse, le synthétiseur reçoit alors une suite de di-syllabes DS munie chacune des 3 facteurs *FP*, *FD* et *FI* et représentée comme suit : DS *FP FD FI*

Quant aux modifications prosodiques nous avons élaboré un ensemble de modèles constitués de 35 règles pour les allongements vocaliques, le traitement de l'accent et l'insertion automatique des pauses (Chenfour, 2001). Un autre modèle de traitement de la fréquence fondamentale, réalisé par A. Benabbou (Benabbou, 2001), a été intégré dans le système TD-PSOLA. Nous signalons que les modèles ont été déterminés à partir d'une étude réalisée sur un grand corpus de parole enregistré par des locuteurs dont le nombre varie de 3 à 10. Le corpus est constitué aussi bien de textes longs que de phrases courtes pour dégager le maximum de manifestations prosodiques indépendamment de la taille du texte lu.

Puisque la pause constitue un facteur très important dans les allongements syllabiques ainsi que dans le traitement mélodique (ré-initialisation de la courbe mélodique), la première tâche qui est réalisée par le module de génération de la prosodie est alors le marquage des pauses, ensuite la répartition des accents et enfin, les allongements syllabiques parallèlement au traitement mélodique. La séquence complète des tâches effectuées successivement par le module prosodique obéit à l'algorithme ci-dessous :

Algorithme :

Soit la chaîne phonétique à synthétiser.

1- Chercher l'emplacement de la prochaine pause en utilisant le modèle de génération automatique des pauses. Le résultat est une séquence de syllabes, extraites de la chaîne phonétique en entrée, se terminant par une pause dont on indique la durée en ms.

2- On analyse, à l'aide du modèle des accents, la chaîne des syllabes délivrée par la phase (1) et on associe un indicateur d'accent à chaque syllabe.

3- Suivant son indicateur d'accent ainsi que sa position dans la séquence (avant la pause par exemple), chacune des syllabes sera alors traitée respectivement par les modèles concernant les allongements syllabiques et aussi par le modèle mélodique, ce qui se traduit par l'affectation d'un facteur de modification de pitch (FP) et un facteur d'allongement de durée (FD) à chaque syllabe. Le facteur d'intensité (FI) est toujours nul (pas de traitement d'intensité).

4- Projection des facteurs prosodiques déduis suivant l'axe des syllabes sur celui des di-syllabes. Le résultat est une séquence de di-syllabes dotées chacune des 3 facteurs prosodiques :

$$\mathbf{DS_1}\ FP_1\ FD_1\ FI_1\ \mathbf{DS_2}\ FP_2\ FD_2\ FI_2\ \ldots\ \mathbf{DS_n}\ FP_n\ FD_n\ FI_n\ \#\,D$$

avec :

- DS_i (i = 1 à n) est la séquence des di-syllabes
- FP_i FD_i FI_i sont les facteurs de modifications prosodiques associés à la di-syllabe DS_i
- # D indique une pause de durée D.

5- Répéter les étapes de 1 à 4 jusqu'à la fin du texte phonétique.

6. Architecture du dictionnaire

Vu que la taille du dictionnaire peut retarder le processus de recherche, il est primordial d'organiser le dictionnaire sous forme d'un fichier à accès direct. Cependant la taille du bloc de données représentant chaque di-syllabe est variable. Nous avons donc utilisé une organisation en un fichier index et un fichier de blocs de données. Les indexes des différents blocs de données sont des clés d'entrée dont l'accès est calculé par une technique de Hash-Code à partir de la forme de la di-syllabe.

6.1. Choix d'un Hash-Code di-syllabique

Nous résumons que les formes possibles d'une di-syllabe sont : CV, VC, VCV, VCC, VCCV et V. On attribue tout d'abord un code à chaque voyelle et à chaque consonne, et on définit les deux ensembles suivants :

V = {1, 2, 3, 4, 5, 6} pour représenter l'ensemble des voyelles.

C = {1, 2, 3,..., 28} pour représenter l'ensemble des consonnes.

Les deux ensembles sont ensuite complétés chacun par le code 0 qui va représenter dans l'ensemble des consonnes la consonne « vide » ou « élément neutre pour la concaténation » notée **ç**. Dans l'ensemble des voyelles il représentera la voyelle « vide » notée **v**.

On aboutit aux deux ensembles complétés :

$V = V \cup \{0\}$

$C = C \cup \{0\}$

En utilisant les deux éléments introduits **ç** et **v**, il serait possible de reformuler tous les modèles de di-syllabes de telle manière à aboutir à une forme générale unifiée :

$V \Leftrightarrow V$**çç**v

$VC \Leftrightarrow VC$**ç**v

$CV \Leftrightarrow$ **v**çCV

$VCV \Leftrightarrow VC$çV

$VCC \Leftrightarrow VCC$**v**

$VCCV \Leftrightarrow VCCV$

Le résultat ainsi obtenu aide à considérer qu'il existe une seule forme générale pour toutes les di-syllabes : VCCV avec $V \in V$ et $C \in C$

D'autre part le cardinal $|V|$ de V est égal à 7, celui de C est 29. Ainsi 3 bits suffiraient pour coder tous les éléments de V; et 5 bits pour coder tous les éléments de C. Une di-syllabe VCCV peut alors être représentée par un code sur 16 bits qui la détermine. Ce Hash-code H (VCCV) est une simple juxtaposition des codes des différents éléments de la di-syllabe en base {3bits, 5 bits, 5 bits, 3 bits}.

V	C	C	V
3bits	5bits	5bits	3bits

6.2. Architecture du dictionnaire des di-syllabes

Nous allons nous baser sur ce qui a été vu précédemment pour trouver le format du dictionnaire et la méthode de recherche appropriée.

Vu que le bloc de données représentant chaque di-syllabe change de taille, le dictionnaire est alors un fichier de blocs de données de tailles variables. Il est obligatoire d'utiliser un fichier index comportant pour chaque di-syllabe la clé d'entrée au fichier de blocs de données, ainsi que la taille du bloc correspondant. Le fichier de blocs de données est alors un fichier binaire à accès direct puisque les clés d'entrée seront récupérées à partir du fichier index.

Le fichier index est un fichier binaire de blocs de taille fixe. Chaque bloc occupe la position correspondante au code di-syllabique associé.

6.3. Méthode de recherche

Etant donnée une di-syllabe VCCV, on calcule tout d'abord son code H (VCCV). Le code représente une clé d'accès direct dans le fichier index. On récupère la clé d'entrée dans le dictionnaire de paramètres, ainsi que la taille du bloc à extraire. Enfin, avec un accès direct au fichier de données, on extrait les informations recherchées (voir Figure 13).

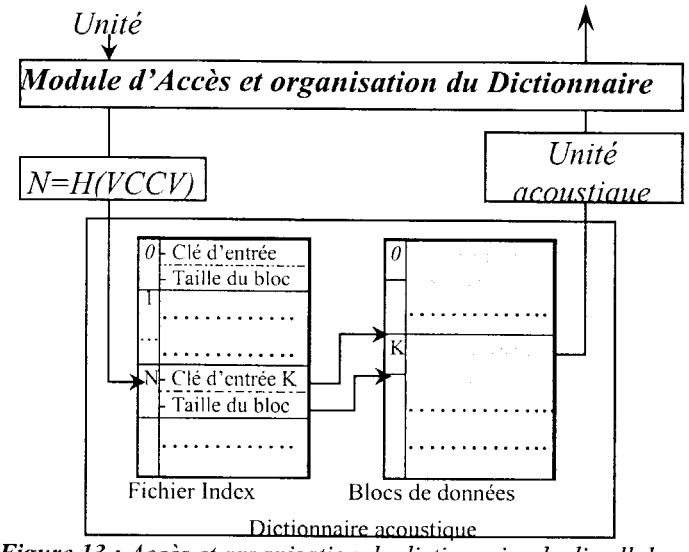


Figure 13 : Accès et organisation du dictionnaire de di-syllabes.

7. Protocole expérimental d'évaluation

Dans la première partie de ce papier, nous avons présenté les détails techniques d'organisation de notre système PARADIS et plus particulièrement des techniques utilisées aussi bien lors de la synthèse que lors de l'analyse. Dans ce paragraphe, nous passons à l'évaluation de la parole synthétique obtenue. La netteté et l'intelligibilité ne sont plus des résultats à prouver vu la nature même de l'algorithme TD-PSOLA (Chenfour, 1997). Nous nous sommes donc concentrés sur des tests de qualité de la prosodie synthétique.

Pour cela, nous avons procédé par deux démarches complémentaires :

- La première démarche consiste à effectuer des tests d'équivalence basés sur la comparaison entre le système prosodique naturel et son équivalent généré par le synthétiseur.
- La seconde est basée sur le jugement du « premier concerné » par le système de synthèse : l'auditeur.

A ce stade, nous pouvons remarquer que la première démarche a été intégralement accomplie lors de l'implémentation des modèles. En effet, pour nous assurer de la validité des modèles ainsi que celle du code implémenté, nous avons effectué des tests d'équivalence au fur et à mesure de la réalisation des modules prosodiques.

La seconde démarche a constitué alors la véritable stratégie d'évaluation de la prosodie synthétique. Une telle démarche est constituée des quatre étapes suivantes :

1. Préparation d'un corpus d'analyse.
2. Elaboration des questionnaires d'analyse.
3. Choix des auditeurs et collecte des réponses aux questionnaires.
4. Traitements et analyses des résultats.

Le premier stade de la démarche consiste alors à élaborer un corpus d'analyse. Celui-ci sera constitué de phrases synthétiques dont la fonction linguistique et la longueur doivent être choisies conformément à la nature du modèle à évaluer. Ainsi par

exemple, pour l'évaluation de la qualité des pauses, les énoncés synthétisés doivent comporter des pauses de paragraphes, des pauses de phrases et des pauses sans ponctuations. L'énoncé doit alors être assez long pour couvrir les différents types de pauses. L'évaluation des accents quant à elle peut se contenter de phrases plus courtes.

Un deuxième stade consiste à concevoir le ou les questionnaires à utiliser pour enregistrer les jugements des différents auditeurs. Il est à remarquer que les auditeurs trouvent en général beaucoup de difficultés à apporter des jugements sur un système de synthèse. La difficulté devient importante lorsqu'on utilise des « questions ouvertes » dont la réponse pouvant être quelconque. Toutes les questions que nous avons alors utilisées dans les différents questionnaires sont des « questions fermées » à réponse bien définie dans un ensemble restreint de possibilités. Le sujet ne votera que par une seule réponse choisie sur une liste qui lui est proposée. Pour certaines questions, la liste des réponses est organisée selon une échelle de valeurs bien déterminées, attribuées aux différentes réponses (Exemple : 1-médiocre, 2-moyenne, 3-bonne, 4-très bonne, 5-excellente). La fonction de la réponse n'est alors pas seulement qualitative, mais également quantitative, et permet ainsi de mesurer les différents jugements.

Dans un troisième stade, on réalise la collecte des résultats saisis au niveau des questionnaires. Il s'agit d'une séquence d'évaluations perceptives des fonctions prosodiques et linguistiques générées automatiquement par le synthétiseur. Les évaluations sont transcrites par chaque auditeur sous forme de réponses aux différentes questions du formulaire.

Enfin, vient le dernier stade essentiel qui consiste à analyser les résultats obtenus et les transformer en des descriptions quantitatives ou qualitatives permettant une évaluation claire de la qualité de la synthèse vocale. Nous avons donc utilisé « Sphinx » comme système d'analyse des données dans notre processus d'évaluation.

En conclusion générale, la plupart des auditeurs (70%) ont jugé « très bonne » la qualité de synthèse. Nous n'avons reçu aucun jugement « médiocre » et 95% des jugements sont entre « bonne » et « excellente ».

8. Conclusion

Nous nous sommes basés sur un ensemble de techniques pour réaliser une première phase linguistique qui prépare le texte en entrée du synthétiseur. Le module TGP a été obtenu automatiquement par application du compilateur des règles LSPERT au fichier des règles. Nous avons aussi intégré quelques règles de correction automatique du texte en entrée dans le cas d'une absence d'une voyelle, dans certaines situations particulières.

L'utilisation de la technique TD-PSOLA et d'un dictionnaire de di-syllabes nous a permis d'affirmer que les modifications pitch-synchrones du signal de synthèse permettent d'obtenir une très bonne qualité de synthèse. Notons par ailleurs, que les optimisations que nous avons apportées au synthétiseur TD-PSOLA nous ont permis d'obtenir des réponses rapides. La qualité de reconstruction du signal est satisfaisante.

L'implantation et l'expérimentation de ces différentes techniques de synthèse de la parole nous ont fait découvrir plusieurs caractéristiques du processus de phonation et de la nature du signal de parole. Ceci nous a ouvert la voie à de multiples recherches et nous a facilité la tâche pendant une autre période de recherche sur le traitement pro-

sodique dans notre laboratoire IHM. Les résultats des travaux concernant le module prosodique combinés à ceux décrits dans cet article constituent ensemble le système PARADIS. Une version complète du système est actuellement opérationnelle.

BIBLIOGRAPHIE

Atal B.-S, Schroeder M. R., *Predictive Coding Of Speech Signals*, Proc. Conf. Commun. and Process., pp. 360-361,(1967).

Benabbou A., *Etude et génération de la mélodie pour le système TD-PSOLA*, thèse pour l'obtention du titre de docteur en informatique, ENSIAS Rabat, Université Mohammed V, (2001).

Benabbou A, Chenfour N., Mouradi A., *Study and Quantification of the Declination for the Arabic Speech Synthesis in System PARADIS*, LREC-2002, (2002).

Charpentier F. J., Stella, M. G., *Diphone Synthesis Using an Overlap-Add Technique for Speech Waveforms Concatenation*. Proc. Int. Conf. on Acoustics, Speech and Signal Processing, Tokyo, pp. 2015-2018, (1986).

Chenfour N., *Réalisation d'un système de synthèse de la parole arabe à partir du texte par concaténation de di-syllabes*, thèse pour l'obtention du titre de docteur de troisième cycle, Faculté des sciences Rabat, Université Mohammed V, (1997).

_____ , *Réalisation d'un système de synthèse de la parole arabe à partir du texte (PARADIS) : Etude et génération des pauses et des durées syllabiques*, thèse pour l'obtention du titre de docteur en informatique, ENSIAS Rabat, Université Mohammed V, (2001).

Chenfour N., Benabbou A, Mouradi A., *Synthèse de la parole arabe TD-PSOLA, génération et codage automatiques du dictionnaire*, ISIVC'2000, RABAT, pp. 112-122, (2000).

Chomsky N., Halle M., *The Sound Pattern of English*, Harper & Row, New York, (1968).

Dutoit T., *High Quality Text-to-Speech of the French Language*, Ph. D. dissertation, Faculté polytechnique de Mons, (1993).

Klatt D. H., Klatt L. C., « Analysis, Synthesis, and Perception of Voice Quality Variations among Female and Male Talkers », J. Acoust. Soc. Amer., vol. 87, n° 2., pp. 820-857, (1990).

Markel J. D. And A. H. Gray, Jr., *Linear Prediction of Speech. Springer-Verlag*, Berlin Heidelberg, New York, (1976).

Moulines E., Charpentier F. J., « Pitch Synchronous Waveform Processing Techniques for Text-to-Speech Synthesis Using Diphones », Speech Communication, vol. 9, n° 5-6, (1990).

Mouradi A., *Synthèse de la parole arabe à partir du texte par la méthode des diphones*, thèse de Doctorat, Fac. des Sciences, Rabat, Université Mohammed V, (1985).

Saito S., Itakura, F., *The Theoritical Consideration of Statistically Optimum Methods for Speech Spectral Density*, Report n°.3107, Electrical Communication Laboratory, N.T.T., Tokyo, (1966).

Taori R., Sluijter R.J. And Kathman E., *Speech Compression Using Pitch Synchronous Interpolation*, ICASSP 95 (pp. 512-515), (1995).

Xavier Marsault, *Compression et cryptage en informatique*, Edition Hermes, (1992).