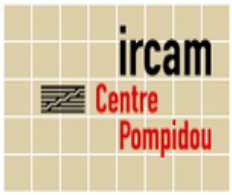


Mémoire de stage de master 2 ATIAM

Synthèse du chant

Luc Ardaillon

01/03/2013 - 31/07/2013

Laboratoire d'accueil : IRCAM - Équipe Analyse-Synthèse des sons

Encadrant : Axel Roebel

Résumé

L'objet du stage présenté ici concerne l'évaluation et l'adaptation de technologies existantes dans le cadre de la synthèse du chant, afin d'identifier les problèmes à résoudre et de proposer des solutions adaptées. Pour cela, un système de synthèse basé sur une méthode de concaténation et transformation d'unités a été développé. Un tel système doit permettre, à partir d'une partition et d'un texte, de créer un fichier audio de voix chantée dont le rendu doit être le plus naturel et le plus expressif possible. Une base de données préalablement enregistrée et segmentée est utilisée pour concaténer les segments nécessaires à la synthèse, déterminés par la phonétisation du texte donné en entrée. Divers traitements sont alors effectués afin de lisser les jonctions entre ces segments et rendre celles-ci imperceptibles. Ensuite, afin de faire correspondre le résultat à la partition, le logiciel superVP est utilisé pour l'analyse, la transformation et la resynthèse des sons, par l'application d'algorithmes récents de haute qualité, notamment en ce qui concerne la transposition et l'étirement temporel. Enfin, certaines pistes pour ajouter de l'expressivité et avoir un rendu plus naturel ont été explorées, avec l'implémentation de certaines règles de contrôle des différents paramètres.

The internship project presented in this report is concerned by the evaluation and adaptation of existing technologies, for the purpose of singing voice synthesis, thus identifying the main problems to be solved to which some possible solutions have been suggested. For this purpose, a singing voice synthesis system has been developped, based on a unit concatenation method. This system should be able, from a score and the associated lyrics, to synthesize an audio file of a singing voice which should sound as naturel and expressive as it can possibly be. A database previously recorded and segmented is used in order to concatenate the various segments needed for the synthesis process, determined by a phonetisation step on the lyrics given as input. Various treatments are then applied in order to smooth the junctions between the segments and make them inaudible to the listener. Then, in order for the result to correspond to the score, the program superVP is used for the analysis, transformation and resynthesis of the sounds, applying recent high quality algorithms for the transposition and time-stretching. Finally, some means of making the result more natural and expressive are explored, with the implementation of a few rules allowing for the control of the various parameters.

Mots-clés : chant, synthèse concaténative, vocodeur de phase, modèle de contrôle, expressivité

Remerciements

Le sujet de ce stage, alliant des problématiques de traitement du signal avec des applications musicales, se trouve au coeur de mes intêrets, notamment en ce qui concerne les processus de transformation du signal sonore, et a donc été pour moi un sujet de prédilection. Je tiens donc avant toutes choses à remercier Axel Roebel pour m'avoir permis de faire ce stage, ainsi que pour son encadrement et la confiance qu'il m'accorde pour poursuivre ces travaux en thèse sous sa direction. Je souhaite également remercier Xavier Rodet pour son suivi, son intéret pour mes travaux, le partage de son expérience sur le sujet et ses précieux conseils qui m'ont particulièrement aidés. Je le remercie également pour la constitution préalable au stage de la base de donnée utilisée, ce qui m'a fait gagner un temps précieux. Je remercie ensuite mes camarades de bureau Antoine, Alessio, Ugo, et Guillaume pour leur compagnie pendant ces 5 mois de stage, et particulièrement Ugo et Antoine pour la relecture de ce rapport. Enfin, je remercie l'ensemble des élèves de la promotion ATIAM grâce à qui cette année passée à l'IRCAM a été particulièrement riche et agréable.

Table des matières

Introduction **9**

Contexte du stage **10**

1 État de l'art **11**
1.1 Méthodes de synthèse du chant . . . 11
1.1.1 Synthèse par modèle physique . . . 11
1.1.2 Synthèse "formantique" . . . 12
1.1.3 Synthèse par concaténation d'unités . . . 13
1.1.4 Synthèse par modèle HMM . . . 14
1.1.5 Synthèse par conversion de parole . . . 15
1.2 Modèle de contrôle . . . 15
1.2.1 Fréquence fondamentale (F_0) . . . 16
1.2.2 Alignement temporel . . . 17
1.2.3 Dynamique . . . 18

2 Approche générale **19**
2.1 Méthode retenue . . . 19
2.2 Base de données . . . 22
2.3 Outils . . . 23
2.3.1 SuperVP . . . 24
2.3.2 SHape-Invariant Phase vocoder (SHIP) . . . 24
2.3.3 Sound Description Interchange Format (SDIF) . . . 25

3 Système de synthèse **27**
3.1 Stratégie de segmentation . . . 27
3.2 Phonétisation du texte . . . 29
3.3 Concaténation . . . 30
3.3.1 Interpolation des enveloppes spectrales . . . 31
3.3.2 Correction des phases . . . 33
3.3.3 Suppression de la 1ère trame . . . 36
3.4 Transformations . . . 37
3.4.1 Transposition . . . 37
3.4.2 Étirement temporel . . . 39

4 Vers une synthèse plus naturelle et plus expressive **41**
4.1 Modèle de controle 41
4.1.1 Alignement temporel 42
4.1.2 Contrôle de la F_0 44
4.1.2.1 Transitions 45
4.1.2.2 Vibrato 45
4.1.2.3 Variations aléatoires de la F_0 46
4.1.3 Dynamique 48
4.2 Prise en compte de la source glottique 49

5 Évaluation **51**

Conclusion **53**

Introduction

Depuis l'invention par Von Kempelen de sa machine parlante il y a deux siècles [1], de nombreux scientifiques se sont attaqués au problème de la synthèse de la voix humaine. Aujourd'hui, la synthèse de la parole a atteint une qualité très satisfaisante et trouve des applications dans de nombreux systèmes de la vie courante. Plus récemment, des efforts ont été déployés pour synthétiser plus spécifiquement la voix chantée, à la fois dans un but purement scientifique pour la compréhension des phénomènes vocaux liés au chant, mais également dans un but artistique, avec l'objectif d'offrir de nouveaux outils pour la création musicale.

Malgré le fait qu'ils soient tous deux le fruit d'un même "instrument", la parole et le chant présentent certaines différences significatives aussi bien du point de vue de la production que de celui de la perception. Alors que pour la parole, l'intelligibilité du message véhiculé est le principal point d'attention, pour le chant, c'est avant tout l'intention musicale (timbre, intonation, ...) qui prime sur le contenu linguistique. En particulier, pour le chant, un contrôle précis et indépendant de la hauteur et l'intensité sur une grande plage de valeurs est nécessaire, alors que les variations prosodiques liées à la parole sont plus restreintes.

La synthèse du chant est un champ de recherche actif depuis les années 60. Depuis, de grand progrès ont été fait, et il est aujourd'hui possible de commencer à envisager des applications commerciales de qualité. Une vue d'ensemble, datant de 2002, des recherches effectuées dans ce domaine et de leurs applications est donnée dans [2]. L'article fait notamment état de deux types de systèmes : un premier type utilise en entrée une partition et applique des règles pour le contrôle expressif des paramètres ; un deuxième type de systèmes peut utiliser en entrée, au lieu d'une partition, la performance d'un chanteur réel, afin de contrôler l'expressivité de manière à donner un caractère plus naturel et humain à la synthèse [3]. À cela, on peut rajouter un troisième type de systèmes utilisant en entrée une interface permettant de contrôler la synthèse en temps-réel [4]. Dans le cadre de ce stage, nous nous intéresserons principalement au premier type, utilisant en entrée une partition et un texte.

Les applications possibles de ces recherches sont multiples. Une première possibilité est l'utilisation d'un système de synthèse pour se substituer à l'enregistrement d'un chanteur réel afin d'interpréter une partition, lorsqu'un chanteur possédant les bonnes compétences musicales ou le type de voix souhaité n'est pas disponible, ou encore si l'environnement ne permet pas un enregistrement dans de bonnes conditions. Cette application est particulièrement intéressante pour la production musicale amateur en home studio. Des techniques de synthèse sont déjà largement utilisées dans ce cadre pour un grand nombre d'instruments, mais encore assez peu pour le chant. Il serait ainsi possible d'inclure dans ses productions musicales des voix chantées sans autre besoin matériel ou humain qu'un ordinateur, offrant ainsi une solution à la fois économique et mobile. Cela pourrait aussi servir à avoir une simple idée du rendu possible, dans l'attente d'un véritable enregistrement, pour l'aide à la composition, comme le font déjà pour les instruments la plupart des logiciels d'édition de partitions actuels. Une autre application envisageable est de créer de nouvelles oeuvres avec la voix ou dans le style d'un chanteur décédé, par l'utilisation d'un système d'apprentissage des caractéristiques du chanteur sur des enregistrements originaux. Pour la création artistique, l'utilisation d'un système de synthèse du chant peut permettre de générer des perfomances avec des articulations ou variations qu'il serait impossible ou très difficile de donner à réaliser à un véritable chanteur, ou encore de créer des interpolations entre différentes voix. D'autres applications intéressantes sont certainement à trouver également dans le domaine du divertissement (cinéma, jeux vidéos, ...). L'éventualité de pouvoir contrôler en temps réel la synthèse du chant avec une interface

adaptée ouvre aussi des perspectives intéressantes pour la création contemporaine dans le domaine du spectacle vivant. Enfin, par un contrôle précis et indépendant des différents paramètres de la voix, un système de synthèse peut servir à collecter de nouvelles connaissances sur le chant et sa perception en vérifiant certaines hypothèses.

Le travail effectué durant se stage exploite les méthodes de l'état de l'art basées sur la concaténation et transformation d'unités, en intégrant des outils récents développés à l'IRCAM afin de concevoir un système de synthèse du chant de haute qualité, à partir d'un texte et d'une partition. Les motivations pour entreprendre ces travaux sont que les systèmes actuellement disponibles pour la synthèse du chant sont, bien que de plus en plus performants, encore limités dans la qualité des résultats obtenus, qui ne sont pas encore comparables à la performance d'un chanteur réel. Pour atteindre cet objectif, il y a notamment des efforts à fournir au niveau du contrôle des paramètres afin de rajouter de l'expressivité et du réalisme. Des études sont donc encore nécessaires afin de déduire les règles régissant l'évolution des différents paramètres. Enfin, la qualité du son est encore limitée par la caractérisation et la modélisation de la source glottique, qui restent à améliorer. Les résultats obtenus pendant ce stage serviront de base pour des recherches plus poussées sur la synthèse et transformation expressive de la voix chantée, notamment dans le cadre d'une thèse que j'effectuerai à l'IRCAM à partir de l'année prochaine.

Après avoir passé en revue l'ensemble des méthodes de synthèse, ainsi que certains résultats relatifs à la modélisation du chant, nous présenterons dans un premier temps une vue d'ensemble du système développé et les différents outils utilisés pour cela. Une troisième partie permettra d'aborder les différents problèmes rencontrés et d'exposer les solutions qui ont pu être apportées. Enfin, certains développements effectués dans le but de permettre une synthèse plus naturelle et plus expressive seront présentés, avant de conclure sur une discussion concernant les résultats obtenus.

Contexte du stage

Le stage présenté dans ce document s'est déroulé du 01 mars au 31 juillet 2013 sous la direction d'Axel Roebel, au sein de l'équipe "analyse et synthèse des sons" de l'IRCAM. Celle-ci poursuit des recherches portant sur l'analyse des signaux audios à des fins de transformations, de synthèse, de transcription et d'indexation. L'analyse et la transformation de la voix sont des sujets centraux étudiés dans l'équipe. Plusieurs algorithmes pour l'analyse et la transformation de la voix ont été développés dans l'équipe. D'autres sujets de recherche menées liés aux travaux effectués dans ce stage sont la conversion de la voix et la synthèse concaténative de la parole à partir du texte.

Chapitre 1

État de l'art

Depuis les premiers travaux entrepris sur la synthèse du chant dans les années 60 [2], un certain nombre de méthodes bien différentes a été proposé. Cette section a pour but de donner une vue d'ensemble des travaux réalisés à ce jour sur la synthèse de la voix chantée, qui serviront de base pour nos recherches. Dans un premier temps, les différentes méthodes de synthèse du chant existantes sont présentées, avec chacune leurs avantages, leurs défauts, et leur champ d'applications. Ensuite, certains résultats concernant la gestion des paramètres de la synthèse, indépendamment de la méthode utilisée, sont exposés.

1.1 Méthodes de synthèse du chant

Les différentes méthodes développées pour la synthèse du chant sont passées en revue dans [5] et [2]. On peut notamment distinguer parmis ces différentes techniques les modèles physiques, basés sur le mode de production du son, et les modèles spectraux, basés avant tout sur la perception du son. Alors que pour les modèles physiques, un changement de paramètre est lié à une modification au niveau du mécanisme de production vocal, un changement de paramètre pour un modèle spectral est directement lié à un changement de sensation pour l'auditeur.

Cette partie présente la synthèse par modèle physique, les synthèses de type formantiques, la synthèse par concaténation d'unité, la synthèse par modèle HMM, et la synthèse par conversion de parole, couvrant ainsi l'ensemble des techniques utilisées à ce jour pour la synthèse du chant.

1.1.1 Synthèse par modèle physique

La synthèse par modèle physique consiste à construire un modèle numérique du système réel de production de la voix, par la résolution des équations régissant le système. Les paramètres de contrôle du modèle sont alors les grandeurs physiques réelles intervenant dans la production de la voix, telles que la tension des cordes vocales, la position des différents articulateurs (machoirs, langue, larynx, ...) déterminant la forme du conduit vocal, ou encore la pression sub-glottique. Un exemple de système de ce type est le sytème SPASM (pour Singing Physical Articulatory Synthesis Model) [6]. Celui-ci utilise cependant une table d'onde afin de générer le signal de source glottique, et une source de bruit pour les phonèmes non voisés de type fricatives.

L'avantage d'un tel système est qu'il fonctionne aisément en temps-réel. Cependant, pour les modèles physiques, le mapping entre le contrôle des paramètres et la qualité de la voix perçue est complexe et peu intuitif, car, étant lié au mode de production, il n'a pas de lien direct avec la perception. Ce type de systèmes est donc particulièrement adapté pour des applications à but pédagogiques, comme l'étude du fonctionnement de l'appareil de production vocal pour les chanteurs. Cela peut par exemple donner des notions pour l'apprentissage du "placement de la voix" (en relation aux différents paramètres physiques entrant en jeu) par intéraction avec le logiciel. Le système SPASM fournit une interface graphique permettant de modifier les valeurs des paramètres de contrôle et d'observer en temps réel les conséquences sur le son produit.

En opposition à la synthèse par modèle physique, l'ensemble des techniques présentées par la suite sont avant tout basées sur une représentation spectrale de la voix.

1.1.2 Synthèse "formantique"

On désigne ici sous le terme de synthèse formantique un ensemble de techniques s'attachant à reproduire les résonnances du conduit vocal, appelés "formants". Ces méthodes sont les plus anciennes.

Un des principaux systèmes de cette catégories est le système CHANT [7, 8], développé à l'IRCAM dans les années 80. Celui-ci est basé sur une technique de synthèse par Formes d'Ondes Formantiques (FOF) qui permet de modéliser le signal comme une somme de sinusoïdes amorties générées périodiquement. La période de répétition de ces sommes de sinusoïdes permet de fixer la fréquence fondamentale, alors que la fréquence de chaque sinusoïde amortie correspond à la fréquence centrale d'un formant dont le temps d'attaque détermine la largeur de bande. Les valeurs des paramètres nécessaires à la synthèse de chaque phonème sont déterminées à l'avance à partir de l'analyse de signaux réels. Les transitions d'un phonème à l'autre sont représentées par les contours des paramètres de contrôle à travers de règles ou schémas d'interpolations.

Ce modèle est adapté à la production de sons harmoniques, et permet en particulier la synthèse de voyelles de haute qualité, mais il ne permet pas de modéliser de façon précise toutes les articulations complexes de la voix humaine. Son usage reste donc limité. Des recherches ont été menées pour synthétiser des consonnes par cette méthode [9], mais cela requiert un réglage précis et fastidieux des paramètres, et bien que beaucoup des caractéristiques de la voix peuvent être approximées de cette manière, on ne peut espérer une synthèse très naturelle et générale avec cette méthode. Ce qui fait toutefois la qualité de cette méthode est l'utilisation d'un modèle contenant un grand nombre de règles pour le contrôle des différents paramètres expressifs tels que la fréquence fondamentale. Au-delà de la voix humaine, cette méthode permet de synthétiser tout son pouvant être représenté comme une somme de résonnances, ce qui intéresse donc les compositeurs qui ne sont pas forcément intéressés par le fait de reproduire une voix naturelle qu'ils pourraient auparailleurs enregistrer eux-même avec un vrai chanteur, mais plutôt par créer des voix hybrides ou "inouïes", qui gardent un caractère vocal perceptible.

Un second système de cette catégorie est le synthétiseur MUSSE (pour Music and Singing Synthesis Equipment) [10, 8]. Il fut d'abord conçu sous forme de système électronique dans les années 70, puis a été adapté sous forme logicielle appelée MUSSE DIG dans les années 90 [11]. Le système utilise comme signal de source un signal en dents de scie riche en harmoniques. Ce signal est ensuite filtré par un ensemble de 5 circuits formantiques en cascade, de manière à modéliser les résonnances du conduit vocal. Ce système a été en grande partie utilisé dans un but d'analyse par

la synthèse de la voix chantée et de ses différentes caractéristiques. Par rapport à l'analyse directe d'enregistrements, la synthèse permet de contrôler séparément les différents paramètres et permet donc de vérifier plus aisément certaines hypothèses.

Enfin, une troisième méthode pouvant s'inscrire dans cette catégorie est la synthèse FM développée par John Chowning [8]. Le principe général de la synthèse FM, utilisé avec une fréquence porteuse multiple de la fréquence modulante, permet de créer des sons harmoniques. En utilisant plusieurs porteuses en parallèles et une seule fréquence modulante, il est possible de créer plusieurs résonnances correspondant aux formants, la fréquence modulente correspondant à la fréquence fondamentale. L'avantage de cette méthode est qu'elle permet de créer des timbres variés et riches avec peu de paramètres, et est donc peu coûteuse en calcul. Mais le modèle utilisé n'a pas de lien ni avec la perception ni avec le système de production physique de la voix, et est donc peu adapté au problème de la synthèse du chant.

1.1.3 Synthèse par concaténation d'unités

La synthèse par concaténation d'unités est une méthode éprouvée en synthèse de la parole, et est actuellement la méthode permettant d'obtenir la meilleure qualité dans ce domaine. Celle-ci consiste à concaténer des segments provenant d'une base de données pré-enregistrée. Selon le système, ces segments peuvent correspondre à des phonèmes, des diphones, ou des unités plus longues pouvant aller jusqu'à des mots ou morceaux de phrases entiers. Les systèmes de l'état de l'art utilisent ainsi des segments de tailles variables, et cherchent le meilleur compromis entre contexte et durée du segment. Le choix des segments est alors effectué grâce à des fonctions de coût, comme expliqué dans [12]. Les discontinuités au niveau des jonctions entre ces segments sont ensuite lissées, et des transformations sont appliquées au signal pour lui donner la prosodie désirée. Cette approche, en conservant un signal au plus proche du son original utilisé, permet une qualité d'élocution et de timbre naturelle très proche de la parole humaine, qui motive donc l'exploration de cette technique pour la synthèse du chant. Plusieurs systèmes de synthèse du chant ont déjà exploré les possibilités de cette méthode, avec plusieurs variantes. Les différences entre ces systèmes sont liées principalement au type de segments contenus dans la base de données et à la stratégie de sélection de ces segments, ainsi qu'aux algorithmes utilisés pour modéliser et transformer le signal.

Un premier exemple simple de ce type, est le projet Burcas [13]. Celui-ci s'appuie directement sur les résultats obtenus en synthèse de la parole, puisqu'il utilise le synthetiseur MBROLA [14], initialement prévu pour la parole, pour faire de la synthèse du chant. Dans ce système, le synthétiseur MBROLA n'est pas modifié, mais utilisé "comme une boîte noire", et utilise simplement 3 paramètres d'entrée : les phonèmes, les durées, et les hauteurs.

Un second exemple est le système Lyricos [15], qui prend en entrée une partition au format MIDI et des paroles inscrites en notation phonétique, ainsi que d'autres paramètres de contrôle musicalement intéressants tels que le vibrato. La base de données utilisée par Lyricos est annotée en phonèmes. Celle-ci est constituée à partir de 500 mots inventés (sans signification particulière) enregistrés sur deux hauteurs distinctes. Une stratégie est mise en place pour choisir les segments les plus proches de la suite de phonèmes spécifiée, les segments pouvant être de longueurs variables (un ou plusieurs phonèmes). Ces segments sont modélisés par l'utilisation d'un modèle sinusoïdal. Après l'application de divers traitements pour lisser les jonctions entre segments et transformer

le signal en fonction des paramètres d'entrée, le modèle sinusoïdal est utilisé pour synthétiser le résultat par la méthode ABS/OLA, tel que décrit dans [15].

Un 3ème système de ce type est celui décrit dans [16] et [17]. Celui-ci s'appuie pour la transformation des segments sur une extension du modèle source-filtre appelée EpR (pour Excitation Plus Resonnance) construit à partir d'un modèle sinusoïdal. Le modèle source-filtre de production de la parole est brièvement présenté en annexe. Les paramètres du modèle EpR sont extraits par l'analyse de véritables signaux de voix correspondants aux différents phonèmes ainsi que les articulations entre phonèmes, pour différentes hauteurs et pour plusieurs niveaux d'intensité. Ceux-ci sont ensuite enregistrés dans une base de données. Pour des valeurs intermédiaires, les paramètres du modèle sont interpolés. Une seconde base contient la modélisation des caratéristiques temporellement variables du chant, considérées comme indépendantes des phonèmes, telles que les attaques ou le vibrato.

Enfin, un dernier exemple est le système Vocaloïd de Yamaha [18]. Il s'agit de l'un des produits à vocation commerciale les plus aboutis du domaine à ce jour. Les bases de données utilisées par Vocaloïd contiennent l'ensemble des diphones ainsi que les parties stables des voyelles provenant de l'enregistrement de chanteurs, sur plusieurs hauteurs distinctes. Différentes bases de données existent pour le japonais et l'anglais. La concaténation et la transformation des segments sont effectuées dans le domaine fréquentiel, avec une approche du type vocodeur de phase. L'approche utilisée par ce système semble être la plus proche de celle qui a été adoptée durant ce stage.

1.1.4 Synthèse par modèle HMM

Tout comme la concaténation d'unité, la synthèse basée sur les modèles HMM a d'abord été utilisée en synthèse de la parole [19, 12] avant d'être appliquée plus récemment à la synthèse du chant. Le système présenté dans [20] pour la synthèse du chant est notamment très similaire à celui présenté dans [12] pour la parole.

La synthèse par modèle HMM est une approche statistique qui permet, à partir d'une base de données, d'apprendre les paramètres les plus appropriés en fonction du contexte et d'utiliser ces paramètres pour effectuer la synthèse. Avec cette approche, le spectre, les paramètres de la source, le vibrato et la durée des phonèmes sont modélisés simultanément par des HMMs dépendantes du contexte. Pour la synthèse, une HMM est formée en fonction des données d'entrée (texte et partition) [20]. Les paramètres issus de cette HMM sont ensuite utilisés pour reconstruire le signal de source et son enveloppe spectrale pour générer la forme d'onde. La constitution des HMMs nécessite l'utilisation d'un corpus d'apprentissage dont les performances du système sont fortement dépendantes, car des contextes qui apparaissent rarement dans les données d'apprentissage ne peuvent pas être bien appris, et il est impossible de couvrir absolument tout les contextes possibles. Ceci est particulièrement vrai pour le chant qui contient un nombre important de paramètres indépendants (paroles, intensité, durées, hauteurs, vibrato, ...). La hauteur doit en particulier être bien couverte car la trajectoire de la fréquence fondamentale a un grand impact sur la qualité de la voix perçue. L'avantage d'une telle représentation paramétrique du signal est qu'il est facile de modifier les caractéristiques du signal synthétisé. Il suffit pour cela simplement de modifier les valeurs des paramètres avant la synthèse, sans nécessiter de traitement direct sur le signal comme c'est le cas pour les méthodes par concaténation d'unités.

Un exemple de système basé sur cette méthode est le synthétiseur Sinsy [21]. Il s'agit d'un système récent (2009), accessible gratuitement en ligne à l'adresse [22]. Des exemples en ligne sur le site web du système permettent de se faire une idée de la qualité obtenue par cette méthode.

La représentation paramétrique de la voix utilisée dans cette méthode étant fortement liée à la théorie du modèle source-filtre, une des raisons évoquées pouvant expliquer les limites d'un tel système en terme de qualité obtenue est l'utilisation d'un modèle de source glottique trop peu fidèle à la réalité [12]. Les méthodes présentées dans [20] et [21] utilisent par exemple un simple train d'impulsions comme source d'excitation. L'utilisation d'un modèle de source paramétrique pourrait donc améliorer les résultats. Une autre solution, proposée dans [23], est d'utiliser des signaux réels de source glottique provenant d'enregistrements, ce qui semble apporter une amélioration significative. Malgré cela, la qualité reste plus pauvre que celle obtenue par les méthodes de concaténation d'unités. D'autres facteurs pouvant expliquer cela sont notamment l'utilisation de techniques par vocodeur trop simplistes, et un modèle statistique qui cause un moyennage trop important ("over-smoothing") des paramètres [23].

1.1.5 Synthèse par conversion de parole

Enfin, une dernière approche possible est la synthèse par conversion de parole. Cette méthode consiste à transformer un enregistrement de parole correspondant au texte à synthétiser en fonction d'une partition donnée. En plus du signal audio original, une segmentation de ce signal en phonèmes est nécessaire afin de connaître les temps nécessaires aux transformations. Pour cette raison, une telle méthode n'est pas vraiment adaptée pour développer un véritable système de synthèse, et est plutôt utilisée dans un contexte d'analyse par la synthèse, afin de tester certaines hypothèses, et étudier par exemple le rôle des différents paramètres dans l'expressivité de la synthèse, par la comparaison avec de vrais signaux de chant.

Dans [24], les auteurs utilisent la synthèse par conversion de parole pour questionner le rôle de la forme des impulsions glottiques dans l'expressivité du chant, et particulièrement au niveau du vibrato.

L'article [25] présente un autre exemple d'utilisation de cette approche. Dans cette étude, le système STRAIGHT [26] est utilisé pour transformer le signal, et trois modèles sont développés pour contrôler la fréquence fondamentale, la durée des phonèmes, et la forme spectrale.

L'ensemble des méthodes présentées ci-dessus offrent différents moyens de synthétiser des signaux de voix chantée. Mais quelle que soit la méthode utilisée, il est nécessaire de contrôler l'ensemble des paramètres de façon adéquate afin d'obtenir un résultat à la fois réaliste et musical. Pour cela, un certain nombre de recherches ont été menées, qui permettent d'établir certaines règles pour le contrôle de ces paramètres, qui vont être présentées dans la section suivante.

1.2 Modèle de contrôle

Si une synthèse de chant est réalisée en se contentant de suivre les seules informations inscrites sur la partition, sans aucune sorte de variations expressives, le résultat sonnera très mécanique. Des variations dépendantes du contexte peuvent donner plus de vie à une interprétation et la rendre ainsi plus intéressante musicalement, et surtout plus naturelle. Ces variations sont de différentes

natures. Elles peuvent par exemple modéliser la transition entre les notes, ou bien servir à ajouter des accents expressifs. Pour cela, il est nécessaire de pouvoir contrôler de manière continue un grand nombre de paramètres tels que la fréquence fondamentale, l'intensité du son, l'étendue et la vitesse du vibrato, ou la position et l'amplitude des formants [11]. Comme cela ne peut pas être réalisé manuellement, et qu'un contrôle précis est nécessaire pour les besoins de la recherche, des règles doivent être établies. Cela passe donc par l'établissement d'un modèle de contrôle, qui est chargé d'appliquer ces différentes règles afin de générer des paramètres appropriés aux besoins d'une synthèse réaliste et expressive. Des solutions existent pour le contrôle automatique de performances musicales expressives, comme le logiciel Director Musices [27], par exemple. Mais la voix présente des caractéristiques propres, différentes de la plupart des instruments, et un tel système, bien que capable de générer des performances musicales intéressantes et réalistes, ne peut pas être directement adopté pour produire une synthèse de chant acceptable, et un certain nombre de règles spécialement établies pour le chant sont requises. Évidemment, il n'y a pas de règles universelles correspondant à tout les styles de chant, et le style doit donc être pris en considération lors de l'établissement de ces règles. Les considérations de style en relation au contrôle des paramètres de la synthèse ont jusqu'à lors été assez peu étudiées, et de nombreuses recherches sont encore nécessaires. Mais un certains nombre de résultats intéressants ont néanmoins déjà pu être établis, procurant ainsi certaines bases pour la conception d'un système de synthèse réaliste et expressif.

Dans [11] et [28], les auteurs présentent le système RULSYS (développé pour une utilisation avec le système de synthèse MUSSE) et les règles qui ont été établies grâce à l'utilisation de ce système dans une démarche d'analyse par la synthèse. Dans cette approche, des performances de chant sont synthétisées, les résultats obtenus sont analysés, et les règles contrôlant la synthèse sont ajustées en fonction de la qualité de ces résultats.

La thèse de Céline Chabot-Canet, soutenue en juin 2013, présente les résultats obtenus par de nombreuses analyses effectuées sur des enregistrement de personnalités de la chanson française de 1950 à nos jours, et constitue une source d'informations intéressante.

Nous nous contenterons ici de couvrir les trois paramètres expressifs qui semblent vraiment indispensables, car directement perceptibles et mesurables, qui sont la fréquence fondamentale, la dynamique (ou intensité) du son, et l'alignement temporel, qui concerne le placement et la durée des différents phonèmes prononcés.

1.2.1 Fréquence fondamentale (F_0)

La fréquence fondamentale est de loin le paramètre le plus important à prendre en compte pour la synthèse de la voix chantée, car c'est elle, plus que l'aspect rythmique ou n'importe quel autre paramètre, qui détermine l'identité d'une mélodie et nous permet de reconnaître une chanson. Au-delà de simplement porter la mélodie, les variations de la fréquence fondamentale sont également un vecteur primordial d'expressivité.

Dans [29], l'auteur étudie les différences de variations de F_0 entre deux voix perçues comme étant de bonne ou de mauvaise qualité. Plutôt que le fait de simplement chanter faux, comme on aurait pu s'y attendre, il semblerait que ce qui donne à la seconde voix sa mauvaise qualité est dû en particulier à un vibrato très irrégulier, qui confère à la voix un caractère instable.

Le vibrato correspond à une modulation quasi-sinusoïdale de la fréquence fondamentale, à une fréquence généralement comprise entre 4 et 7 Hz et d'une amplitude généralement comprise entre

2% et 5% de la fréquence fondamentale ($\leq$ 85cents) [30]. Sundberg suggère dans que l'amplitude du vibrato augmente avec l'amplitude du son. Il apparaît que le vibrato est un paramètre particulièrement important, intrinsèque à la voix chantée naturelle, et ce pour de nombreux styles de chant. Dans [7], l'auteur précise que le vibrato est important pour la reconnaissance de l'identité d'un chanteur, car les fréquences des différents partiels "balayent" les formants, révélant donc mieux le timbre de la voix à l'auditeur qu'un son plat. Cela est particulièrement vrai pour le registre aigu où l'enveloppe spectrale est sous-échantillonée. L'étude présentée dans [30] s'intéresse également aux variations du spectre en lien avec le vibrato et souligne l'existance d'une modulation d'amplitude synchrone à celui-ci, dû notamment au balayage des formants par les harmoniques. De plus, d'après [31], le vibrato est plus constant chez les professionnels que chez les amateurs, ce qui confirme donc les résultats de l'étude [29].

Dans [30], l'étude indique que l'inclusion de variations aléatoires de la fréquence et l'amplitude du vibrato, ainsi que les variations aléatoires de la F_0 n'apportent pas de préférence sur la qualité de la voix obtenue comparativement à des exemples utilisant des valeurs fixes. Cela semble être en contradiction avec de précédents rapports stipulant que l'inclusion de variations aléatoires appropriées dans la synthèse du chant est nécessaire pour éviter un son mécanique et peu naturel [32]. L'article [7] souligne également l'importance des variations aléatoires de l'ordre de +/-0.5% de la F_0 pour donner un aspect naturel à la voix. Néanmoins, certains styles comme le rock présentent peu de vibrato, et ce seul attribut ne suffit donc pas à conférer à la voix un caractère naturel.

D'autres recherches récentes ont également étudié les différents types de variations de la fréquence fondamentale dans le chant [25, 33, 31]. Dans ces études, 4 types de fluctuations différentes sont identifiées, désignées sous les noms d'*overshoot*, *preparation*, *vibrato*, et *fine-fluctuations*. L'*overshoot* désigne une déflection excédant la hauteur cible après un changement de note. La *preparation* correspond à une déflection dans la direction opposée au changement de hauteur observée juste avant un changement de note. Le terme *fine-fluctuations* désigne une variation irrégulière de la F_0 à des fréquences comprises entre 10 et 20Hz, d'une amplitude moyenne de 20 cents (1.2%) et maximum de 100 cents. Les *overshoot*, *preparation* et *vibrato* sont modélisés dans cette étude par une fonction de transfert d'un système du second ordre, mais cette modélisation n'est pas très intuitive (surtout pour le *vibrato*). Les *fine-fluctuations*, elles, sont générés par filtrage passe-haut de bruit blanc puis normalisation de l'amplitude. Dans [31], les auteurs affirment que l'*overshoot* est le type de variations qui a la plus grande influence sur la qualité perçue du résultat, mais que les 4 sortes de fluctuations de la F_0 sont importantes. Mais le système a été évalué sur des comptines japonaise uniquement, et l'on peut donc se demander si les types de fluctuations de F_0 identifiées ici sont propres à ce style de chant ou revètent un caractère plus général.

Dans [28], Sundberg a étudié le type particulier de trajectoire suivi par la F_0 propre à la technique de chant "coloratura" (succession rapide de notes courtes chantées uniquement sur des voyelles).

1.2.2 Alignement temporel

Le rythme est un autre aspect musicalement déterminant représenté directement dans la partition. Celui-ci détermine de façon précise les temps de début et fin de chaque note de la mélodie. Pour la plupart des instruments, cela reste donc assez simple, puisqu'au delà de l'aspect temporel, une note est simplement déterminée par sa hauteur et son amplitude, le timbre étant directement lié à l'instrument en question. Mais pour le chant, les paroles associées à la mélodie apportent une variabilité supplémentaire qu'il faut donc prendre en compte d'un point de vue temporel. Les

termes “alignement temporel” recouvrent ici à la fois l’alignement du début de chaque note avec les phonèmes, et la durée de chaque phonème (sachant qu’à un seul phonème peut correspondre plusieurs notes, et inversement, à une note correspondent souvent plusieurs phonèmes).

Plusieurs études ont abordé cette question, comme [29] qui décrit certaines règles relatives à l’alignement temporel. L’article suggère que les notes doivent être alignées sur les “onsets” des voyelles, les éventuelles consonnes la précédent faisant donc parties de la note précédente. La transition entre deux notes se fait alors avant le début de la voyelle qui commence donc directement sur la hauteur cible de la seconde note.

Dans [25], la durée de chaque phonème est déterminé en fonction de la note associée (type de note et tempo). Ce système considère que chaque enchainement entre consonne et voyelle peut se découper en une partie “consonne”, une partie “transition” et une partie “voyelle”. La partie “transition” se situe entre 10ms avant et 30ms après la frontière entre la consonne et la voyelle. La partie consonne est allongée d’un facteur fixe déterminé à l’avance, la partie transition ne bouge pas, la partie voyelle est allongée de façon à ce que le tout atteigne bien la durée de la note souhaitée. Ce modèle sous-tend que les consonnes ont chacune une durée “naturelle” déterminée (différente en chant qu’en parole) indépendante de la durée de la note. Mais aucune précision n’est apportée sur une solution à adopter dans le cas d’un tempo rapide, pour lequel cette stratégie ne serait plus tout à fait adaptée. Un modèle est donc nécessaire qui doit permettre à la fois l’utilisation de note tenues anormalement longue pour la parole, ou au contraire de notes très courtes, avec des articulations particulièrement rapides.

1.2.3 Dynamique

Un troisième facteur expressif important pour toute performance musicale, quel que soit l’instrument, est l’intensité du son. On désigne par le terme “dynamique” les évolutions de l’intensité du son perçue au cours du temps, qui peuvent être directement liées à des indications de nuances sur la partition, mais peuvent aussi se situer à un niveau plus fin, au niveau des transition entre les différentes notes ou les différents phonèmes par exemple.

Cet aspect est assez peu représenté dans les études présentées précédemment, mais un constat qui est soulevé par de nombreux auteurs est l’existance d’une dépendance entre l’intensité du son et son timbre [8, 28, 7]. On peut en effet remarquer qu’une voix de forte intensité est plus riche en harmoniques aigues qu’une voix de faible intensité. Il s’agit d’un effet non linéaire, et on ne peut donc pas se contenter, pour gérer la dynamique du chant, de simplement modifier l’amplitude du signal. Cela aurait perceptivement le même effet qu’entendre une personne chanter plus ou moins près du micro, mais avec la même intensité. Pour le chant, on préfèrera donc parfois le terme “effort vocal” au terme “intensité”, qui rend mieux compte de cet effet. Une solution souvent adoptée pour prendre en compte cet effet est de modifier la pente spectrale du signal de source en fonction de l’intensité. Cette solution est particulièrement facile à mettre en place pour les systèmes de synthèses paramétriques basés sur un modèle source-filtre de production de la voix, mais est moins aisée pour des systèmes de type concaténation d’unités. La raison de cette modification de la pente spectrale est due à un changement de la forme des impulsions de source glottique avec l’intensité. Une modélisation paramétrique de la source comme le modèle LF [34] peut permettre de contrôler cet effet avec plus de précision. Des investigations à ce sujet sont notamment menées dans [24].

Un autre aspect lié à la dynamique est la possible présence d’une modulation d’amplitude synchrone au vibrato, de type tremolo, comme soulevé dans [25].

Chapitre 2

Approche générale

Dans cette partie, l'approche générale adoptée pour la conception d'un système de chant est d'abord présentée, afin de donner une première vu d'ensemble. Les principaux outils et algorithmes sur lesquels s'appuie le système sont ensuite également présentés, afin de permettre une meilleure compréhension du système et des travaux effectués durant ce stage.

2.1 Méthode retenue

La méthode qu'il a été choisi d'explorer dans le cadre de ce stage est la synthèse par concaténation et transformation d'unités. Comme évoqué dans la première partie, cette méthode, en conservant un signal proche de l'enregistrement d'un chanteur réel, permet de générer une synthèse naturelle, qui donne actuellement les meilleurs résultats en synthèse de la parole, et semble donc prometteuse pour la synthèse du chant. Comme expliqué précédemment, cette méthode consiste à sélectionner dans une base de données les segments nécessaires à la synthèse, puis à les concaténer tout en essayant de supprimer au maximum les discontinuités audibles, et enfin à transformer le résultat de cette concaténation en hauteur et en durée pour le faire correspondre à la partition.

La figure 2.1 présente une vue générale du système établi. Celui-ci prend en entrée un texte, une partition, et une base de données préalablement enregistrée et annotée.

Le système peut être découpé en 4 parties :

- le modèle de contrôle, qui permet, en fonction de la partition, de décrire l'évolution des différents paramètres nécessaires à la synthèse
- la sélection des segments nécessaires à la synthèse
- le module de concaténation
- le module effectuant les transformations et la resynthèse

Entrées du système :

Les entrées comprennent un texte, une partition, des paramètres expressifs, et une base de donnée. Le texte peut être écrit en français ou en notation phonétique. La partition se présente pour le moment sous forme d'un tableau contenant la durée de chaque note en terme de nombre de noires, le tempo étant défini par une autre variable, et d'un tableau contenant pour chaque note un numéro midi définissant sa hauteur. La conversion de hauteur midi en fréquence exprimée en Hz se

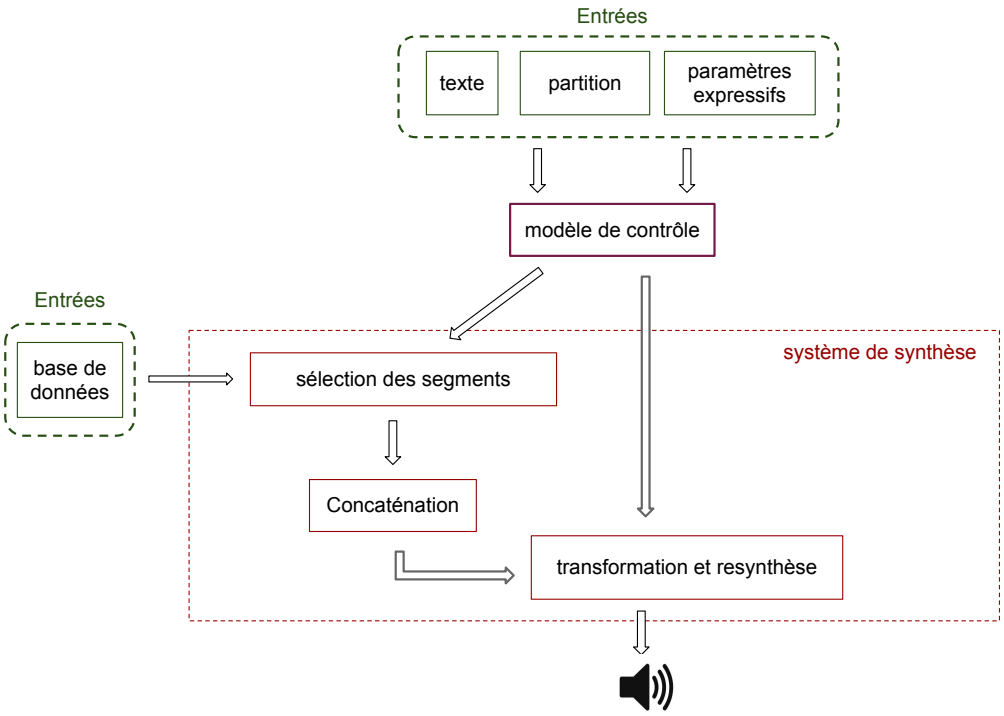


FIGURE 2.1 – Vue d'ensemble de l'approche utilisée pour la synthèse

fait par la formule suivante :

$$f = 2^{(p-69)/12} \times 440 \tag{2.1}$$

où p correspond au numéro midi de la note et f à la fréquence en Hz. Les paramètres expressifs peuvent comprendre par exemple la fréquence, l'amplitude, et le temps d'attaque du vibrato, ou le portamento (qui désigne la durée de la transition entre 2 notes). Il n'est pour l'instant possible que d'effectuer un réglage général de ces paramètres, mais il serait utile par la suite de permettre un éventuel réglage indépendant pour chaque note, afin d'augmenter la flexibilité du système et de donner un rendu plus vivant. La base de donnée contient l'audio correspondant à l'ensemble des contextes phonétiques du français et des marqueurs labellisés. Plus de détails sur la constitution de cette base sont donnés dans la section suivante.

Modèle de contrôle :

Le modèle de contrôle permet, à partir des différentes entrées du système, de déterminer précisemment les paramètres nécessaires à la synthèse. Ceux-ci concernent notamment l'alignement temporel des phonèmes, l'évolution de la fréquence fondamentale, et l'évolution de la dynamique. Par manque de temps, l'aspect dynamique n'a pas été abordé dans ce stage. Les sorties du modèle de contrôle développé sont donc les temps de début (et donc la durée) de chaque phonème et une courbe décrivant l'évolution précise et continue de la fréquence fondamentale (échantillonnée à 1000Hz) sur toute la durée du signal à synthétiser. Ces valeurs sont déterminées à l'aide de règles pré-établies. Mais le modèle de contrôle étant tout à fait indépendant de la méthode de synthèse utilisée, il serait tout à fait possible d'utiliser des valeurs provenant d'une autre source, comme de l'analyse d'un enregistrement de chant réel, par exemple.

Sélection des segments :

La sélection des segments se base sur la suite de phonèmes à synthétiser pour générer une suite de labels correspondant aux segments de la base qui seront nécessaires à la synthèse de ce texte. Mais, en fonction de la durée de chaque phonème déterminée par le module de contrôle, les segments ne seront pas nécessairement sélectionnés dans leur totalité. Les informations issues du module de contrôle sont donc utilisées afin d'ajuster les temps des segments à sélectionner dans la base. Contrairement à ce qui est pratiqué généralement en synthèse de la parole [12], aucune fonction de coût n'est ici nécessaire pour la sélection des segment avec la base actuelle, car une seule occurence de chaque segment y est représentée. Bien entendu, cette base est un outil de travail pour le développement d'un premier système et pourra être amenée à évoluer par la suite. Celle-ci pourrait par exemple contenir des enregistrements sur plusieurs hauteurs et niveaux d'intensitées, et différents types de voix.

Concaténation :

La concaténation des différents segments est effectuée dans le domaine temps-fréquence, grâce à une analyse basée sur le vocodeur de phase, avec le logiciel superVP. Pour cela, le signal est fenêtré sous forme de trames successives et superVP produit un fichier au format SDIF [35] contenant une analyse pour chacune des trames du segment. Cette analyse contient notamment le résultat d'une Transformée de Fourier Discrète (TFD) pour une représentation de type vocodeur de phase, mais aussi d'autres informations nécessaires aux algorithmes de transformation et synthèse, comme un modèle sinusoïdal, par exemple. Une telle analyse est effectuée pour chaque segment sélectionné

dans la base, et son résultat est copié dans un nouveau fichier SDIF, à la suite des trames du segment précédent, tout en corrigeant les temps et les phases de chaque trame afin d'établir une cohérence dans la succession de ces dernières dans le fichier de sortie.

En plus de cette analyse temps-fréquence, d'autres informations sont nécessaires pour la transformation du signal. Des analyses de la fréquence fondamentale et de l'enveloppe spectrale sont ainsi effectuées également par superVP et concaténées de la même manière dans des fichiers séparés.

Transformation - resynthèse :

Comme expliqué précédemment, le processus de concaténation crée un fichier au format SDIF contenant toutes les informations nécessaires pour pouvoir resynthétiser le signal. Mais pour que le résultat corresponde à celui attendu, il est nécessaire de tranformer le signal. SuperVP effectue ces transformations en même temps que la resynthèse. Il prend donc pour cela en entrée le fichier SDIF concaténé, lui applique les transformations nécessaires, et produit en sortie directement un fichier audio. Les transformations nécessaires sont notamment une transposition et un étirement/compression temporel variables au court du temps. A cela s'ajoute également, comme nous le verrons en 3.3.1, une opération de filtrage par l'enveloppe spectrale. Comme cette opération de filtrage ne peut pas être effectuée en même temps que la transposition, dû à certaines contraintes internes à superVP, la resynthèse s'effectue en deux phases. Dans un premier temps, le son est transposé suivant la courbe déterminée par le module de contrôle. Ensuite, le résultat est en même temps étiré ou compressé en différents points, suivant des ratios prédéterminés, et filtré par l'enveloppe spectrale appropriée afin d'obtenir le résultat final. Les données issues du modèle de contrôle sont utilisées pour contrôler les paramètres de ces transformations.

2.2 Base de données

Avant le début du stage, une base de donnée pour la synthèse du chant avait déjà été constituée par Xavier Rodet, et a été utilisée durant ce stage pour le développement et l'évaluation d'un premier système de synthèse fonctionnel. Cette base a été enregistrée avec la contribution d'un chanteur professionnel ayant une voix de type ténor.

Pour les besoins de la synthèse, la base doit couvrir l'ensemble des possibilités d'enchaînements entre différents phonèmes (en se limitant seulement aux diphones, c'est à dire aux enchaînements de deux phonèmes). Il est à noter que la base constituée est destinée pour le moment uniquement à la langue française, et ne couvre donc que les articulations propres à cette langue.

Pour cela, la stratégie d'établissement de la base a été la suivante :

- Dans un premier temps, le chanteur chante les combinaisons de type CVCVC, où C représente une consonne et V une voyelle. Pour une voyelle donnée, l'ensemble des consonnes est parcouru, avant de passer à la voyelle suivante. Cela donne un texte du type "babab, dadad, vavav, ...".
- Ensuite, les combinaisons du type V1V2V1, où V1 et V2 représentent 2 voyelles, sont parcourues de la même manière. Cela donne donc par exemple des enchainement de type "aoa, ioi, eoe, ..."
- Enfin, la dernière série couvre les enchaînements du type VC1C2V, ou V correspond au son [e] de l'Alphabet Phonétique International (API), et C1 et C2 sont deux consonnes. Le chanteur couvre l'ensemble des consonnes C1 en gardant C2 fixe avant de passer à la consonne C2 suivante. Cela donne la suite "ebme, edme, evme, ...".

La dernière série n'a été enregistrée que pour la voyelle [e], mais cela n'a pas d'importance car seule la transition entre les deux consonnes nous intéresse ici, et l'on peut supposer que la voyelle utilisée n'a pas d'influence sur la prononciation des consonnes. La voyelle sert donc ici uniquement à ajouter un certain contexte afin de permettre de prononcer l'enchainement de consonnes dans un contexte de chant.

Afin de conserver une certaine homogénéité sur toute la base, il aurait été souhaitable que le chanteur reste tout le temps sur la même note. Cependant, ce point a mal été contrôlé lors de l'enregistrement, et la hauteur du son dévie beaucoup entre le début et la fin de la base, avec une fréquence fondamentale allant de 100-110Hz à environ 150Hz, soit une déviation d'environ +50% de la hauteur initiale. Cela n'est pas critique pour une première évaluation du système développé, mais il serait souhaitable d'avoir à l'avenir une base plus homogène pour obtenir une synthèse de qualité. Une justification de cela est que l'algorithme de transposition utilisé, bien que de haute qualité, possède certaines limites au-delà desquelles certains artefacts génants pourraient apparaître. Afin de pouvoir proposer un système de synthèse flexible offrant une grand marge de manoeuvre, tout en conservant une qualité optimale, il pourrait donc s'avérer utile d'enregistrer la base sur plusieurs hauteurs fixes afin d'utiliser la plus proche de la hauteur cible souhaitée et ainsi limiter les ratios de transposition à appliquer, et donc limiter les artefacts liés à la transposition. Cependant, il est à noter que les limites en terme de qualité obtenue relativement aux ratios de transposition utilisés n'ont pas été évaluées. De plus, la hauteur du son est reliée à d'autres paramètres tels que la forme de la source glottique. Ces considérations n'ont pas été prises en compte durant ce stage, mais pourrait l'être par la suite pour proposer une synthèse plus naturelle. Il serait donc également souhaitable de garder une hauteur constante sur la base afin de garder ces autres paramètres constants et en faciliter l'analyse.

La stratégie décrite ci-dessus permet de couvrir avec certitude l'ensemble des segments nécessaires afin de synthétiser toutes les articulations possibles de la langue française. La base de donnée ainsi constitué possède une certaine redondance, mais un tri sera effectué au moment de la segmentation pour ne conserver qu'une seule occurence de chaque segment. La base de données utilisée durant ce stage totalise environ 1h30 de son. Mais une fois la segmentation effectuée, il serait envisageable, si nécessaire, d'éliminer l'ensemble des sons situés en-dehors des segments pour ne conserver que les portions de son utiles pour la synthèse, et ainsi réduire de façon assez conséquente la taille de la base.

Les instructions du chanteur lors de l'enregistrement étaient d'éviter les variations de type vibrato. Toutefois, il semble que cela soit un exercice difficile, et une certaine dose de vibrato est tout de même présente dans l'enregistrement. Mais nous verrons que cela a finalement assez peu d'importance avec l'implémentation actuelle du système.

2.3 Outils

La plupart des briques nécessaires au développement d'un premier système fonctionnel de synthèse du chant sont déjà existantes, et une grande partie du travail effectué pendant ce stage a consisté à assembler et adapter ces différentes briques élémentaires afin de construire un système complet. Dans cette section sont donc présentés les principaux outils sur lesquels s'appuient le système développé durant ce stage. Le but de ce stage était notamment d'étudier les limites de ces outils pré-existants et d'identifier, en fonction des problème rencontrés et des résultats obtenus, les évolutions nécessaires pour permettre une synthèse de meilleure qualité.

2.3.1 SuperVP

SuperVP est un logiciel pour le traitement, la transformation et l'analyse des sons en temps différé et en temps réel, développé au sein de l'équipe analyse et synthèse des sons de l'IRCAM. SuperVP est basé sur une implémentation du vocodeur de phase ([36], chapitre 7) étendue utilisant en interne une représentation temps fréquence du son. Il permet un ensemble important d'effets de transformation du son de haute qualité. Certaines des transformations accessibles sont par exemple la dilatation/compression temporelle avec préservation des transitoires, la transposition avec préservation du timbre, le filtrage, la synthèse croisé, le traitement séparé des composantes de source et filtre du son, etc. Un grand nombre de paramètres sont accessibles, permettant ainsi le contrôle précis des algorithmes et résultats obtenus. En plus des transformations, superVP permet aussi un nombre important d'analyses, comme le calcul du spectrogramme, l'estimation de l'enveloppe spectrale ou de la fréquence fondamentale, et la détection des transitoires. De plus, l'ensemble des fonctionnalités de superVP sont accessibles en ligne de commande, ce qui est particulièrement avantageux pour son utilisation par d'autres systèmes, comme dans le cas de ce stage. SuperVP est notamment utilisé comme moteur d'analyse/synthèse par le logiciel audiosculpt.

L'objectif de ce stage était de développer un système utilisant des algorithmes récents de haute qualité pour la transformation. Un certains nombre d'algorithmes d'analyse et transformation des sons, et particulièrement de la voix, nécessaires pour la synthèse du chant, sont déjà intégrés au sein de superVP. Il n'a donc pas été nécessaire de réimplémenter ces algorithmes, et superVP a été utilisé pour l'analyse, la transformation et la synthèse dans le système développé tout au long de ce stage. L'interface en ligne de commande permet de spécifier un fichier d'entrée et un fichier de sortie ainsi que l'analyse ou la/les transformation(s) à effectuer, en sélectionnant les algorithmes et paramètres à utiliser. Les divers algorithmes de transformation implémentés peuvent utiliser des fichiers de paramètres pour spécifier les paramètres de la transformation. Ces fichiers prennent la forme de "break-point functions", stockées dans des fichiers texte, qui à des temps déterminés associent une valeur du paramètre à contrôler. Cela permet par exemple de spécifier l'évolution au cours du temps du facteur de transposition à utiliser pour atteindre une courbe de F_0 cible. Entre les temps définis dans ces fichiers de paramètres, les valeurs sont interpolées linéairement. Les résultat des analyses effectuées sont stockées dans des fichiers au format SDIF, qui peuvent également être utilisés pour paramétrer certains algorithmes, lors d'une autre analyse ou transformation, et pour la resynthèse après une éventuelle modification des données stockées. Les commandes d'appels de superVP utilisés dans notre système sont présentées en annexe.

2.3.2 SHape-Invariant Phase vocoder (SHIP)

Dans le système proposé, il est nécessaire, pour que le résultat de la synthèse corresponde à la partition donnée en entrée, de modifier le son original en hauteur et en durée. Pour effectuer ces transformations, superVP permet de manipuler le signal dans le domaine temps-fréquence, en se basant sur l'architecture du vocodeur de phase. Le vocodeur de phase consiste à effectuer une Transformée de Fourier à Court Terme (TFCT), puis à resynthétiser le signal par un processus d'addition-recouvrement (ou "overlap-add" en anglais) après l'application d'une transformée inverse sur chacune des trames. Pour appliquer des transformations, il est nécessaire, avant la resynthèse, de manipuler les trames tout en corrigeant les phases afin de conserver une cohérence d'une trame à l'autre. Afin de calculer les phases nécessaires à la synthèse, les fréquences exactes correspondant à chaque point de la DFT sont calculées en fonction de l'évolution des phases dans le signal original. (Pour une explication plus détaillée de l'algorithme du vocodeur de phase, se référer par exemple

à [36]). Mais dans l'implémentation classique du vocodeur de phase, les phases de chaque point frequentiel de la Transformée de Fourier Discrete (TFD) sont adaptées indépendamment les unes des autres. A cause d'erreurs d'estimation de fréquence, cela mène à une désynchronisation inter-partiels (entre les différentes sinusoïdes composants le son) et intra-partiels (entre les différents points fréquentiels correspondants à une même sinusoïde) des phases donnant lieu à un artefact connu en anglais sous le nom de "phasiness", qui se caractérise par un son diffus, un manque de netteté. Certaines améliorations, telles que celles présentées dans [37] et [38] permettent d'éviter la désynchronisation intra-partiel. Le vocodeur de phase, avec ces améliorations, permet des transformations de haute qualité pour la plupart des sons musicaux.

Mais dans le cas de la voix, en raison de son mode de production particulier basé sur le modèle source-filtre, il est particulièrement important de conserver la synchronisation de la phase entre les différentes harmoniques (synchronisation inter-partiels), car une desynchronisation de ces composantes affecte la perception des impulsions de la source glottique. Pour palier à ce problème, l'algorithme SHIP [39], implémenté dans superVP, propose une méthode qui permet, tout en se basant sur le vocodeur de phase, de conserver l'alignement vertical des phases des différentes harmoniques. La forme d'onde originale des impulsions glottiques est ainsi conservée, et, pour les signaux de voix, le résultat des transformation est donc perçu comme étant plus net qu'avec un vocodeur de phase classique (incluant les améliorations précédemment évoquées). Il s'agit d'un algorithme de type SOLA (Synchronous OverLapp-Add) [40] effectué dans le domaine fréquentiel. Au lieu d'adapter indépendamment les phases de chaque partiel, ce qui cause leur désynchronisation, le point d'intercorrélation maximum entre deux trames successives est recherché, et les phases sont ensuite modifiées de la même manière pour l'ensemble des partiels afin de faire correspondre le point de jonction des deux trames au point de maximum d'intercorrelation, sans déplacer la position des trames comme c'est d'habitude effectué pour l'algorithme SOLA dans le domaine temporel. Les phases étant toutes ajustées de la même manière, "en bloc", cette méthode permet donc de conserver l'alignement vertical des phases entre les différentes harmoniques, et ainsi de conserver la forme d'onde originale.

L'un des avantages de cet algorithme, par rapport un simple algorithme SOLA dans le domaine temporel, est la distinction entre les composantes sinusoïdales et partie bruitée qui est faite lors du calcul de l'intercorrélation. Cela mène à une meilleure qualité, car seules les composantes sinusoïdales sont utilisées dans le calcul de l'intercorrelation, afin d'éviter l'impact du bruit. Un second avantage important est qu'un tel algorithme permet également des transformations avancées du son dans le domaine fréquentiel qui ne peuvent être effectuées dans le domaine temporel.

2.3.3 Sound Description Interchange Format (SDIF)

Afin de permettre le stockage et la manipulation des données issues des différentes analyses, superVP écrit celles-ci dans des fichiers au format SDIF. Le format SDIF, pour "Sound Description Interchange Format" est un standard établi conjointement par l'IRCAM, le CNMAT de l'université de Berkeley, et le Music Technology Group de l'université Pompeu Fabra de Barcelone [35]. Ce format permet de stocker une grande variété de descriptions du son pour l'analyse et la synthèse, comme des modèles spectraux, sinusoïdaux, et temporels, ou des descripteurs sonores tels que la sonie et la fréquence fondamentale, ou bien encore de simples marqueurs temporels. La segmentation de la base de donnée est par exemple stockée sous forme d'un fichier sdif de marqueurs labelisés. Le format SDIF sonsiste en une structure de données et un ensemble extensible de descriptions standards du son.

Un fichier SDIF se présente sous forme d'une succession de trames. Celles-ci se distinguent par leur position temporelle et leur type. Elles doivent être arrangées de telle manière que le temps d'une trame à l'autre ne soit jamais décroissant. Pour chaque trame, les données sont stockées dans une ou plusieurs matrices de tailles variables, et dont le type dépend des données à stocker.

La bibliothèque "easySDIF" permet de manipuler ces fichiers dans le langage python, utilisé pour les divers déloppements effectués durant ce stage. Les types standards de l'IRCAM sont connus par celle-ci. Mais il est également possible de définir de nouveaux types. Pour cela, il suffit d'ajouter leur définition dans l'entête du fichier.

Un exemple concret de fichier SDIF est donné en annexe. Toutes les informations sur la description du standard, du format des fichiers, et des principaux types utilisés sont accessibles à l'URL [35].

Chapitre 3

Système de synthèse

3.1 Stratégie de segmentation

Comme, nous l'avons évoqué dans la partie 1, une différence notable entre les différents systèmes de synthèse par concaténation d'unités sont les bases de données utilisées. Une première base de donnée ayant déjà été constituée avant le début du stage, des investigations ont tout d'abord été menées afin de déterminer la stratégie de segmentation la plus apropriée pour définir les segments à utiliser pour la synthèse. Plusieurs possibilités ont été envisagées :

- une segmentation en phonèmes
- une segmentation en diphones longs en positionnant les marqueurs au milieu de chaque phonèmes
- une segmentation en diphones courts et parties stables pour les voyelles, comme effectuée dans [18]

Une simple segmentation en phonème à l'avantage de permettre une segmentation semi-automatique, en utilisant le programme ircamAlign pour repérer les différent phonèmes dans la base, ce qui aurait pu être un gain de temps sur une segmentation complètement manuelle. Celle-ci a également l'avantage de permettre un contrôle précis et direct de la durée de chaque phonème ainsi que de sa position dans le temps. Mais cette solution a dans un premier temps été écartée, car elle ne permet pas de modéliser les transitions entre les différents phonèmes marquées par la coarticulation (transition entre deux phonèmes, avec un mouvement continu des formants), et ne semble donc pas adaptée pour obtenir une élocution naturelle. Pour cette raison, une segmentation en diphones semble préférable, puisqu'elle permet de bien conserver les caractéristiques de la coarticulation pour chaque couple de phonèmes. La $3^{\text{ème}}$ solution a l'avantage de bien séparer les parties transitoires des voyelles, dues au phénomène de coarticulation, de leur partie stable, ce que ne permet par la seconde méthode, et a donc été retenue. Cela offre plus de souplesse pour le contrôle, car la partie transitoire et la partie stable d'une voyelle peuvent ainsi être traitées séparément lors de la gestion de l'alignement temporel. C'est notamment ce qui est fait dans [25] pour le modèle de contrôle de durée, où la partie transitoire n'est pas modifiée, alors que la partie stable est, elle, étirée de manière à donner la longueur souhaitée à la voyelle.

Pour effectuer cette segmentation de façon manuelle, une analyse de l'enveloppe spectrale de toute la base, par l'algorithme "true-envelope", a été effectuée. L'enveloppe spectrale est ensuite

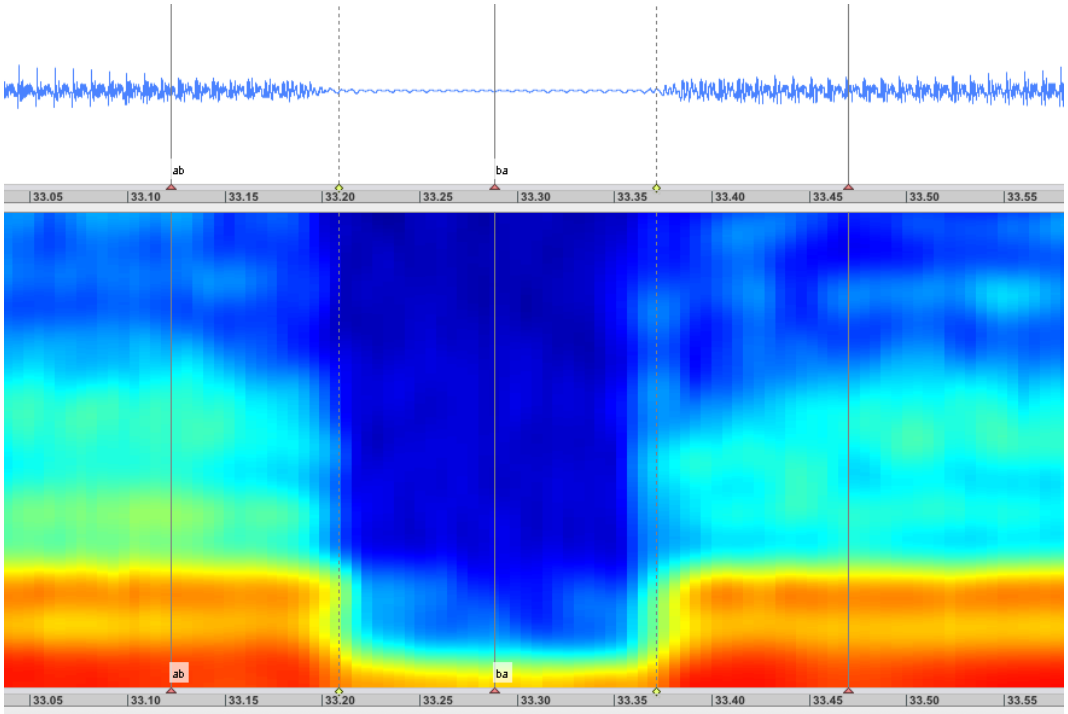


FIGURE 3.1 – Exemple de segmentation pour les diphones “ab” et “ba”

affichée dans le logiciel audiosculpt, permettant ainsi de repérer facilement les positions des transitions entre phonèmes correspondant à des mouvements importants des formants. La visualisation de l'enveloppe, conjuguée à une écoute attentive, permet d'effectuer une segmentation précise et efficace. Pour chaque diphone, les marqueurs sont placés, pour les voyelles, après la transition, à l'endroit où l'enveloppe spectrale redevient stable (et à partir duquel on n'entend donc plus de variation notable du timbre). Ainsi, les diphones sont gardés au plus courts et ne contiennent donc, pour les voyelles, que la partie transitoire. En ce qui concerne les consonnes, on se contente de placer le marqueur au milieu car, contrairement au voyelles, les consonnes ne sont jamais tenues, et il n'est donc pas nécessaire de déterminer une partie stable. La figure 3.1 donne un exemple de segmentation pour les diphones “ab” et “ba”.

La segmentation de la base est effectuée manuellement dans audiosculpt et les marqueurs sont enregistrés dans un fichier au format SDIF. Pour chaque segment, le marqueur de début du segment comporte un label identifiant le diphone ou la voyelle correspondante, écrite en notation phonétique avec l'alphabet SAMPA (voir annexe). Le prochain marqueur (en trait plein) présent dans la base, après ce dernier, marque la fin du segment. Celui-ci peut être également le marqueur de début du diphone suivant, comme c'est le cas dans l'exemple de la figure 3.1 pour le marqueur central, en quel cas il comporte le label du diphone suivant, mais il peut également s'agir d'un simple marqueur sans label, comme le dernier marqueur de cette même figure.

Comme expliqué précédemment, la base de données constituée présente une certaine redondance. Ainsi, le segment correspondant au son “gagag” contient deux fois le diphone “ga”. Une

seule occurence du diphone est alors annotée. De même, les segments “aoa” et “oao”, tout deux présents dans la base à des temps différents, contiennent tout deux les diphones “ao” et “oa”. Dans ce cas, un même diphone risque probablement d'être annoté plusieurs fois, mais cela n'a pas d'importance, car lors de la sélection des segments, seule la première occurence rencontrée est utilisée. Cette simplification est effectuée ici car on fait l'hypothèse que dans la base utilisée, les différentes occurences pour un même segment sont très similaires. On pourrait cependant imaginer d'utiliser par la suite une base de données incluant plus de variations, dont les différentes occurence pour un même segment présenteraient des caractéristiques différentes. Il faudrait alors choisir le segment le plus adapté en fonction du contexte.

La stratégie de segmentation ainsi établie permet une élocution naturelle au niveau des transitions entre phonèmes et donne des résultats assez satisfaisants. Mais au moment d'aborder le problème de l'alignement temporel du texte avec la partition, il est apparu qu'un contrôle assez précis et indépendant des positions et durées de chaque phonème est nécessaire, ce que ne permet pas la segmentation en diphones choisie. La segmentation en phonèmes apparaît donc finalement comme complémentaire à cette première approche. Une seconde couche de segmentation pour marquer les points de frontière entre les phonèmes dans chaque diphone a donc été effectuée. Celle-ci est conservée dans un second fichier de marqueurs au format SDIF, et les marqueurs correspondants apparaîssent en pointillé sur la figure 3.1. Malgré le travail supplémentaire nécessaire, cette double segmentation permet de conserver les avantages des 2 méthodes en permettant un contrôle précis sur chaque phonème tout en conservant une bonne qualité d'élocution.

La base de données segmentée et labelisée se présente donc finalement sous la forme de 3 fichiers, le premier contenant l'audio, le second contenant la segmentation en diphones et voyelles “stables”, et le 3$^{\text{ème}}$ contenant les points de séparation entre les phonèmes de chaque diphone. Le nombre total de marqueurs est d'environ 4000 (2600 dans le premier fichier pour la segmentation en diphones, et 1400 supplémentaires pour la segmentation en phonèmes dans le deuxième fichier).

3.2 Phonétisation du texte

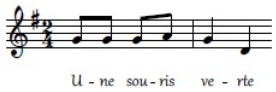
Afin que son utilisation soit la plus simple possible, sans nécessiter de connaissances particulières en phonétique, il est souhaitable que le système puisse prendre en entrée un texte écrit directement en français. Dans ce cas-là, une étape de phonétisation de ce texte est alors nécessaire afin de déduire les segments de la base utiles à la synthèse du texte. Pour ce faire, notre système utilise le programme “Liaphon”.

Ce programme permet, à partir d'un texte d'entrée écrit en langage usuel, de donner en sortie une description, sous forme d'un ensemble de symboles phonétiques, de la prononciation attendue du texte d'entrée (en prenant en compte un certain nombre de règles relatives au contexte). Le programme Liaphon a été développé par Frédéric Béchet au laboratoire d'informatique d'Avignon. Il est déjà utilisé notamment par le système d'alignement entre texte et audio ircamAlign, et pour la synthèse de la parole à partir du texte par le système ircamTTS. La sortie du programme Liaphon est une description phonétique donnée dans l'alphabet Lia. Ce programme s'utilise avec une interface en ligne de commande.

Cependant, l'alphabet Lia n'est pas le plus pratique et le plus compact à manipuler pour l'annotation de la base de données. L'alphabet SAMPA semble mieux adapté à cette tâche, et une traduction de l'aphabet lia vers la notation SAMPA est donc effectuée. SAMPA est un système de

conversion de l'alphabet phonétique international (API) en caractère ASCII 7 bits faciles à manipuler pour effectuer l'annotation, qui est également utilisé par le système présenté dans [16]. La plupart des symboles de l'alphabet SAMPA ne comportent qu'un seul caractère. La correspondance entre les symboles Lia et SAMPA sont donnés dans un tableau en annexe.

Contrairement à la voix parlée, où la compréhension du message porté par le texte prime, dans la voix chantée, c'est avant tout la prononciation qui importe, car chaque articulation est liée à une note et donc à un déroulement précis dans le temps en fonction du rythme déterminé par la partition, et des stratégies particulières sont donc nécessaire pour assurer une phonétisation du texte correspondant à la prononciation souhaitée pour le chant. Prenons en exemple, pour illustrer ce point, la comptine bien connue "une souris verte" dont voici la partition :



Dans cet exemple, le programme Liaphon, qui est avant tout prévu pour la parole, considérera les "e" de "une" et "verte" comme muets, et le résultat de la phonétisation sera donc, écrit en alphabet SAMPA : "y n s u R i v E R t". On pourra remarquer que ce résultat ne comporte que 4 syllabes. Il manque donc deux syllabes pour que le texte corresponde à la mélodie qui en requiert 6.

Pour parer à cela, il est donc possible d'insérer dans le texte initial des tirets (comme sur la partition ci-dessus) permettant de séparer les syllabes quand c'est nécessaire, avant la phonétisation, comme c'est aussi effectué dans [13].

On pourra donc écrire "U-ne souris ver-te", ce que le programme liaphon phonétisera alors en "y n @ s u R i v E R t @". Ainsi, le texte prononcé correspondra bien à ce qui est attendu pour coller à la mélodie. Evidemment, malgré cette possibilité, certaines erreurs peuvent toujours subsister. Mais le système permet également de prendre en entrée un texte directement en notation phonétique SAMPA. Lors de la phonétisation, le résultat de celle-ci est enregistrée dans un fichier texte que l'on peut alors modifier afin de corriger les erreurs, avant de le réutiliser comme texte d'entrée pour la synthèse.

Une fois l'étape de phonétisation effectuée, il est encore nécessaire de regrouper les phonèmes en diphones, en liant chaque phonème au suivant, afin d'identifier les segments à utiliser dans la base. En plus des diphones, des voyelles isolées sont également utilisées pour les tenues de notes. Ainsi, on obtient finalement la suite de labels suivants, correspondant aux segments de la base à utiliser : "#y y yn n@ @ @s su u uR Ri i iv vE E ER Rt t@ @ @#". Les "#" correspondent à des silences.

3.3 Concaténation

Le système effectue la concaténation des segments tel que décrit dans la partie 2.1. Mais simplement copier les trames d'analyses d'un segment après les trames du segments précédent dans le fichier final ne suffit pas pour obtenir un résultat satisfaisant à la resynthèse. En effet, les différents segments concaténés n'étant originalement pas contiguës dans la base de données, ils présentent des caractéristiques différentes, sources de discontinuités audibles et altérant donc le résultat de la synthèse. La figure 3.3 illustre ce problème en montrant le résultat de la concaténation de deux

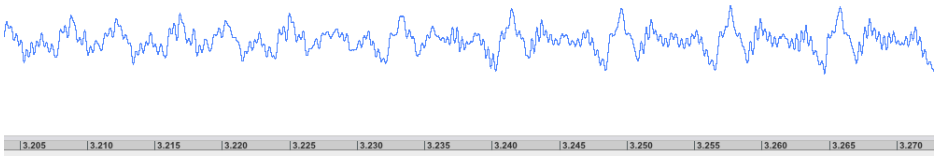


FIGURE 3.2 – Résultat de la concaténation de deux segments correspondants à la voyelle “i”

segments de la voyelle “i” provenant d’emplacements différents dans la base de données. On peut voir que la forme d’onde change de façon soudaine au milieu. Cette discontinuité est due à deux facteurs : l’enveloppe spectrale et les phases. Ce problème est notamment soulevé dans [17] et [15], qui présentent chacun des solutions pour effacer ces discontinuités. La solution adoptée dans [17] est notamment d’étaler les discontinuités de l’enveloppe spectrale et des phases sur un ensemble de trames autour de la jonction entre les deux segments.

Les sections suivantes présentent les solutions qui ont été implémentées avec succès dans notre système afin de supprimer ces discontinuités.

3.3.1 Interpolation des enveloppes spectrales

Bien que le timbre d’une voyelle soit caractérisé par ses formants, qui possèdent des fréquences et largeurs de bande supposées stables, un chanteur ne peut jamais produire deux sons exactement identiques, d’une part à cause du contexte de chaque voyelle (phonème précédent et suivant) qui influe sur la prononciation de cette dernière (à cause du phénomène de coarticulation), et d’autre part à cause d’une variabilité inévitable intrinsèque à tout son naturel. Pour cette raison, il peut exister d’un segment à l’autre des écarts suffisamment importants pour créer une discontinuité génante pour l’auditeur, qu’il est donc nécessaire de corriger. La figure 3.3 montre sous forme de sonagramme (réalisé dans audiosculpt) l’enveloppe correspondant au résultat de la synthèse de la phrase "une souris verte" de la comptine bien connue, au cours du temps. Les marques verticales représentent les points de jonction entre les segments. On peut bien observer sur cette figure les différences de formants aux niveaux de ces jonctions.

Une solution évidente pour supprimer cet effet est d’effectuer une interpolation des enveloppes spectrales au niveau des jonctions entre les segments. Pour cela, il est nécessaire d’estimer ces enveloppes et d’effectuer une séparation source-filtre du signal, afin de pouvoir effectuer l’interpolation.

L’estimation de l’enveloppe spectrale du signal est effectuée par superVP par l’algorithme “true-envelope” [41, 42]. En fonction de la fréquence fondamentale du signal analysé, son enveloppe spectrale est plus ou moins bien échantillonée, et l’ordre optimale nécessaire à l’analyse est donc variable. Comme évoqué dans la partie 2.2, la base de données utilisée présente d’assez fortes variations de hauteur, et il est donc nécessaire de fournir à l’algorithme la fréquence fondamentale du signal analysé afin d’obtenir une analyse fiable et homogène. L’ordre optimal est calculé, en fonction de la fréquence fondamentale, par la formule suivante :

$$\hat{O}_{TE} = (Fs/f0) * 0.5 \tag{3.1}$$

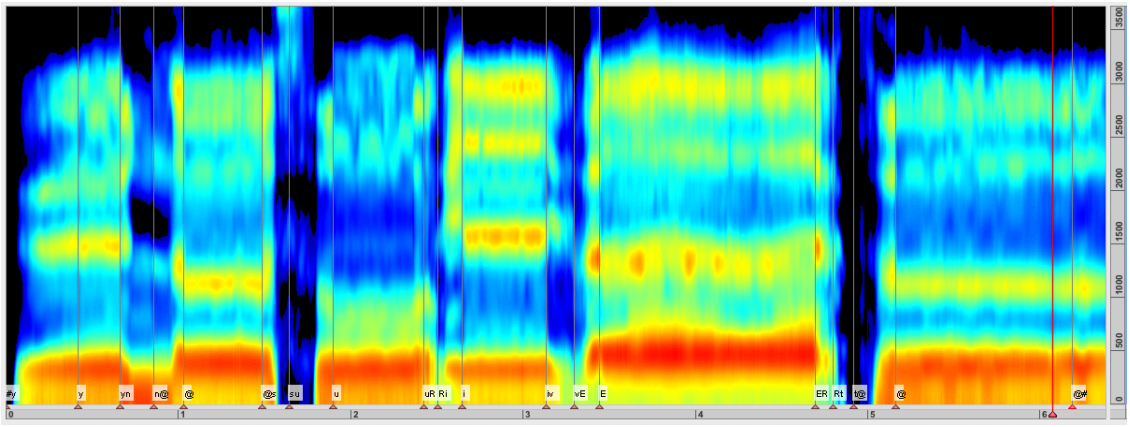


FIGURE 3.3 – Résultat de la synthèse du chant “une souris verte”, avec conservation des enveloppes spectrales originales

où $\hat{O}_{TE}$ est l'ordre optimal de l'algorithme, Fs est la fréquence d'échantillonage, et $f0$ la fréquence fondamentale [41]. La fréquence fondamentale des différentes trames du segment est donnée par une analyse effectuée par superVP avant l'application de l'algorithme true-envelope.

Une fois l'enveloppe spectrale extraite pour chaque trame du signal, on souhaiterait pouvoir l'interpoler au niveau des jonctions entre les segments. Dans le cas où deux segments joints présentent des écarts importants au niveau des formants, il serait préférable de pouvoir effectuer cette interpolation sur une durée assez importante, afin que celle-ci se fasse de façon progressive, en évitant donc des mouvements trop brusques de l'enveloppe qui seraient indésirables. De plus, on souhaite également conserver l'enveloppe originale intacte dans les zones de coarticulation. Pour les voyelles, on effectue donc l'interpolation sur toute la durée de la partie stable. Par exemple, pour concaténer les segments [su], [u], et [ur], on effectuera une interpolation entre l'enveloppe du “u” à la fin de [su] et celle du “u” au début de [ur]. Pour des voyelles particulièrement courtes ne présentant pas de partie stable, une durée minimum d'interpolation de 45ms est imposée, en prenant une marge de part et d'autre de la jonction. Dans le cas des consonnes, l'interpolation est effectuée sur une durée fixe, de part et d'autre de la jonction.

Pour interpoler les enveloppes spectrales, on concatène les analyses “true-enveloppe” des segments dans un fichier SDIF pour les trames dont on souhaite conserver l'enveloppe, avec les temps correspondants, et on ne copie rien là ou l'on souhaite interpoler les enveloppes. SuperVP peut ensuite utiliser un tel fichier pour remplacer l'enveloppe originale du signal par celle contenue dans le fichier, en effectuant une interpolation linéaire des enveloppes entre les trames les plus proches aux temps où l'enveloppe n'a pas été conservée.

La figure 3.4 montre le résultat d'une telle interpolation sur le même exemple que la figure 3.3. On peut voir sur la figure 3.4 que les variations de formants se font progressivement et que les discontinuités observées dans la figure 3.3 n'apparaissent plus.

La stratégie présentée dans cette partie pour l'interpolation des enveloppes dans le cas de voyelles tenues est similaire à celle présentée dans [18].

Une conséquence de l'interpolation des enveloppes spectrales est la suppression de toutes variations temporelles de l'enveloppe autre que le mouvement linéaire lié à l'interpolation sur les voyelles

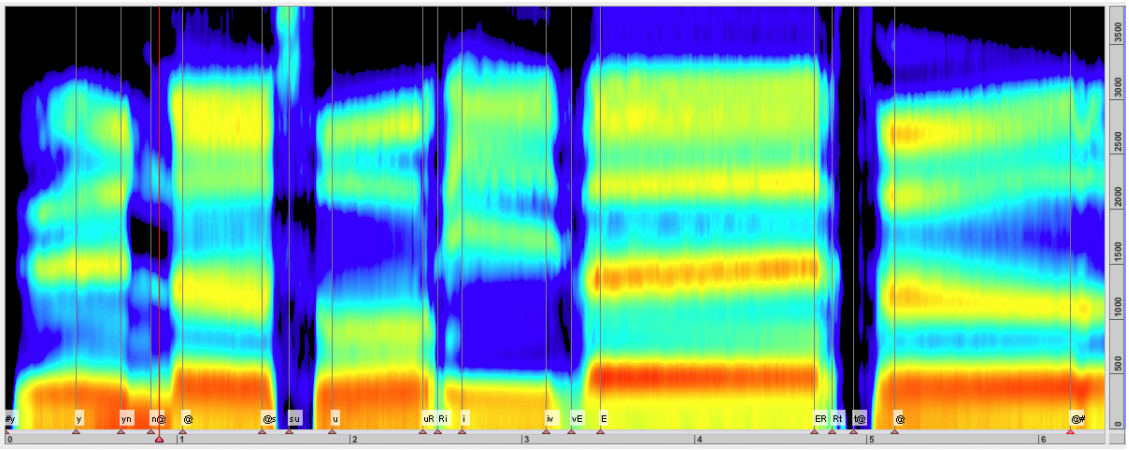


FIGURE 3.4 – Résultat de la synthèse du chant “une souris verte”, avec interpolation des enveloppes spectrales

tenues, ce qui donne un son plus droit et mécanique. Il serait donc souhaitable par la suite de réintroduire une certaine dose de variabilité de l'enveloppe spectrale pour obtenir un son plus naturel. Une autre possibilité, pour conserver des variations de l'enveloppe spectrale sur les notes tenues serait d'interpoler l'enveloppe “par morceaux”, sur de courtes durées, et non plus sur la totalité de la partie stable comme c'est le cas actuellement. On pourrait également exploiter la redondance de la base de données pour choisir le segment dont l'enveloppe spectrale colle le mieux à celles des segments voisins.

3.3.2 Correction des phases

Afin d'obtenir une qualité optimale du son lors de la resynthèse, l'algorithme SHIP présenté en 2.3.2 est utilisé pour effectuer les transformations, et en particulier la transposition. Comme dit précédemment, cet algorithme permet, pour les sons voisés, de conserver l'alignement vertical des phases d'une trame à l'autre. Pour cela, les phases des différentes harmoniques sont modifiées de la même manière, “en bloc”, et pour cette raison, seule la phase de l'harmonique ayant la plus forte amplitude, et donc la plus grosse influence sur le calcul de l'intercorrélation, sera vraiment bien ajustée. Mais, les relations de phases entre les différentes harmoniques étant, pour un son continu, à peu près constantes d'une trame à l'autre, les sauts de phases (horizontales) entre deux trames sur chaque harmonique restent faibles et imperceptibles.

Mais dans notre cas, nous cherchons à concaténer des segments de sons initialement non reliés. Rien n'assure donc que l'alignement vertical des phases des différents partiels soient les mêmes entre les deux segments. Dans ce cas, nous ne pouvons pas simplement utiliser l'algorithme proposé (SHIP), car d'importantes discontinuités de phases aparaissent alors, qui sont très génantes à l'écoute. La figure 3.5 illustre ce problème. Celle-ci montre l'évolution temporelle de 5 harmoniques pour deux segments joints présentant les mêmes amplitudes mais des relations de phases différentes. Le processus d'ajustement des phases réalisé par l'algorithme SHIP à la jonction entre deux trames est représenté ici de façon simplifiée. On peut observer sur ce schéma que la phase de la fondamentale est bien ajustée au niveau de “l'overlapp” entre les deux segments, car c'est elle qui

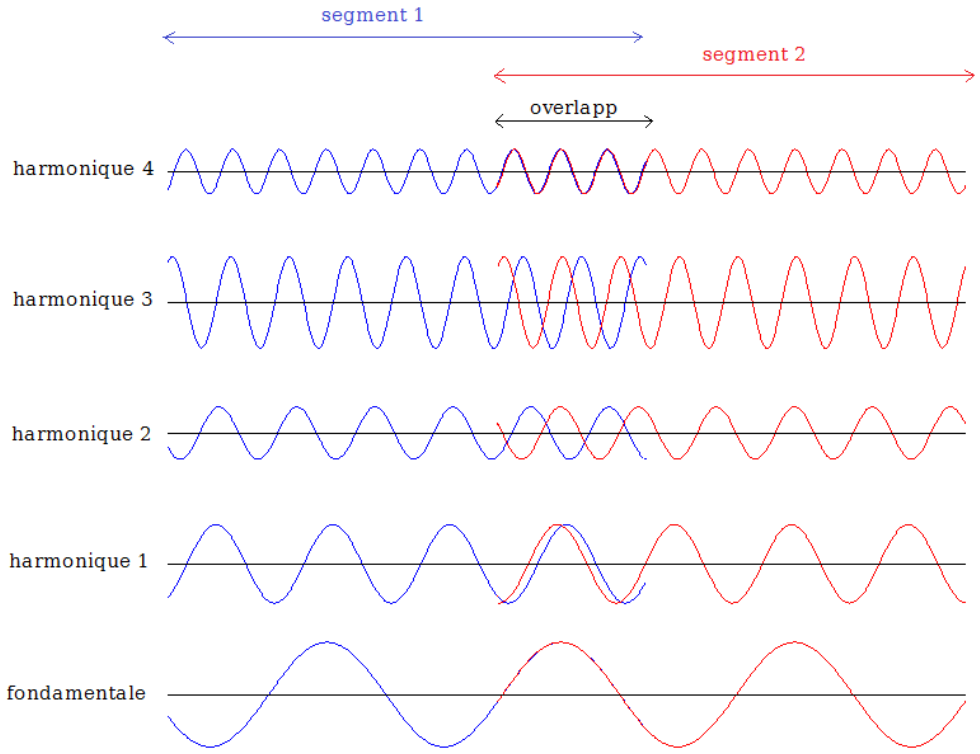


FIGURE 3.5 – Problème de cohérence des phases au niveau des jonctions entre segments avec l'utilisation de l'algorithme SHIP

a la plus grande amplitude et influe donc le plus sur le calcul de l'intercorrélation. Par chance, les phases sont également assez cohérentes pour l'harmonique 4. Mais pour les harmoniques 1, 2 et 3, les phases ne sont pas cohérentes, ce qui mène à des interférences destructrices entre les sinusoïdes des deux segments sur la durée de l'overlapp. Cela crée des discontinuitées dans les sinusoïdes (qui à gauche et à droite de l'overlapp ne sont pas perturbées) très génantes à l'écoute. Un exemple de cet effet peut être observé sur le spectrogramme présenté sur la partie gauche de la figure 3.6.

Si l'on utilise à la place de l'algorithme SHIP un vocodeur de phase classique, les phases des différentes harmoniques sont alors adaptées indépendamment les unes des autres, et ce problème disparaît. Mais on perd alors les avantages de l'agorithme SHIP évoqués précédemment, à savoir un rendu plus net et plus naturel. La solution adoptée est donc d'utiliser l'algorithme SHIP, mais en "étalant" les différences de phases, là où elles sont trop importante, sur un certain nombre de trames, avec des écart de phase suffisamment faibles pour être imperceptibles à l'oreille, en passant ainsi petit à petit de l'alignement de phases du segment de gauche à celui du segment de droite.

Il est donc nécessaire, dans un premier temps, de calculer les relations de phase entre les différentes harmoniques de la trame considérée. Etant donné que pour chaque sinusoïde, l'avancement de la phase est proportionnel à sa fréquence, ces relations de phases sont en constante évolution. Mais comme la voix produit des sons harmoniques, celles-ci se répètent de façon périodique. Afin

de pouvoir comparer ces relations de phases d'une trame à l'autre, il est nécessaire de déterminer un point d'ancrage pour les calculer. Nous avons donc choisi de calculer ces relations au point où la phase de la fondamentale est égale à 0. Pour repérer ces valeurs, nous nous appuyons sur le modèle sinusoïdal calculé par superVP lors de l'analyse de chaque segment. Théoriquement, les sons voisés étant harmoniques, seules les harmoniques devraient être représentées par le modèle, tout le reste étant considéré comme du bruit. Mais il arrive que le modèle repère d'autres sinusoïdes, à des fréquences intermédiaires, ne correspondant donc pas à des harmoniques. La connaissance de la fréquence fondamentale, grâce à une analyse de F_0 effectuée précédemment, permet de repérer facilement les harmoniques parmis les sinusoïdes du modèle, en ne considérant que les fréquences multiples de la fondamentale. Pour chaque harmonique ainsi répérée, la phase donnée dans l'analyse de superVP est celle au centre de la fenêtre, et est comprise dans l'interval $[-\pi; +\pi]$. On doit alors calculer la différence de temps dt entre le centre de la fenêtre et le point où la fondamentale a sa phase nulle. Celle-ci se calcul par la formule :

$$dt = -\varphi_{centre}/\omega_0 \tag{3.2}$$

avec

$$\omega_0 = f_0 \times Fe \times 2\pi$$

où φ_{centre} est la phase de la fondamentale au centre de la trame considérée, ω_0 est la fréquence fondamentale en $rad.s^{-1}$, f_0 est la fréquence normalisée donnée par superVP (la valeur 0.5 correspondant à la fréquence de nyquist), et Fe est la fréquence d'échantillonnage. La fréquence fondamentale n'étant en général jamais tout à fait constante, cette formule est une approximation.

Une fois le temps dt connu, on peut donc calculer la phase de chaque harmonique au point de la phase 0 de la fondamentale avec la formule :

$$\varphi_0^i = princarg(\varphi_{centre}^i + 2\pi \times i \times f_0 \times Fe \times dt) \tag{3.3}$$

où φ_0^i est la phase de l'harmonique i au point où la phase de la fondamentale est nulle, et φ_{centre}^i est la phase de l'harmonique i au centre de la trame. La fonction $princarg$ donne l'argument principal de la phase, compris dans l'intervalle $[-\pi; +\pi]$. Celui-ci se calcul de la manière suivante :

$$princarg(\varphi) = (\varphi + \pi)\%(-2\pi) + \pi \tag{3.4}$$

Une fois la relation de phase φ_0^i de chaque harmonique par rapport à la fondamentale connue, il est nécessaire d'appareiller les harmoniques entre deux trames successives afin de calculer les différences d'une trame à l'autre. Cela se fait simplement en retenant l'indice de chaque harmonique dans le modèle sinusoïdal, le rang de l'harmonique étant simplement donné par le rapport de fréquences entre l'harmonique considérée et la fondamentale. Cela nous permet donc de calculer la différence de relation de phase entre les trames n et $n-1$ pour l'harmonique i par l'égalité :

$$\Delta\varphi_0^i(n) = princarg(\varphi_0^i(n) - \varphi_0^i(n-1)) \tag{3.5}$$

Si celle-ci est supérieure à un certain seuil pour une harmonique donnée, il est alors nécessaire de corriger les phases dans la trame i afin de réduire cet écart. On applique donc la condition suivante :

$$\text{si } abs(\Delta\varphi_0^i(n)) > \Delta\varphi_{max}, \text{ alors}$$

$$\varphi_0^i(n) = princarg(\varphi_0^i(n-1) + \Delta\varphi_{max} \times sign(\Delta\varphi_0^i(n))) \tag{3.6}$$

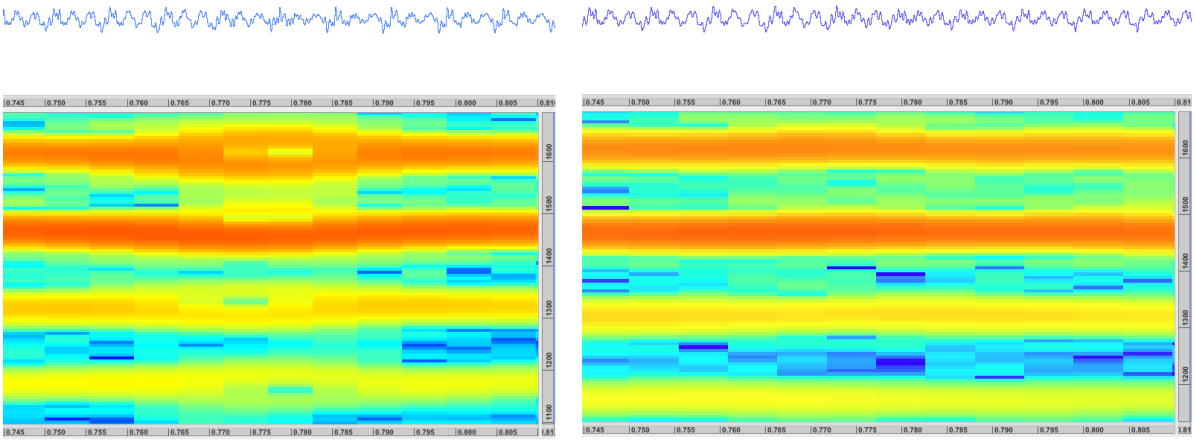


FIGURE 3.6 – comparaison de la jonction entre deux segments sans, puis avec la correction des phases

Ce processus est répété pour chaque trame, jusqu'à ce que l'écart soit inférieur au seuil $\Delta\varphi_{max} = 0.1$. Nous pouvons alors appliquer cette correction sur les phases du modèle sinusoïdale, qui sont utilisées par l'algorithme SHIP. Mais la resynthèse du signal est effectuée grâce aux données de la DFT effectuée par superVP, dont il est donc nécessaire de corriger également les phases. Pour cela, la correspondance entre les sinusoïdes du modèle sinusoïdal et les points de la DFT est donnée par superVP. La phase de chaque point de la DFT correspondant à une harmonique dont la phase a été ainsi corrigée est donc mise à jour de la même manière que la sinusoïde, en ajoutant la même correction.

La figure 3.6 montre l'effet de cette correction des phases sur le signal synthétisé, en comparant le sonagramme avec et sans la correction. Un zoom a été effectué sur 4 harmoniques du signal afin de bien observer l'effet sur chaque sinusoïde. On voit bien sur la partie droite de la figure que les discontinuitées à la jonction entre les deux segments ont disparues, grâce à cette correction. (L'axe horizontale correspond au temps en s, et l'axe vertical aux fréquences en Hz).

3.3.3 Suppression de la 1ère trame

Les fréquences correspondant à chacun des points de la DFT sont calculées en se basant sur la phase du point fréquentiel correspondant dans la trame précédente. Ce calcul n'est donc pas possible pour la première trame, pour laquelle les phases initiales sont alors considérées comme étant nulles, ce qui mène donc à des valeurs incorrectes. L'utilisation de cette trame lors de la concaténation posera donc problème au moment de la resynthèse, car les phases ne pourront pas être adaptées correctement. Afin d'éviter ce problème, il faut donc prendre soin de toujours supprimer la première trame de chaque segment au moment de la concaténation.

3.4 Transformations

3.4.1 Transposition

Afin de faire correspondre la fréquence fondamentale du signal de synthèse à la courbe générée par le modèle de contrôle, il est nécessaire de transposer le son original. SuperVP propose deux modes pour effectuer la transposition : celle-ci peut être appliquée par étirement temporel puis rééchantillonage dans le domaine temporel, ou bien dans le domaine fréquentiel directement au sein de chaque trame, avec une méthode telle que celle décrite dans [43]. A cause de la concaténation, la deuxième solution a été utilisée, car les trames de part et d'autre des jonctions peuvent avoir des hauteurs relativement éloignées, et il est nécessaire que les fréquences d'une trame à l'autre soient cohérentes au moment de l'overlap-add pour la resynthèse, afin d'éviter toute discontinuité. La première solution ne permet pas cela, puisque la transposition est obtenue au moment du ré-échantillonage, et donc après l'étape de resynthèse par overlap-add. De plus, pour conserver le timbre du son original, l'algorithme de transposition préserve l'enveloppe spectrale, en effectuant une séparation source-filtre au moyen de l'algorithme "true-envelope" [42] (cela est possible avec les deux modes de transposition).

Afin de connaître le bon ratio de transposition à appliquer, il est dans un premier temps nécessaire de connaître la fréquence fondamentale du son original. Pour cela, comme évoqué précédemment, nous effectuons une analyse de F_0 pour chaque segment, grâce à superVP, que nous concaténons dans un fichier SDIF. La taille de la fenêtre pour cette analyse doit être adaptée en fonction de la F_0 minimale attendue, car l'algorithme de détection de F_0 doit trouver les différentes sinusoïdes harmoniques et l'on doit donc s'assurer que la résolution est suffisamment fine pour pouvoir "résoudre" ces dernières. Pour cela, la taille de la fenêtre utilisée doit être d'environ 4 périodes de la F_0 minimale (~0.035s pour une F_0 de 115Hz).

Un fichier de paramètre est ensuite créé, contenant pour chaque trame le montant, en cents, de la transposition à appliquer. Un cent correspond à un centième de 1/2 ton. Ces valeurs sont calculées en fonction de la fréquence fondamentale originale F_0 et de la fréquence cible F_{cible}, par la formule :

$$1200 \times log2(F_{cible}/F_0) \tag{3.7}$$

La transposition est appliquée partout, même sur les consonnes non voisées (notamment les fricatives), car malgré la segmentation, la frontière entre les phonèmes est parfois un peu floue, et si le segment d'une consonnes contient le début d'une voyelle et que l'on ne le transpose pas, alors que l'on transpose la voyelle juste après, on risque d'obtenir un artefact, avec un saut de fréquence audible sur le début de la voyelle qui ne serait donc pas naturel. Comme les segments non voisés n'ont pas de fréquence fondamentale pour pouvoir calculer le ratio de transposition à appliquer, la F_0 la plus proche dans le segment concerné est utilisée. L'algorithme de détection de la F_0 donne un score de confiance pour chaque valeur de F_0 calculée. Afin de ne pas utiliser de valeur fausse, on utilise donc uniquement les valeurs au-dessus d'un seuil minimum.

Lors de la création du fichier de paramètres pour la transposition, une attention particulière doit être portée sur les temps spécifés pour chaque valeur de transposition. La raison à cela est que la base de données n'a pas une fréquence fondamentale fixe, et il peut donc y avoir un écart relativement important de fréquence fondamentale au niveau de jonctions. Il est donc absolument nécessaire d'appliquer un ratio de transposition différent pour les deux trames situées de part et d'autre de la jonction. Or, superVP effectue une interpolation linéaire entre les valeurs du fichier de paramètre pour les temps n'ayant pas de valeur de transposition spécifiée dans ce fichier. Dans

ce cas, une des trames risque d'utiliser un ratio de transposition intermédiaire qui conduira à une incohérence des fréquences après transposition de part et d'autre de la jonction, qui devraient être les mêmes pour éviter toute discontinuité. Les temps spécifiés dans le fichier de paramètre doivent donc correspondre précisément aux temps utilisés par superVP pour le calcul du ratio de transposition. Le changement de ratio de transposition au niveau d'une jonction doit donc survenir entre les temps utilisés respectivement par les trames gauche et droite afin que celui-ci soit instantané et qu'il n'y ait pas de risque d'interpolation entre les deux ratios.

Malgré l'attention portée à ce dernier point, un problème de cohérence a continué pendant longtemps à persister au niveau des jonctions entre segments. Une solution temporaire pour éviter ce problème a été d'utiliser une version secondaire de la base de données entièrement transposée sur une fréquence fixe. Ainsi, la F_0 étant constante, il ne risque plus d'y avoir de saut de fréquence entre deux segments, et le problème disparaît. Mais cette solution reste peu désirable, car le signal original est ainsi tranposé deux fois au lieu d'une seule, ce qui risque donc de produire plus d'artefacts. Finalement, une expérience utilisant des signaux de synthèse simples, comprenant 3 harmoniques à des fréquences et phases connues, ont permis d'identifier l'origine du problème. Ce dernier était dû au fait que pour le calcul de l'adaptation des phases effectué par l'algorithme SHIP, les phases utilisées sont celles avant la transposition, afin de pouvoir appareiller les sinusoïdes entre deux trames pour le calcul de l'intercorrélation. Cela ne pose pas de problème dans le cas général, puisque pour un son continu, la fréquence fondamentale d'une trame à l'autre varie peu. Mais dans le cas de la concaténation de segments avec des fréquences différentes, cela ne fonctionne plus. SuperVP a donc été mis à jour afin de prendre en compte ce cas en utilisant les phases après transposition au niveau de la jonction entre deux segments. Une nouvelle variable a été ajoutée dans les fichiers SDIF permettant d'identifier la dernière et première trame de chaque segment, pour lesquelles ce traitement particulier doit être appliqué.

Enfin, concernant la transposition, un dernier problème a été identifié, concernant la qualité du son obtenu après transformation. Lorsque l'on transpose un son avec conservation de l'enveloppe spectrale, on déplace des régions fréquentielles du signal de la source glottique vers une autre fréquence. Des zones de faible amplitude particulièrement bruitées peuvent être ainsi déplacées sur des formants, et être particulièrement amplifiées, résultant donc en un bruit trop important après la tranposition. Du fait que le bruit est principalement contenu dans les hautes fréquences, au-delà de la "Voiced-Unvoiced Frequency" (VUF), qui marque la séparation entre la partie voisée et la partie bruitée du spectre, ce problème se pose surtout pour des transpositions de ratio < 1. Il est donc préférable d'effectuer des transposition vers le haut que vers le bas. Aucune solution n'a pour l'instant été apportée à ce problème, mais une possibilité serait, au lieu de simplement déplacer en fréquence le signal original de la source glottique, de resynthétiser directement les sinusoïdes en-dessous de la VUF. Mais il faudrait, pour faire cela correctement, prendre en compte les paramètres du signal de source, ce qui n'est donc pas évident.

Une autre raison pour laquelle une transposition de ratio > 1 est préférable est que l'enveloppe spectrale est mieux échantillonée pour les sons grave que pour les sons aigües, et lors d'une transposition vers le bas, l'enveloppe spectrale, modélisée à partir du signal original, se retrouve donc sous-échantillonée.

3.4.2 Étirement temporel

Un deuxième type de transformations appliquées au signal est l'étirement temporel. Celui-ci est utilisé pour donner aux différents phonèmes la longueur souhaitée, déterminée par le modèle de contrôle. Afin de conserver une élocution naturelle, les zones de coarticulation ne sont pas étirées ni compressées. La longueur des notes tenues est donc obtenue par étirement de la partie centrale stable de la voyelle. Afin de limiter les ratios d'étirement utilisés pour des notes particulièrement longue, une solution possible serait de concaténer plusieurs fois un même segment de voyelle stable. cela permettrait d'obtenir la longueur souhaitée sans altérer les potentielles variations temporelles des caractéristiques du signal.

Chapitre 4

Vers une synthèse plus naturelle et plus expressive

Dans la partie précédente, un certain nombre de problèmes ont été exposés, concernant la concaténation et la transformation des différents segments. La plupart de ces problèmes ont été résolus, permettant ainsi une synthèse concaténative d'un signal de chant de bonne qualité, avec des jonction de segments "transparentes" pour l'auditeur, à partir d'un texte et d'une partition. Mais la partition ne contient que des informations de haut-niveau sur le rythme et la mélodie, qui ne suffisent pas à effectuer une synthèse réaliste, à la fois naturelle et expressive. Pour permettre cela, comme il l'a été exposé dans la première partie de ce rapport, un modèle de controle intégrant certaines règles doit être établi, permettant une gestion plus fine, et automatisée, des paramètres de la synthèse.

Il semble pertinent de faire ici une distinction entre les termes "naturel" et "expressif". Nous désignons par le terme "naturel" une synthèse que l'on peut confondre avec un véritable enregistrement de voix humaine. Le terme "expressif" désigne, lui, le fait que le rendu d'une synthèse est musicalement intéressant. On ne pourra par exemple pas dire d'un chanteur amateur que son chant n'est pas naturel, puisque ce dernier est produit directement par un être humain, mais celui-ci pourra être peu expressif, car la production d'un chant expressif demande une maîtrise particulière de sa voix pour laquelle un certain entraînement est nécessaire.

4.1 Modèle de controle

Dans cette partie sont exposés les travaux menés durant ce stage sur le développement d'un modèle de contrôle, dans le but de permettre une synthèse plus naturelle et expressive. Comme présenté dans [17] et dans la partie 1.2 de ce rapport, ce modèle prend en entrée la partition, le texte, et certains paramètres expressifs afin de générer une évolution fine, quasi-continue des paramètres de la synthèse, désignée dans [17] sous le terme *microscore*. De même, l'article [16] présente un *expressiveness module* avec un ensemble de paramètres ajustables pour chaque note. La sortie de ce modèle est ensuite donnée directement comme entrée du système de synthèse, dont il est totalement indépendant, afin de générer la performance de notre "chanteur virtuel". Les paramètres expressifs haut-niveaux qu'il est possible de spécifier sont par exemple le type d'attaque ou de

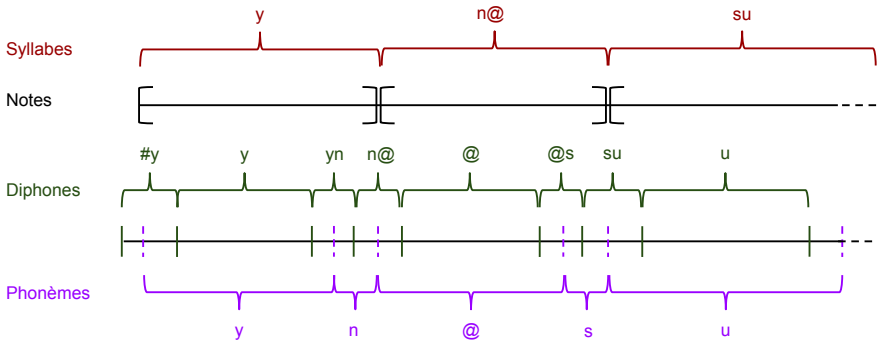


FIGURE 4.1 – Illustration des différents niveaux de segmentation

transition, ou la fréquence et l'amplitude du vibrato.

Le travail effectué sur cette partie a été mené sur la fin du stage et reste donc assez limité, mais a tout de même un apport non négligeable sur la qualité des résultats obtenus. L'aspect de la dynamique n'a pas du tout été abordé, et les travaux présentés ici portent donc uniquement sur l'aspect temporel et la gestion de la fréquence fondamentale. Les paramètres expressifs paramétrables sont pour l'instant fixés pour l'ensemble de la partition. Il serait bien évidemment souhaitable de pouvoir appliquer une paramétrisation indépendante pour chaque note de la partition, mais cela n'a pas encore été implémenté. Une autre possibilité intéressante serait d'apprendre des règles de contrôle à partir d'une pièce de chant enregistrée.

4.1.1 Alignement temporel

Comme expliqué dans la première partie, l'alignement temporel désigne à la fois l'alignement du début de chaque note avec les phonèmes, et la durée de chaque phonème. Pour le chant, on peut distinguer plusieurs niveaux de segmentation qu'il semble utile de prendre en considération, en relation aux double aspect linguistique et musical du chant, et il est donc nécessaire d'établir des relations entre ceux-ci. Ces différents niveaux de segmentation sont la note et le phonème, et, du fait du système utilisé pour la synthèse dans le cas présent, le diphone. A ceux-ci peut également s'ajouter la syllabe. Le schéma de la figure 4.1 illustre la correspondance entre ces différents niveaux de segmentation. (Il est à noter que nous entendons ici par le terme diphone l'ensemble des segments utilisés pour la synthèse, ce qui concerne donc également les parties stables des voyelles)

j	w	p	t	k	b	d	g	f	s	S	v	z	Z	l	R	m	n
0.16	0.15	0.18	0.18	0.15	0.13	0.14	0.15	0.22	0.22	0.22	0.15	0.18	0.18	0.18	0.12	0.18	0.18

TABLE 4.1 – Durée fixée pour chaque consonne, en s.

La syllabe étant une unité segmentale de base en parole, après le phonème, on aurait pu intuitivement penser à aligner chaque syllabe sur une note. C'est d'ailleurs ainsi qu'est écrit le texte sur une partition de chant, en positionnant chaque syllabe sous une note, et c'est ce qui est également représenté sur la figure 4.1. Mais les résultats de précédentes études soutiennent que pour le chant, les notes doivent être alignées non pas sur les syllabes, mais sur le début des voyelles [29, 44, 8, 13, 16, 18], et c'est donc également ainsi que nous avons choisi de procéder. Les consonnes éventuelles que l'on peut trouver au début d'une syllabe se retrouvent donc associées à la note précédent celle associée à la voyelle de la syllabe en question, formant ainsi une sorte de "syllabe inversée" avec la voyelle précédente.

Pour effectuer l'alignement temporel, on commence donc par associer chaque phonème du texte à une note, en faisant en sorte de commencer une nouvelle note à chaque syllabe (si une voyelle est tenue sur plusieurs notes, alors celle-ci doit être répétée dans le texte autant de fois que le nombre de notes correspondantes). Ensuite, on calcule la durée de chaque phonème par rapport à la durée de la note associée. Pour chaque consonne, une durée fixe par défaut est associée. Ces durées ont été déterminées "manuellement", en faisant l'hypothèse que l'articulation des phonèmes est particulièrement exagérée dans les enregistrement de la base de données, et elles correspondent donc à des valeurs légèrement inférieures à la moyenne des durées mesurées dans la base pour chaque phonème. Pour chaque note, la voyelle occupe le temps restant après avoir soustrait la longueur des consonnes à la durée de la note (la durée de la note étant donnée dans la partition en fonction du tempo). Le tableau 4.1 donne les durées fixées pour chacune des consonnes. Cette approche donne des résultats satisfaisants, avec une élocution naturelle, pour des tempo relativement lents. Mais il serait nécessaire par la suite d'évaluer la nécessité de compresser plus ou moins la durée des consonnes lorsque le tempo est plus rapide, ou pour une partition contenant des notes particulièrement courtes. Avec l'application de ces règles, on obtient donc en sortie du module de contrôle un tableau contenant le temps de début de chaque phonème (silences inclus) attendu à l'issu de la synthèse, qui peut ensuite être utilisé en entrée de n'importe quel système de synthèse. Pour le moment, les temps et durées de phonèmes générés sont entièrement déterminés par la partition et les règles évoquées. Mais dans le cas d'un chant polyphonique avec plusieurs voix chantant un même texte, il serait nécessaire d'introduire une certaine variabilité afin que les voix ne soient pas alignées exactement de la même manière, ce qui ne serait pas naturel. Dans [8], Xavier Rodet évoque l'utilisation de différences de temps de 20 à 200ms entre les différentes voix, au niveau des onsets de chaque note.

Ce qui suit concerne les méthodes utilisées dans notre système de synthèse afin d'obtenir un résultat correspondant aux temps ainsi déterminés pour chaque phonème, et ne fait donc pas vraiment partie du modèle de contrôle. Néanmoins, cela est tout de même relié au problème de l'alignement temporel, et il semble donc pertinent d'en parler ici.

La deuxième couche de segmentation en phonèmes effectuée sur la base de données permet de connaître la durée originale de chaque phonème dans la base après concaténation des segments, et d'en déduire ainsi les ratios de compression nécessaires pour donner à chaque phonème la durée cible, ainsi que les temps auxquels ces ratios doivent s'appliquer lors de la resynthèse. En ce qui

concerne les voyelles, comme expliqué dans la partie 3.1, celles-ci sont réparties sur trois segments différents, le premier contenant la transition avec le phonème précédent, le second contenant la partie stable ou "noyau" de la voyelle, et le troisième contenant la transition entre la voyelle et le phonème suivant. Afin de donner à la voyelle la durée voulue, plusieurs règles sont établies.

- Si la somme des durée de ces trois morceaux est inférieure à la durée totale souhaitée, alors on obtient cette durée cible en étirant uniquement la partie stable de la voyelle, comme c'est également le cas dans [44]. Le ratio d'étirement est donc égal à $(d_{cible} - (d_1 + d_3))/d_2$ où d_{cible} est la durée cible de la voyelle, et d_1, d_2, et d_3 sont respectivement les durées des trois parties de la voyelle.
- Si la durée totale initiale des trois parties de la voyelle est plus longue que la durée souhaitée, alors on ne sélectionne qu'une partie du segment du noyeau, de manière à ajuster la durée d_2 pour obtenir l'égalité $d_1 + d_2 + d_3 = d_{cible}$.
- Si la durée cible de la voyelle est particulièrement courte, il est possible que même avec $d_2 = 0$, on ait encore $d_1 + d_3 > d_{cible}$. Dans ce cas, on n'utilise plus du tout la partie stable de la voyelle, et l'on joint directement les deux parties de transition entre elles en ajustant d_1 et d_3 de la même manière pour que $d_1 + d_3 = d_{cible}$. Pour ajuster ces valeurs, on diminue le temps de fin du premier diphone et on augmente le temps de début du deuxième de la valeur $\Delta = (d_1 + d_3 - d_{cible})/2$, où d_1 et d_3 correspondent aux durées initiales de chaque morceau de transition de la voyelle avant ajustement des temps.

Les temps ainsi ajustés correspondent aux temps des segments à extraire de la base pour la concaténation. De cette manière, les parties de transition contenues dans les diphones ne sont jamais étirées ni compressées, et il semble que cette manière de faire corresponde dans une certaine mesure à ce qui se passe dans la réalité, car lorsque l'on parle rapidement, à cause de la coarticulation, les voyelles n'ont pas le temps d'atteindre un régime stables et sont en constante transition entre les phonèmes précédent et succédant la voyelle. Néanmoins, cette stratégie pose problème pour des notes vraiment courtes car les mouvements des formants ne vont plus assez loin vers leur valeur cible (correspondant à leur position dans la partie stable) pour reconnaître la voyelle prononcée. Cette question n'est pas vraiment abordée dans la littérature, et une étude plus poussée avec des tests perceptifs serait donc nécessaire pour déterminer dans quelle mesure les durées des consonnes devrait être alors réduites pour laisser plus de temps à l'évolution de la voyelle, ou bien dans quelle mesure les parties de transitions devraient être également compressées (au lieu d'être simplement coupées) pour arriver plus rapidement sur le régime stable de la voyelle. La stratégie développée dans [44] dans une telle situation est de compresser l'ensemble de la syllabe de la même manière (mais le calcul de la durée de la syllabe n'est pas clair, puisque les notes tombent sur les débuts des voyelles). L'article [13] présente une variante de ce modèle.

Le calcul des ratios d'étirement et les temps auxquels ils doivent s'appliquer permettent de générer un fichier de paramètre utilisé au moment de la resynthèse pour obtenir les durées de chaque phonème désirées.

4.1.2 Contrôle de la F_0

Le paramètre certainement le plus important pour générer une synthèse à la fois naturelle et expressive est la gestion de la fréquence fondamentale. Un son parfaitement droit est impossible à produire vocalement et sonnerait très mécanique. On ne peut pas se contenter de produire pour chaque note la fréquence exacte correspondant à sa hauteur, et il est donc indispensable d'introduire une certaine dose de variations. Cette problématique a déjà été abordée dans la partie 1.2.1 où

différents types de variations ont été identifiés. Nous pouvons distinguer notamment le vibrato, les transitions, ainsi que des variations aléatoires de la fréquence fondamentale. Ce qui suit présente une première investigation dans la problématique du contrôle de la F_0 et se base principalement sur les résultats des différents articles cités dans la partie 1.2.1. Afin d'aller plus loin, il serait nécessaire de constituer un véritable corpus d'extraits de chant, dans différents styles et avec différents interprètes, pour pouvoir vraiment analyser ces variations de fréquence fondamentale, en fonction du contexte.

4.1.2.1 Transitions

Le type de courbe à utiliser pour créer des transitions réalistes est un point très peu documenté dans les différents articles sur la synthèse du chant. Dans le système actuel, les transitions entre deux notes sont modélisées sous forme d'une simple rampe, par interpolation linéaire des fréquences des deux notes. Un paramètre "portamento" permet de régler la durée de la transition. Ce modèle n'est pas réaliste, mais est très simple à implémenter, et permet tout de même une variation continue de la fréquence, préférable à un saut instantané. La transition intervient avant l'onset de chaque note, comme suggéré dans [44] et [8]. Pour un texte quelconque, la transition aura donc souvent lieu sur les consonnes juste avant le début de la voyelle. Dans le cas de consonnes non voisées, la forme de cette transition a donc probablement peu d'importance. Cependant, dans le cas d'un enchaînement de voyelles, celles-ci ont certainement un rôle très important à jouer dans le réalisme du résultat. Pour cela, une approche déjà préférable à une simple rampe linéaire serait l'utilisation de la fonction sigmoïde $(1 - \exp(-x))$, qui permettrait une transition moins segmentée.

Les articles [25, 33, 31] font état de deux types de variations intervenant aux niveau des transitions, qui sont la préparation et l'overshoot, déjà présentées en 1.2.1. D'après les résultats présentés, celles ci semblent jouer des rôles importants dans la perception du résultat pour un rendu réaliste, et il serait donc intéressant d'implémenter de telles variations. Mais elles sont modélisées par une équation d'un système du second ordre, ce qui semble assez peu intuitif pour le contrôle. Il serait donc bon de trouver une autre solution plus intuitive pour implémenter de telles variations.

4.1.2.2 Vibrato

Le vibrato est un type de variation caractéristique concernant les signaux de voix chantée et est particulièrement importante pour l'expressivité, et ce dans la plupart des styles de musique (bien qu'il soit moins présent dans des styles de chant pop ou rock). Il est notamment absolument essentiel pour le chant lyrique. Les différents paramètres et effets du vibrato ont déjà été évoqués dans la partie 1.2.1. Dans notre implémentation, nous nous sommes principalement basés sur l'article [30] pour l'étude du vibrato.

Un paramètre important pour le vibrato non évoqué dans la partie 1.2.1 est le type de courbe régissant l'évolution temporelle de son amplitude, et éventuellement, associés à cette courbe, le temps d'attaque et de relachement (release). La courbe implémentée dans notre système est du type "attack-sustain-release", avec un temps d'attaque et un temps de release paramétrables de façon indépendante. Une autre possibilité assez simple serait l'utilisation d'une courbe de type hanning. Le simple ajout d'un vibrato, sur les voyelles tenues, permet déjà d'obtenir un résultat bien plus convaincant qu'avec un son plat.

Dans [30], il est notamment noté que l'un des effets du vibrato est une modulation d'amplitude synchrone à ce dernier, due au "balayage" des formants par les différentes harmoniques. Dans notre implémentation, la simple modulation de fréquence de la source permet de créer cet effet perçu du vibrato, grâce à la préservation de l'enveloppe spectrale lors de la transposition.

Comme dans [30], nous générons le vibrato à l'aide d'une table d'onde. Cela consiste à précalculer à l'avance une période du vibrato, stockée en mémoire sous forme d'un tableau, et à lire les valeurs contenues dans ce tableau, en avançant à chaque fois d'un certain pas, et en bouclant quand on arrive à la fin de la période. Le vibrato, tout comme la courbe de F_0 générée, est échantilloné à une fréquence de 1000Hz. La fréquence du vibrato peut être variée en adaptant le pas d'avancement entre chaque valeur extraite de la table d'onde, et l'amplitude peut être variée en multipliant simplement chacune de ces valeurs par un facteur. Cette méthode est très peu coûteuse en calculs.

Afin de tenter d'obtenir un résultat moins mécanique qu'avec un simple vibrato régulier, nous avons tenté d'ajouter, comme dans [30], certaines variations aléatoires de la fréquence et de l'amplitude. Celles-ci sont implémentées sous forme d'un bruit blanc filtré avec un filtre passe-bas. Les filtres utilisés sont des filtres RII avec, pour les variations d'amplitude, une fréquence de coupure à 5.5Hz, et pour les variations de fréquence du vibrato une fréquence de coupure à 3.5Hz, selon les valeurs suggérées dans [30]. L'étude présentée dans [30] conclu en disant que le résultat d'écoutes informelles indique que l'ajout de variations aléatoires est détecté par rapport à des paramètres stables, mais aucune préférence notable n'est relevée, et ces variations ne semblent donc pas essentielles. Mais il est tout de même indiqué à la fin de l'article que des études plus poussées sur ce point seraient nécessaires. L'écoute des premiers résultats obtenus par notre implémentation semble confirmer cela. Ces variations semblent en effet être peu naturelles si l'amplitude choisie est trop élevée, et sont peu perceptibles si l'on en diminue l'amplitude. Mais très peu de tests ont été effectués et une investigation plus approfondie serait nécessaire, avec différentes configurations des paramètres, afin de déterminer la pertinance de telles variations aléatoires.

4.1.2.3 Variations aléatoires de la F_0

Nous avons vu que l'ajout de vibrato permet de donner à la voix un caractère plus expressif. Mais, même sans vibrato, le son ne peut être parfaitement plat. Dans [44] et [8], les auteurs suggèrent que l'ajout d'une certaine dose de variations aléatoires de la fréquence fondamentale (hors vibrato) permet de donner à la voix un caractère plus humain. Nous avons donc implémenté la possibilité d'ajouter de telles variations, sous forme d'un bruit filtré passe-bas avec une fréquence de coupure à 3.5Hz. Le résultat de ce filtrage est ensuite normalisé entre les valeurs -1 et 1, afin de pouvoir en contrôler aisément l'amplitude par multiplication par un simple facteur. Tout comme les variations aléatoires du vibrato, l'influence de ces variations sur la perception du résultat n'a pas été évaluée.

Le modèle de contrôle de la fréquence fondamentale, avec l'implémentation de ces différentes variations, génère une courbe de F_0 cible pour l'ensemble du chant à synthétiser, qui est ensuite donnée en entrée du système de synthèse. Comme dans l'implémentation du système la transposition est effectuée avant l'étirement temporel, il est nécessaire de calculer la courbe de F_0 cible avant l'étirement temporel, afin que le résultat final après l'étape d'étirement-compression temporel corresponde bien à la courbe générée par le modèle de contrôle. Pour cela, il suffit de conserver les valeurs de F_0 cible, mais en modifiant les temps correspondants en fonction des ratios d'étirement appliqués par la suite.

Le schéma de la figure 4.2 illustre le processus générale pour la production de la fréquence fondamentale, avec les différentes variations évoquées précédemment. Celui-ci est identique à celui de l'article [30].

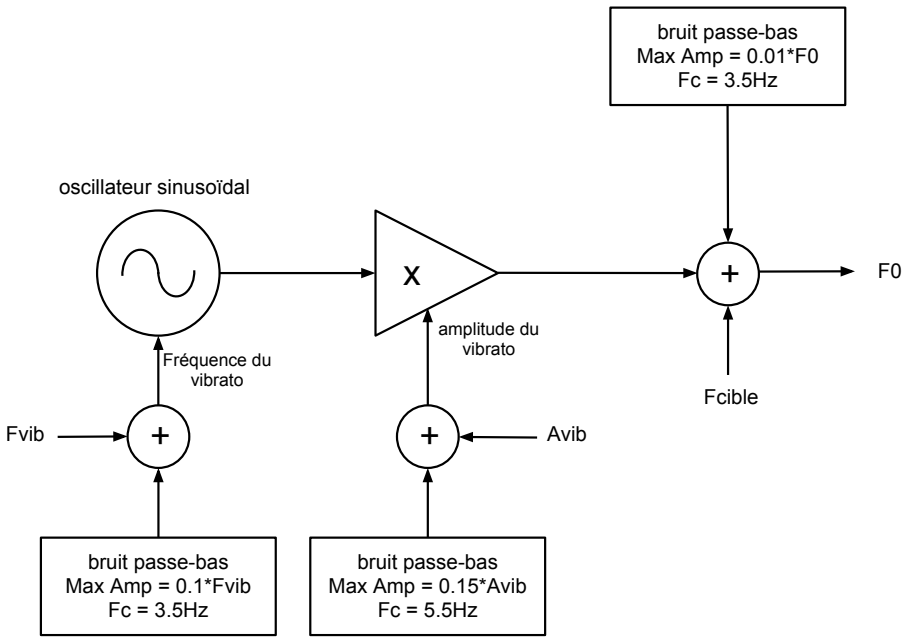


FIGURE 4.2 – Calcul de la F_0, avec les différentes variations

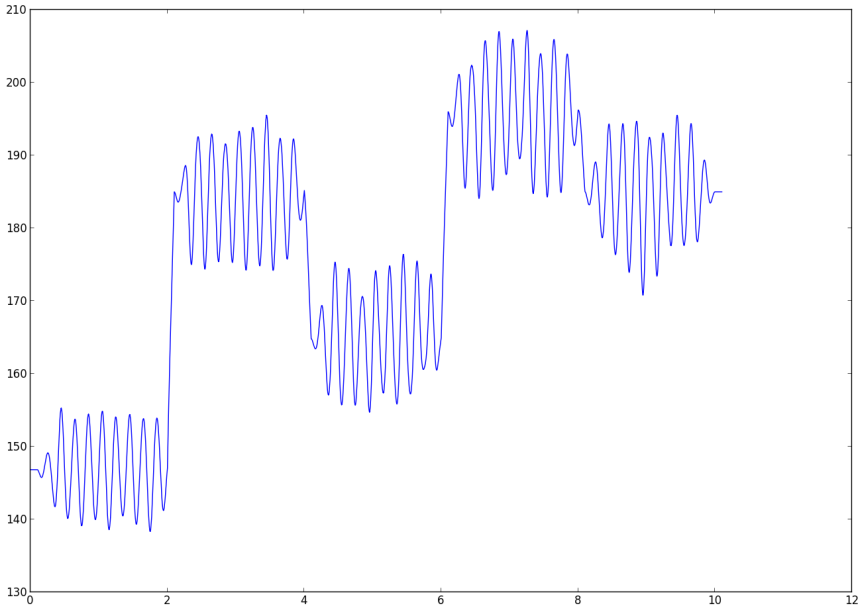


FIGURE 4.3 – Exemple de courbe de F_0 générée avec les différentes variations

La figure 4.3 montre un exemple de courbe de F_0 ainsi générée par le module de contrôle, ou l'on peut observer notamment les variations d'amplitude du vibrato. L'axe horizontale de cette figure correspond au temps en s, et l'axe vertical aux fréquences en Hz.

4.1.3 Dynamique

Comme précisé précédemment, la question de la gestion de la dynamique du signal n'a pas du tout été abordée durant ce stage. L'amplitude du signal en sortie correspond donc à l'amplitude originale du signal enregistré dans la base de données. Cependant, pour pouvoir tout de même modéliser les attaques et fin des notes sans avoir à modifier l'amplitude du signal, les silences sont considérés comme étant des consonnes, et les attaques et fin de note (ou "release") sont contenus dans des diphones de types "#a" et "a#", où # désigne un silence. Cela permet donc des attaques et release naturels, mais de durée non réglable, sans avoir à changer la dynamique du signal original. Par la suite, il faudra modifier cela pour utiliser des variations de dynamiques contrôlables directement sur les phonèmes concernés.

4.2 Prise en compte de la source glottique

Nous avons vu qu'une bonne gestion des paramètres de la synthèse comme la durée des phonème, ou l'évolution de la fréquence fondamentale permettent de conférer un caractère plus naturel à la voix de synthèse. Mais ce caractère naturel passe également par la gestion du timbre. Les diverses transformations effectuées sur la voix, et notamment la transposition, s'appuient sur un modèle source-filtre de production de la parole. Un tel modèle suppose la production d'un signal harmonique de source généré par des impulsions à intervales réguliers au niveau de la glotte, qui sont ensuite filtrées par le conduit vocal. En effet, lors de la transposition, afin de garder les caractéristiques spectrales, et donc timbrales, de la voix originale, il est nécessaire de séparer le signal de source de l'enveloppe spectrale, qui peuvent alors être modifiés indépendamment, afin de manipuler d'un côté la hauteur du son, et d'un autre côté le timbre. Mais les impulsions glottiques ne peuvent être assimilées à de simples diracs, et possèdent donc une enveloppe spectrale propre indépendante de l'enveloppe liée au conduit vocal. Alors que l'enveloppe spectrale liée au conduit vocal dépend principalement de l'articulation et donc des différents phonèmes prononcés, l'enveloppe de la source glottique est, elle, en premier lieu dépendante des paramètres expressifs tels que la hauteur et l'intensité du son produit. Afin de produire une synthèse plus naturelle, il semble donc nécessaire de pouvoir manipuler séparément ces deux enveloppes. Il faut donc pour cela être capable d'estimer l'enveloppe spectrale liée à la source glottique, et la soustraire ensuite à l'enveloppe global du son analysé par l'algorithme "true-enveloppe". Une telle manipulation est proposée dans superVP pour une utilisation lors de la transposition. L'enveloppe du signal de source est alors transposée, alors que l'enveloppe liée au conduit vocal ne change pas. Mais l'estimation des paramètres de la source glottique est encore trop peu robuste pour être tout à fait fiable. Toutefois, cette possibilité a été testée, et apporte, dans certains cas, une certaine amélioration dans le rendu du timbre. L'influence de cette prise en compte de la source glottique pour la transposition pour la synthèse du chant a notamment été étudiée dans [24].

Chapitre 5

Évaluation

Une évaluation rigoureuse et comparative des résultats obtenus par rapport à l'état de l'art est difficile pour plusieurs raisons. D'une part, il semble qu'il n'existe pas à ce jour de système pour le chant en français, la plupart des systèmes actuellement disponibles ne permettant pour le moment que des synthèses en anglais ou en japonais. Ces langues présentant des prononciations très éloignées du français, la qualité de l'élocution est donc difficile à comparer avec nos résultats. De plus, le système développé ne prend pas en compte à ce jour l'aspect dynamique du chant, important pour le caractère naturel et l'expressivité du résultat, et l'ensemble du modèle de contrôle est encore assez peu avancé. Une évaluation rigoureuse des résultats aurait probablement nécessité la mise en place d'une étude perceptive. Mais, même dans l'état actuel du système, un nombre déjà assez important de paramètres peuvent être modifiés et influer sur le résultat obtenu. La tenue d'une telle étude aurait donc nécessité un grand nombre de tests couvrant l'ensemble des combinaisons de paramètres, ce qui n'est donc pas évident à mettre en place, et nécessite un certain temps. Bien que cette possibilité ait été évoquée, la durée du stage n'a pas permis de mener une telle étude.

Mais en-dehors du caractère perceptif concernant la qualité obtenue, certains critères objectifs permettent tout de même d'évaluer le système par rapport aux objectifs fixés. En générant, à partir du modèle de contrôle, des fichiers de marqueurs pour les temps attendus des différents phonèmes, nous avons par exemple pu vérifier que l'alignement temporel correspond bien, dans le fichier audio synthétisé, à ce qui était attendu, puisque les différents phonèmes sont bien alignés sur les marqueurs correspondants. De plus, les différentes solutions apportées aux problèmes soulevés par le processus de concaténation ont permis d'effacer toute discontinuités, créant ainsi des jonctions tout à fait transparentes pour l'auditeur, et le système atteint donc de ce point de vue les objectifs fixés. Enfin, l'élocution est, pour des tempo relativement lents, assez naturelle, et donc plutôt satisfaisante de ce point de vue. Cependant, la durée permise pour effectuer l'interpolation des enveloppes spectrales étant assez courte pour les consonnes, il arrive parfois que l'on perçoive tout de même des variations très rapides des caractéristiques spectrales et de l'amplitude des segments aux jonctions qui paraissent alors peu naturelles.

Du point de vue des performances, la synthèse n'est pour l'instant pas très rapide, notamment avec la correction des phases, mais cela est dû aussi en bonne partie au temps nécessaire aux différentes analyses qui sont effectuées au fur et à mesure pour chaque segments. On pourrait donc facilement accélérer le processus de synthèse en effectuant l'ensemble de ces analyses "hors-ligne", sur l'ensemble de la base. Il est à noter que cela demanderait beaucoup d'espace mémoire, car la

quantité de données contenues dans ces analyses est plus importante que celle contenue dans le fichier audio. Nous avons cependant pour le moment, pour les besoins du développement, préféré continuer de faire ces analyses au fur et à mesure, afin de pouvoir changer facilement les paramètres, sans avoir à chaque fois à réanalyser toute la base, ce qui serait très long.

Conclusion

Les travaux effectués dans ce stage auront permis de développer un premier système de synthèse du chant dont les résultats restent à évaluer. Un des avantages notables du système développé est l'utilisation de l'agorithme SHIP, qui mène à des transformations de haute qualité. Une des difficultées rencontrées durant ce stage est que superVP et l'algorithme SHIP étaient initialement prévu pour traiter des sons continus, pour lesquels toutes les trames d'analyses sont cohérentes par rapport à la trame précédente. Cela à donc posé des problèmes lors de son utilisation sur des données concaténés, comme évoqué dans ce rapport, en particulier à la section 3.6. Mais ces problèmes ont été identifiés et superVP a ainsi pu être adapté pour pouvoir prendre en compte ce nouveau cas, non prévu initialement. Des solutions concernant l'adaptation des phases et l'interpolation des enveloppes spectrales ont été mises en places, permettant de lisser les jonctions entre segments, les rendant ainsi imperceptibles à l'auditeur, et permettant donc une élocution naturelle.

Un modèle de contrôle a ensuite été developpé afin de permettre un contrôle réaliste et expressif des paramètres de la synthèse. Celui-ci recouvre la gestion de l'alignement temporel et la gestion de la fréquence fondamentale. La stratégie implémentée pour l'alignement temporel donne des résultats satisfaisants pour des vitesses d'élocution relativement lentes. Des expérimentations sont encore nécessaires afin de trouver un modèle plus adéquate pour permettre une synthèse naturelle avec des vitesses d'élocution plus rapides. Les règles de contrôle de la F_0 implémentées ont eu un apport non négligeable sur la qualité des résultats obtenus, notamment grâce à l'ajout de vibrato et de transitions continues. La possibilité d'ajouter des variations aléatoires du vibrato et de la fréquence fondamentale a également été implémentée, mais leur apport semble relativement limité, bien qu'il n'ait pas été rigoureusement évalué. Des efforts sont donc encore nécessaires au niveau du modèle de contrôle afin d'affiner et compléter les différentes règles déjà implémentées pour le contrôle de l'alignement temporel et de la fréquence fondamentale. Un troisième aspect important pour l'expressivité du chant est la dynamique du signal, qui n'a pas du tout été abordé durant ce stage, mais devrait par la suite être pris en compte lors de la synthèse.

Au-delà du modèle de contrôle, une autre piste à explorer pour améliorer la qualité des résultats obtenus serait la prise en compte des caractéristique de la source glottique lors de la transposition, de manière à adapter au mieux le timbre de la voix en fonction de la hauteur et de l'intensité. Une possibilité pour cela serait également de complètement synthétiser la source, ce qui permettrait un contrôle direct de ses paramètres. En ce qui concerne la gestion de l'enveloppe spectrale, l'interpolation linéaire effectuée actuellement n'est probablement pas très naturelle, et il serait donc utile d'étudier plus précisemment les mouvement de formants entre deux zone de coarticulation afin de produire un modèle d'interpolation plus réaliste.

Enfin, il serait probablement souhaitable, pour obtenir un résultat plus naturel, d'utiliser une base de données plus stable, enregistrée sur plusieurs hauteurs et niveaux d'intensité, et d'autres expérimentation sur le contenu de la base et la stratégie de sélection des segments seraient enisageables.

En conclusion, ce stage, bien qu'il n'ait pas permis d'apporter des solutions à tous les problèmes, aura permis d'identifier les principales problèmatiques relatives à la synthèse du chant et pose ainsi les bases nécessaires pour pouvoir effectuer des recherches plus avancées sur le sujet, qui seront notamment menées dans le cadre d'une thèse que j'entreprendrai à l'IRCAM, au sein de l'équipe analyse-synthèse, à partir de l'année prochaine.

Bibliographie

[1] Homer Dudley and Thomas H Tarnoczy. The speaking machine of wolfgang von kempelen. *The Journal of the Acoustical Society of America*, 22 :151, 1950.

[2] Xavier Rodet. Synthesis and processing of the singing voice. In *Proc. 1st IEEE Benelux Workshop on Model based Processing and Coding of Audio (MPCA-2002)*, pages 15–31, 2002.

[3] Jordi Janer, Jordi Bonada, and Merlijn Blaauw. Performance-driven control for sample-based singing voice synthesis. In *Proc. DAFx*, volume 6, pages 42–44, 2006.

[4] Perry R Cook. Real-time performance controllers for synthesized singing. In *Proceedings of the 2005 conference on New interfaces for musical expression*, pages 236–237. National University of Singapore, 2005.

[5] Perry R Cook. Toward the perfect audio morph ? singing voice synthesis and processing. In *Proceedings of the 1st. International Conference on Digital Audio Effects (DAFX), Barcelona*, 1998.

[6] Perry R Cook. Spasm, a real-time vocal tract physical model controller ; and singer, the companion software synthesis system. *Computer Music Journal*, 17(1) :30–44, 1993.

[7] Xavier Rodet, Yves Potard, and Jean-Baptiste Barriere. The chant project : from the synthesis of the singing voice to synthesis in general. *Computer Music Journal*, 8(3) :15–31, 1984.

[8] Max V Mathews and John R Pierce. *Current directions in computer music research*. Mit Press, 1989.

[9] Xavier Rodet and Philippe DEPALLE. *High Quality Synthesis-by-Rule of Consonants*. Ann Arbor, MI : MPublishing, University of Michigan Library, 1985.

[10] Bjorn Larsson. Music and singing synthesis equipment (musse). *Speech Transmission Laboratory Quarterly Progress and Status Report (STL-QPSR)*, 1(1977) :38–40, 1977.

[11] Gunilla Berndtsson. The kth rule system for singing synthesis. *Computer Music Journal*, 20(1) :76–91, 1996.

[12] Heiga Zen, Keiichi Tokuda, and Alan W Black. Statistical parametric speech synthesis. *Speech Communication*, 51(11) :1039–1064, 2009.

[13] Marcus Uneson. Outlines of burcas-a simple concatenation-based midi-to-singing voice synthesis system. 2002.

[14] Thierry Dutoit, Vincent Pagel, Nicolas Pierret, François Bataille, and Olivier Van der Vrecken. The mbrola project : Towards a set of high quality speech synthesizers free of use for non commercial purposes. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 3, pages 1393–1396. IEEE, 1996.

[15] Michael W Macon, Leslie Jensen-Link, James Oliverio, Mark A Clements, and E Bryan George. A singing voice synthesis system based on sinusoidal modeling. In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, volume 1, pages 435–438. IEEE, 1997.

[16] Jordi Bonada, Òscar Celma, Àlex Loscos, Jaume Ortolà, Xavier Serra, Yasuo Yoshioka, Hiraku Kayama, Yuji Hisaminato, and Hideki Kenmochi. Singing voice synthesis combining excitation plus resonance and sinusoidal plus residual models. In *Proceedings of International Computer Music Conference*. Citeseer, 2001.

[17] J Bonada and A. Loscos. Sample-based singing voice synthesizer by spectral concatenation. 2003.

[18] Hideki Kenmochi and Hayato Ohshita. Vocaloid-commercial singing synthesizer based on sample concatenation. In *INTERSPEECH*, pages 4009–4010, 2007.

[19] Takayoshi YoshimuraÝ, Keiichi TokudaÝ, Takashi MasukoÝÝ, Takao KobayashiÝÝ, and Tadashi KitamuraÝ. Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis. 1999.

[20] Keiichiro Oura, Ayami Mase, Yoshihiko Nankaku, and Keiichi Tokuda. Pitch adaptive training for hmm-based singing voice synthesis. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 5377–5380. IEEE, 2012.

[21] Keiichiro Oura, Ayami Mase, Tomohiko Yamada, Satoru Muto, Yoshihiko Nankaku, and Keiichi Tokuda. Recent development of the hmm-based singing voice synthesis system - sinsy. *Proc. ISCA SSW7*, pages 211–216, 2010.

[22] sinsy website : online hmm-based singing synthesis system. http ://www.sinsy.jp/. dernier accès : 07/2013.

[23] Tuomo Raitio, Antti Suni, Junichi Yamagishi, Hannu Pulakka, Jani Nurminen, Martti Vainio, and Paavo Alku. Hmm-based speech synthesis utilizing glottal inverse filtering. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19(1) :153–165, 2011.

[24] Axel Roebel, Stefan Huber, Xavier Rodet, and Gilles Degottex. Analysis and modification of excitation source characteristics for singing voice synthesis. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 5381–5384. IEEE, 2012.

[25] Takeshi Saitou, Masataka Goto, Masashi Unoki, and Masato Akagi. Speech-to-singing synthesis : converting speaking voices to singing voices by controlling acoustic features unique to singing voices. In *Applications of Signal Processing to Audio and Acoustics, 2007 IEEE Workshop on*, pages 215–218. IEEE, 2007.

[26] Hideki Kawahara, Ikuyo Masuda-Katsuse, and Alain de Cheveigné. Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based f0 extraction : Possible role of a repetitive structure in sounds. *Speech communication*, 27(3) :187–207, 1999.

[27] Erica Bisesi, Richard Parncutt, and Anders Friberg. An accent-based approach to performance rendering : Music theory meets music psychology. In *Proceedings of ISPS 2011*, pages 27–32. the European Association of Conservatoires (AEC), 2011.

[28] Johan Sundberg. Synthesizing singing. *The Journal of the Acoustical Society of America*, 83 :S30, 1988.

[29] Johan Sundberg. The kth synthesis of singing. *Advances in Cognitive Psychology*, 2(2-3) :131–143, 2006.

[30] Robert Maher and James Beauchamp. An investigation of vocal vibrato for synthesis. *Applied Acoustics*, 30(2) :219–245, 1990.

[31] Takeshi Saitou, Masashi Unoki, and Masato Akagi. Development of an f0 control model based on f0 dynamic characteristics for singing-voice synthesis. *Speech communication*, 46(3) :405–417, 2005.

[32] PB Oncley. Frequency, amplitude, and waveform modulation in the vocal vibrato. *The Journal of the Acoustical Society of America*, 49 :136, 1971.

[33] Takeshi Saitou, Naoya Tsuji, Masashi Unoki, and Masato Akagi. Analysis of acoustic features affecting" singing-ness" and its application to singing-voice synthesis from speaking-voice. In *INTERSPEECH*, 2004.

[34] Gunnar Fant, Johan Liljencrants, and Qi-guang Lin. A four-parameter model of glottal flow. *STL-QPSR*, 4(1985) :1–13, 1985.

[35] sdif sourceforge. http ://sdif.sourceforge.net. dernier accès : 07/2013.

[36] Udo Zölzer, Xavier Amatriain, and John Wiley. *DAFX : digital audio effects*, volume 1. Wiley Online Library, 2002.

[37] Miller Puckette. Phase-locked vocoder. In *Applications of Signal Processing to Audio and Acoustics, 1995., IEEE ASSP Workshop on*, pages 222–225. IEEE, 1995.

[38] Jean Laroche and Mark Dolson. Phase-vocoder : About this phasiness business. In *Applications of Signal Processing to Audio and Acoustics, 1997. 1997 IEEE ASSP Workshop on*, pages 4–pp. IEEE, 1997.

[39] Axel Roebel. A shape-invariant phase vocoder for speech transformation. In *Proc. Digital Audio Effects (DAFx)*, 2010.

[40] Salim Roucos and Alexander Wilgus. High quality time-scale modification for speech. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'85.*, volume 10, pages 493–496. IEEE, 1985.

[41] Axel Röbel, Fernando Villavicencio, and Xavier Rodet. On cepstral and all-pole based spectral envelope modeling with unknown model order. *Pattern Recognition Letters*, 28(11) :1343–1350, 2007.

[42] Axel Röbel and Xavier Rodet. Efficient spectral envelope estimation and its application to pitch shifting and envelope preservation. In *Proc. DAFx*, 2005.

[43] Jean Laroche and Mark Dolson. New phase-vocoder techniques for pitch-shifting, harmonizing and other exotic effects. In *Applications of Signal Processing to Audio and Acoustics, 1999 IEEE Workshop on*, pages 91–94. IEEE, 1999.

[44] Michael Macon, Leslie Jensen-Link, E Bryan George, James Oliverio, and Mark Clements. Concatenation-based midi-to-singing voice synthesis. In *Audio Engineering Society Convention 103*. Audio Engineering Society, 1997.

[45] Gilles Degottex. *Glottal source and vocal-tract separation*. PhD thesis, UPMC-Ircam. France, 2010.

Annexes

Modèle source-filtre de production de la voix

Une grande partie des méthodes évoquées dans ce rapport pour synthétiser le chant s'appuient sur le modèle "source-filtre" de production de la voix. Une description rapide en est donc donnée ici.

La théorie du modèle source-filtre considère que la production de la voix humaine peut être représentée par un simple filtrage d'un signal de source d'excitation glottique par le conduit vocal. Cela peut donc être interprété comme un processus linéaire de la forme : S(z)=G(z)*V(z)*L(z) où S(z) représente la parole produite, et G(z), V(z) et L(z) représentent respectivement le signal de source, le filtre correspondant au conduit vocal, et le rayonnement au niveau des lèvres [23]. Ce modèle ne prend pas en compte explicitement le couplage entre la glotte et le conduit vocal. Un tel modèle, pour être réaliste, requiert une bonne modélisation de la source glottique. Le modèle Liljencrants-Fant (LF) [34] est couramment utilisé pour cela. Le filtre lié au conduit vocal dépend de la position des différents dispositifs articulatoires du système de production vocal, tels que la langue, le larynx, ou le voile du palet. Le rayonnement des lèvres peut être approximé par une simple dérivation du signal.

Le modèle source-filtre est présenté en détail dans de nombreuses sources, parmis lesquelles on peut citer la thèse de Gilles Degottex [45].

Commandes superVP utilisées

Dans cette section sont présentées les différentes commandes d'appel à superVP utilisées dans notre programme pour l'analyse les transformations et la synthèse et les paramètres qui les composent.
L'option -A permet de sélectionné le type d'analyse à effectuer : -Af0 pour une analyse de fréquence fondamentale, -Afft pour une analyse de type vocodeur de phase, et -Atenv pour une analyse de l'enveloppe spectrale par l'algorithme true-envelope.

L'option -S permet de spécifier le fichier source.

Les options -B et -E permettent de spécifier, respectivement, les temps de début et fin de l'analyse à effectuer dans le fichier d'entrée.

-M permet de fixer, en secondes, la taille de la fenêtre d'analyse.

-W définie le type de fenêtre à utiliser.

-Np correspond au facteur de suréchantillonage de la DFT pour augmenter la précision

L'option -oversamp détermine le pas d'avancement entre 2 trames d'analyse. Par exemple, -oversamp 4 signifie que le temps entre 2 trames est égale à 1/4 de la taille de la fenêtre d'analyse.

Le paramètre -O détermine le format du fichier de sortie (qui dépend de l'analyse effectuée).

Le nom du fichier de sortie est toujours déclaré en dernier.

Analyse de fréquence fondamentale :

"svp_float -Af0 fm50,fM450,F2000 -S "+databaseSnd+" -B"+startTime+"s -E"+endTime+"s -M0.04s -Np4 -oversamp 8 -Whanning -Of4 f0_file.sdif"

En plus des paramètres précédemment cités, on précise, pour l'analyse de fréquence fondamentale, les valeurs minimales (fm) et maximales (fM) auxquelles ont peut s'attendre. F correspond à la fréquence maximale des sinusoïdes présentes dans le signal.

Analyse de type vocodeur de phase pour les transformation et la resynthèse :

"svp_float -OS7 -S"+databaseSnd+" -B"+startTime+"s -E"+endTime+"s -Afft +150Hz -F0 f0_file.sdif -Np1 -M0.045s -oversamp 4 -Whanning -P0 -shape 1 -syntr -transmo 2 fft_file.sdif"

La valeur de F0 maximale est spécifiée par le "+150Hz", ainsi que le fichier f0_file.sdif contenant la fréquence fondamentale pour tout le fichier analysé. Cela permet de déduire au mieux l'ordre utilisé pour la séparation source-filtre lors des transpositions. L'option "-transmo 2" signifie que l'on souhaite effectuer une transposition dans le domaine fréquentiel, et l'option "-syntr" signifie que cette transposition sera effectuée à la synthèse. Le paramètre -shape 1 signifie que l'on souhaite utiliser l'algorithme SHIP pour les transformations.

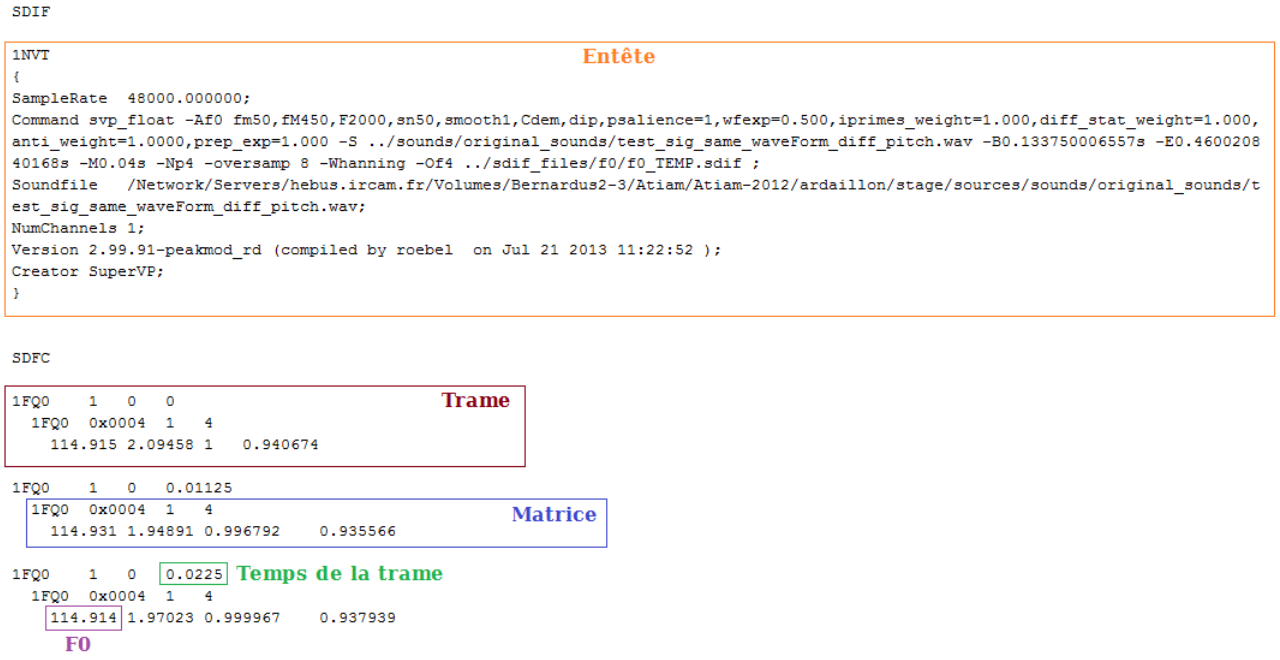
Analyse de l'enveloppe spectrale :

"svp_float -OS1 -S"+databaseSnd+" -B"+startTime+"s -E"+endTime+"s -Atenv +150Hz -F0 f0_file.sdif -Np1 -M0.045s -oversamp 4 -Whanning env_file.sdif"

Exemple de fichiers sdif au format texte

Afin de pouvoir afficher le contenu d'un fichier SDIF sous une forme lisible, le programme "sdiftotext" permet de convertir un fichier SDIF en fichier texte.

Voici ci-dessous un exemple de fichier SDIF, une fois mis sous forme textuelle. Il s'agit d'une analyse de fréquence fondamentale, choisie ici pour le peu d'informations qu'elle contient, permettant un affichage compact.



Nous pouvons observer sur cette image l'entête du fichier, contenant des informations d'ordre général sur son contenu, et les trois premières trames issues de l'analyse. L'entête contient notamment la fréquence d'échantillonage, la commande superVP ayant servie à générer le fichier, le nom et emplacement du fichier analysé, et des informations sur la version de superVP utilisée.

La signature pour la trame contenant les informations sur la fréquence fondamentale est 1FQ0. Cette trame contient une matrice de type 1FQ0, qui a son tour contient les informations concernant la fréquence fondamentale (il est donc possible d'utiliser une signature identique pour une trame et pour une matrice). Le temps de chaque trame est indiqué par une valeur, comme annoté sur la figure.

Pour chaque trame d'un fichier SDIF, il existe des matrices requises et des matrices optionnelles. Les matrices contiennent des colonnes de données. A nouveau, il existe pour les matrices des colonnes obligatoires et des colonnes optionnelles. La trame 1FQ0 contient uniquement la matric 1FQ0, obligatoire. La matrice 1FQ0 contient 4 colonnes de données, dont seule la première est obligatoire et contient la fréquence fondamentale du son analysé au temps de la trame.

Correspondance entre alphabet phonétiques LIA et SAMPA

Le tableau ci-dessous, copié depuis la documentation du programme ircamTTS, donne la correspondance enter les symboles des alphabets phonétiques Lia et SAMPA.

LIA	XSAMPA	EXAMPLES
ii	i	idiot, ami
ei	e	ému, été
ai	E	perdu, maison
aa	a	alarme, patte
oo	O	obstacle, corps
au	o	auditeur, beau
ou	u	coupable, loup
uu	y	punir, élu
EU	2	creuser, deux
oe	9	malheureux, peur
eu	@	petite, fortement
in	e~	peinture, matin
an	a~	vantardise, temps
on	o~	rondeur, bon
un	9~	lundi, brun
yy	j	piétiner, choyer
ww	w	quoi, fouine
pp	p	patte, repas, cap
tt	t	tête, net
kk	k	carte, écaille, bec
bb	b	bête, habile, robe
dd	d	dire, rondeur, chaud
gg	g	gauche, égal, bague
ff	f	feu, affiche, chef
ss	s	soeur, assez, passe
ch	S	chanter, machine, poche
vv	v	vent, inventer, rêve
zz	z	zéro, raisonner, rose
jj	Z	jardin, manger, piège
ll	l	long, élire, bal
rr	R	rond, charriot, sentir
mm	m	madame, aimer, pomme
nn	n	nous, punir, bonne
##	_	(silence marker)

Contrairement à ce qui été écrit dans ce tableau, nous utilisons un “#” pour marquer les silences dans l’alphabet SAMPA, et non un “_”.