



**HAL**  
open science

# Un modèle de contrôle pour la synthèse par fonctions d'ondes formantiques avec OM-Chant

Raphaël Foulon, Jean Bresson

► **To cite this version:**

Raphaël Foulon, Jean Bresson. Un modèle de contrôle pour la synthèse par fonctions d'ondes formantiques avec OM-Chant. Journées d'Informatique Musicale, 2013, Saint-Denis, France. hal-01156651v2

**HAL Id: hal-01156651**

**<https://hal.science/hal-01156651v2>**

Submitted on 27 Mar 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# UN MODÈLE DE CONTRÔLE POUR LA SYNTHÈSE PAR FONCTIONS D'ONDES FORMANTIQUES AVEC OM-CHANT

Raphaël Foulon, Jean Bresson

IRCAM - UMR STMS, équipe Représentations Musicales  
1, place Igor Stravinsky, Paris F-75004

## RÉSUMÉ

Récemment, un système de contrôle du synthétiseur CHANT a été implémenté dans l'environnement de composition assistée par ordinateur OpenMusic. Ce synthétiseur est basé sur le modèle de production vocale, et utilise la technique des fonctions d'ondes formantiques (FOF). Nous présentons un ensemble d'outils développés dans le but de fournir un contrôle expressif de CHANT, permettant de formaliser et spécifier des phrases et effets vocaux de manière riche et efficace. Après un rapide état de l'art sur la synthèse vocale et le contrôle de CHANT, nous présentons la modélisation que nous avons utilisée pour synthétiser une phrase vocale, et détaillons l'implémentation de nos procédures de contrôle dans OpenMusic. Nous proposons ensuite quelques exemples d'applications, réalisés lors de la création de l'opéra *Re Orso* de Marco Stroppa, pour conclure avec quelques pistes pour étendre ce travail.

## 1. INTRODUCTION

La synthèse par fonctions d'ondes formantiques (FOF), et en particulier le synthétiseur CHANT, ont connu un essor dans les années 1980, pour ensuite être marginalisés, progressivement éclipsés par d'autres techniques de synthèse telles que la synthèse additive ou les systèmes à tables d'ondes. Basée sur le modèle de la production vocale, cette technique permet la production de timbres voisés de très haute qualité, et peut être aisément étendue à tout autre type de sons. Un système de contrôle de CHANT a été ré-implémenté dans l'environnement de CAO<sup>1</sup> OpenMusic, permettant de structurer les différents paramètres de la synthèse (fréquence fondamentale, ainsi que la fréquence centrale, l'amplitude et la largeur de bande de chaque formant) qui seront transmis au synthétiseur.

Le système CHANT a pour spécificité de présenter une logique de contrôle continue : contrairement à des protocoles plus courants comme CSound ou MIDI, celui-ci s'effectue via un ensemble de paramètres spécifiés à des instants donnés, qui seront interpolés temporellement avant d'être intégrés dans un processus de synthèse. Les protocoles de contrôle définis dans OpenMusic, basés sur la notion d'événements (éléments atomiques du contrôle, localisés et étendus dans le temps), donnent cependant au

contrôle un caractère discret, plus naturel dans les environnements symboliques et les processus compositionnels.

La contribution que nous présentons ici concerne des procédures de contrôle pour le synthétiseur CHANT, permettant de spécifier avec précision et expressivité les comportements du système lors de la transition entre le domaine discret (OpenMusic) et (pseudo) continu (CHANT). Il s'agit d'élaborer un ensemble de « règles », qui permettront de générer une suite de paramètres correspondant à un effet désiré (production d'une consonne, d'un vibrato...); chaque règle ayant pour but de formaliser un aspect spécifique du processus de synthèse, sur une certaine durée.

Cependant, notre but ne se réduit pas à la reproduction du jeu d'un chanteur : nous chercherons entre autres à nous inspirer des modes de production de la voix dans le but de « dépasser » les propriétés traditionnelles de la synthèse vocale.

## 2. ÉTAT DE L'ART

### 2.1. CHANT et les fonctions d'ondes formantiques

La synthèse par FOF s'inspire du modèle de production de la voix, et permet de reproduire – par exemple – le comportement des résonateurs de l'appareil vocal [8]. Cette méthode de synthèse consiste à générer un ou plusieurs trains de sinusoides amorties, à une fréquence donnée (fréquence fondamentale, ou  $f_0$ ) pouvant être paramétrée. Chaque train de sinusoides contribue au spectre sonore par un *formant*, correspondant à une zone spectrale riche en intensité. Dans le synthétiseur CHANT, implémenté par Xavier Rodet et son équipe [9], il est possible de contrôler cette technique à partir de nombreux paramètres, tels que les fréquences centrales des formants, leur amplitude, leur largeur de bande, temps d'excitation, phase, etc. Dans l'implémentation initiale de la synthèse par FOF, le synthétiseur était associé à l'environnement Formes [10], qui permettait de générer des « phrases » par l'évaluation de processus temporels déterminant les valeurs des paramètres de synthèse. Les processus ainsi créés avaient également pour vocation d'activer et de paramétrer un ensemble de règles codées dans le synthétiseur, permettant d'obtenir des vibratos, de corriger le spectre du signal vocal [1], ou encore de créer des « consonnes » entre les voyelles synthétisées [11].

1. Composition assistée par ordinateur.

## 2.2. Production de phonèmes

Parmi les techniques permettant de simuler la voix, et plus particulièrement la voix chantée, nous pouvons citer la synthèse par concaténation de diphones, implémentée dans le programme *Diphone Studio* [12], qui propose de construire des phrases vocales en effectuant des morphings entre plusieurs phonèmes. Une autre approche est l'utilisation de règles élaborées suite à l'étude du fonctionnement des comportements des signaux vocaux, qui permettent de les reproduire au sein d'un processus de synthèse. Le synthétiseur MUSSE (Music and Singing Synthesis Environment) en est un exemple, proposant d'effectuer une conversion texte vers chant, tout en introduisant de nombreuses règles (vibrato, consonnes, intonation) [13].

## 2.3. Contrôle de CHANT dans OpenMusic

Dans un précédent article, deux bibliothèques permettant le contrôle de la synthèse « chant » dans OpenMusic ont été présentées [5] ; la première, *chant-lib*, permettant de reproduire le comportement du synthétiseur à partir d'un modèle basé sur la concaténation d'évènements implémentés dans le langage Csound, et la seconde, *OM-Chant*, utilisant le mode de contrôle « continu » du synthétiseur. Plus récemment, *OM-Chant* a été étendue afin d'intégrer la notion d'évènements définis grâce à des objets musicaux de haut niveau tels que des matrices de paramètres [6]. Ces évènements de synthèse peuvent être des valeurs de fréquence fondamentale ( $f_0$ ), des matrices de paramètres de FOF, des générateurs de bruit ou des filtres formantiques.

Dans ce contexte, qui est le point de départ de notre travail, les comportements pouvant apparaître en cas d'évènements distants, contigus ou superposés restent à concevoir.

## 3. PRODUCTION D'UNE PHRASE VOCALE

Avant d'implémenter de nouvelles procédures de contrôle, nous avons effectué une étude des phénomènes de production vocale, à partir d'analyses temps-fréquence d'extraits audio. Les analyses ont été effectuées grâce au logiciel *Audiosculpt* [3], en utilisant la méthode LPC<sup>2</sup>. Les extraits audio analysés proviennent de sources variées : bases de données de phonèmes, enregistrements d'opéra, ou encore extraits sonores de la pièce *Chréode* de Barrière [2], produits grâce à une précédente implémentation de CHANT, qui témoignent d'une utilisation originale du synthétiseur.

### 3.1. Modèle utilisé

Nous partons d'une modélisation élémentaire de la production vocale, ne considérant que sa composante voisée.

2. Linear Predictive Coding.

Nous dégageons deux phénomènes dans ce modèle<sup>3</sup> : les états stables et les transitions. Le critère de discrimination utilisé pour définir les deux phénomènes est basé sur le comportement des formants. Le sonagramme visible sur la figure 1 présente un exemple sur lequel ces deux états sont identifiés.

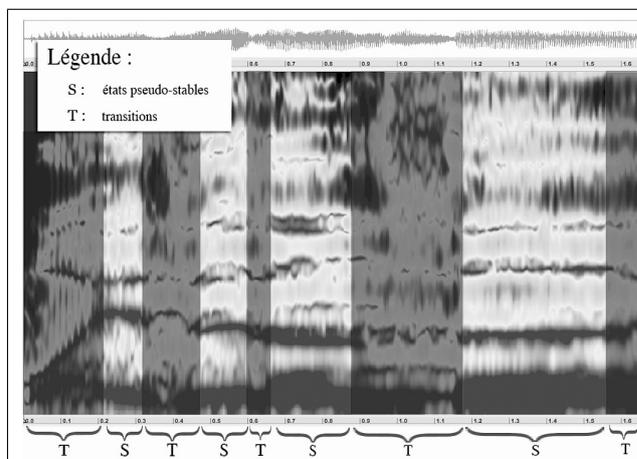


Figure 1. Analyse d'un extrait de voix chantée, identification des états stables (S) et des transitions (T).

Nous définissons un état stable (plus précisément, *pseudo-stable*) comme un état continu dans le temps, au sein duquel les paramètres formantiques n'évoluent pas ou peu : les variations ne doivent pas excéder 10% pour les fréquences centrales, 3dB pour les amplitudes et 20% pour les largeurs de bande. Dans le cadre de la voix parlée, les états stables peuvent être assimilés aux voyelles, ou dans le cas du chant lyrique, aux notes tenues. Cependant, les parties centrales de certaines consonnes voisées, telles que le 'm' ou le 'l', pourront également être modélisées par un état stable, du fait de la faible variation des formants en leur centre. Un exemple sera évoqué dans la partie 3.2.

Pendant l'étape de synthèse sonore, ces états pourront être produits par des FOF aux paramètres formantiques constants. Cependant, une certaine quantité de variations doit être introduite dans ces paramètres pour donner du « caractère » à la voix, de sorte que le rendu sonore ne paraisse pas artificiel. Deux solutions ont été abordées et testées avec succès : la variation de  $f_0$  (modulée aléatoirement ou par un vibrato<sup>4</sup>), et la variation aléatoire des paramètres de FOF.

Par opposition aux états stables, les transitions sont les parties du spectre dans lesquelles les formants présentent une évolution significative. Elles peuvent être assimilées aux consonnes (bien que cela soit réducteur lorsqu'on prend en compte les consonnes voisées), ou aux articulations en chant lyrique (staccato, legato, etc.).

Alors que les états stables peuvent être représentés par des valeurs de paramètres de FOF, les transitions seront

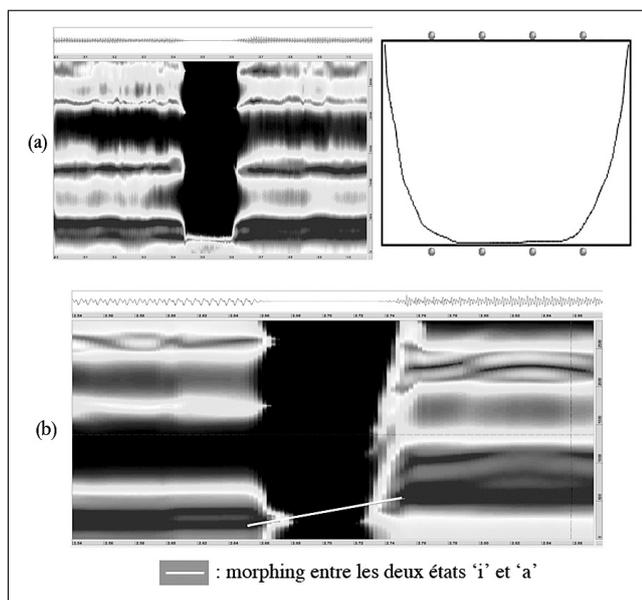
3. Ce modèle s'inspire des travaux effectués lors de l'implémentation de règles pour le contrôle de CHANT dans l'environnement Formes [11].

4. Cette solution est particulièrement adaptée au chant d'opéra.

définies à l'aide de ce que nous appellerons *trajets formantiques*. Ces trajets permettent de définir le comportement temporel des fréquences, amplitudes, largeurs de bande (voire d'autres paramètres, tels que le temps d'excitation) entre deux états stables.

Dans le but de modéliser des phénomènes vocaux (consonnes, articulations...), nous avons élaboré un certain nombre de trajets formantiques, à partir d'observations de signaux. Les phénomènes ainsi modélisés pourront être reproduits au sein d'un processus de synthèse.

Afin d'illustrer ce concept de trajet formantique, nous allons décrire la procédure de contrôle mise en place lors de la reproduction de la consonne 'b', dans la syllabe 'iba'. Pour simplification, seul le paramètre de la fréquence centrale du premier formant sera pris en compte ici.



**Figure 2.** (a) de gauche à droite : spectrogramme de la syllabe 'əbə', et trajet formantique s'appliquant à la fréquence centrale du premier formant dans la transition 'b'. (b) spectrogramme de la syllabe 'iba', obtenue par synthèse après application des différents trajets formantiques.

Le processus de reproduction de consonne peut être décrit ainsi : dans un premier temps, un trajet formantique est modélisé à partir de l'analyse d'un phonème, comme celui visible sur le spectrogramme de la figure 2 (a). Dans cet exemple, le trajet formantique est élaboré à partir d'analyses de consonnes 'b', et en particulier de l'observation du comportement de l'amplitude du premier formant (qui suit une courbe convexe). Puis, nous effectuons un *morphing* (par interpolation linéaire) à partir de deux états stables jouxtant la transition à créer (par exemple, les deux voyelles 'i' et 'a'). On applique enfin à ce morphing (par multiplication) le trajet formantique subi par l'amplitude du premier formant. Pour modéliser correctement la transition, il faut en réalité répéter ce processus pour chaque formant, et pour chacun de ses paramètres. Le sonagramme du résultat est en partie visible sur la figure 2(b). Le procédé est similaire pour la création d'une articulation type

staccato/legato : modéliser les trajets effectués par les amplitudes des différents formants est suffisante pour un obtenir un rendu réaliste.

Pour obtenir un contrôle plus fin sur le trajet formantique, nous introduisons un paramètre supplémentaire : la courbe de morphing, qui permet d'effectuer au début de la procédure une interpolation non linéaire entre deux états-stables, pour plus de réalisme. Nous appellerons par la suite l'ensemble (trajets formantiques / courbe de morphing) le *profil formantique* d'une transition.

Définir les différents phénomènes vocaux (syllabes, groupes phonème-articulation, etc.) par leur seul profil formantique fait la force de cette méthode : elle permet d'aborder le problème du contrôle de la synthèse de manière modulaire, contrairement aux techniques de concaténation de diphones, dans laquelle une consonne est définie de manière statique, en fonction du triphone<sup>5</sup> dans laquelle elle est présente. Notons cependant qu'une approximation a été faite dans ce modèle : il a été démontré que dans le cas de la voix parlée, la production d'une consonne peut dépendre de la voyelle précédant le triphone concerné [11].

### 3.2. Cas particuliers liés à la synthèse vocale

Comme nous l'avons présenté plus tôt, on ne peut classer les phonèmes<sup>6</sup> strictement en états stables et en transitions, et des cas particuliers sont à aborder.

L'exemple du 'm'<sup>7</sup> peut s'appliquer à d'autres phonèmes tels que le 'n' ou le 'l', ou même le 'ʒ' ou le 'z', si l'on ne considère que leurs composantes voisées. Une première manière de modéliser la consonne est d'utiliser une unique transition. Cependant, si le 'm' dure plus de 200ms, le trajet formantique se trouvera trop étiré dans le temps, et le rendu sonore pourra être jugé artificiel. Une autre modélisation, plus fine, consiste à modéliser la consonne grâce à deux transitions, qui formeront le début et la fin du 'm', et d'un état stable, qui formera sa partie centrale. Ainsi, on pourra par exemple obtenir un rendu de meilleure qualité pour un 'm' étendu dans le temps (de plusieurs secondes).

Par souci de concision, nous n'allons pas détailler d'autres cas particuliers, mais nous pouvons évoquer :

- les débuts et fins de phrases : ils pourront être représentés par des états stables ayant des paramètres d'amplitude nuls (silencieux), suivis d'une transition.
- le 'r' « roulé » : il s'agit d'un son pseudo-périodique, que l'on peut modéliser avec un pattern 'état stable-transition' répété, que l'on verra plus en détail dans la section 6.1.

## 4. IMPLÉMENTATION DANS OM-CHANT

La bibliothèque OM-Chant propose plusieurs objets, dotés d'attributs temporels tels qu'une date de début et une durée, représentant des événements de contrôle pour

5 . Un triphone est un triplet voyelle-consonne-voyelle.

6 . On parle ici de 'phonèmes' au sens donné par la phonétique.

7 . 'm' est une consonne occlusive nasale bilabiale voisée.

le synthétiseur CHANT. Ces objets permettent de formaliser les processus de synthèse, à partir d'événements de synthèse dissociés et contrôlant différents paramètres, tels que des valeurs de  $f_0$ , représentées par des objets *ch-f0*, et des paramètres formantiques (matrices de  $n$  formants), représentés par des objets *ch-FOF*. Ces différents objets permettent de régler les paramètres de synthèse et leur évolution dans le temps. Lorsque deux événements se suivent (qu'ils soient de type *ch-f0* ou *ch-FOF*, une interpolation linéaire est effectuée entre les valeurs des paramètres à la fin du premier et au début du second événement. Pour plus de précisions concernant le contrôle de CHANT dans l'environnement OpenMusic, on pourra se référer à [6].

#### 4.1. Génération des transitions

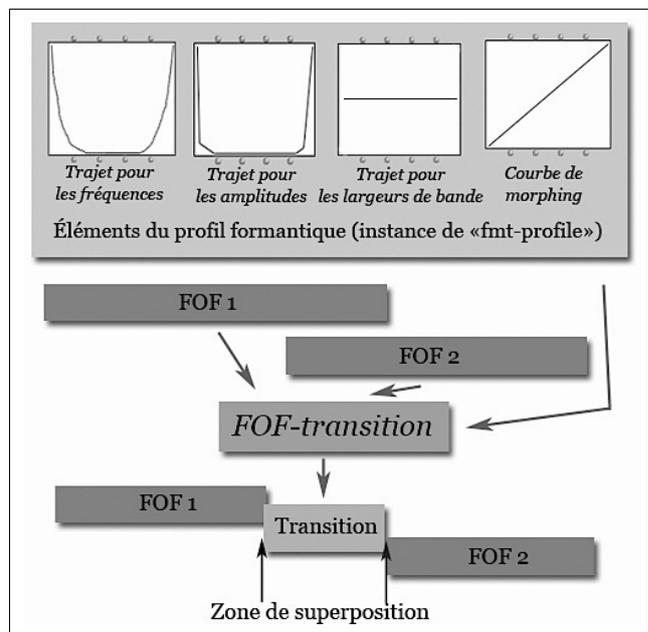
Nous avons implémenté une série d'outils permettant de traiter les cas d'événements distants, en les considérant comme des états stables, et générant les transitions qui les séparent. Dans la bibliothèque OM-Chant, la fonction *ch-transitions* permet de gérer et de transformer une séquence d'événements, en appliquant une fonction de manière itérative (cette fonction lui étant donnée en argument). C'est en se basant sur ce système qu'une première fonction, *FOF-transition*, a été implémentée. Cette fonction prend en entrée la définition du profil formantique de la transition désirée, ainsi que deux événements de type *ch-FOF*, les transforme si besoin est, et insère entre les deux un nouvel événement de transition (également du type *ch-FOF*). Ainsi, utilisée comme fonction-lambda avec *ch-transitions*, elle permet de gérer et transformer une séquence d'événements *ch-FOF* en insérant des événements transitoires.

La définition du profil formantique donné en paramètre se fait par l'intermédiaire de la classe *phoneme*. Celle-ci est également une matrice, permettant de stocker des courbes, correspondant respectivement aux trajets effectués par chacune des fréquences, amplitudes et largeurs de bande des formants (et extensible à d'autres paramètres). Est aussi stockée la courbe de morphing entre les deux états stables.

Sur la figure 3 se trouve un exemple de traitement d'une transition avec *FOF-transition*, sous forme schématisée.

Du point de vue algorithmique, *FOF-transition* effectuée dans un premier temps le « morphing » (ou interpolation) entre les paramètres des deux FOF passées en entrée, puis échantillonne les trajets formantiques avant de les multiplier au résultat.

Ce procédé pose le problème du taux d'échantillonnage : quelle granularité appliquer à l'événement de transition ? Nous avons introduit un paramètre optionnel permettant de régler la période d'échantillonnage de la transition. Par défaut, celle-ci est de 1ms. En effet, il s'est avéré qu'une fréquence d'échantillonnage de 1000Hz offrait un rendu de qualité dans le cas de consonnes et d'articulations communes de type staccato, legato, avec un temps de calcul négligeable (de l'ordre de quelques millisecondes pour le rendu d'une seconde de trajet formantique).



**Figure 3.** Illustration du fonctionnement de *FOF-transition*, qui prend en entrée deux événements « FOF », et insère une transition à l'endroit où ils se superposent, conformément au profil formantique de la consonne 'b' (simplifié, tous les formants ayant le même profil) passé en paramètre.

Une fonction analogue, *f0-transition*, permet de définir le trajet effectué par la fréquence fondamentale entre deux événements de type  $f_0$ . Le trajet est formalisé par une courbe, qui vient multiplier l'interpolation linéaire effectuée entre les deux objets *ch-f0* qui jouxtent la transition.

#### 4.2. Dictionnaire de transitions

La gestion des différentes définitions de profils formantiques est centralisée, via l'utilisation d'un dictionnaire. Il s'agit d'un patch OpenMusic associant à une liste d'identifiants (clés<sup>8</sup>), représentés par des chaînes de caractères, une liste d'objets de type *phoneme*. Il devient alors possible de définir un dictionnaire de transitions, qui peut être étendu à volonté par l'utilisateur.

Le dictionnaire peut aussi avoir un rôle créatif, au delà de celui de « base de données » : il est possible de faire la requête d'un profil formantique hybride, en sélectionnant deux clés correspondant à deux profils distincts, ainsi qu'une valeur de pondération permettant de calculer un profil intermédiaire entre ceux-ci.

#### 4.3. Transitions paramétrées

Afin d'introduire un aspect « dynamique » dans notre système de transitions, nous sommes partis du cas pratique suivant : comment créer une phrase contenant des articulations évoluant progressivement du *legato* au *staccato*, sans

8 . « Clés primaires », pour utiliser le vocabulaire des bases de données.

avoir à modéliser chaque articulation intermédiaire entre les deux types de jeu ? Nous avons alors mis en place un mécanisme permettant de donner un paramètre en entrée lors de la création d'un profil formantique, pour permettre à *phoneme* d'être instancié dynamiquement. Ainsi, il devient alors possible d'imaginer une transition paramétrée modélisant le staccato, le legato, ainsi que l'ensemble des profils intermédiaires entre ces deux articulations. Dans cet exemple, une valeur de 0 donnée en entrée créera un profil de staccato très net, 1 un profil de portamento, et les valeurs comprises dans ]0 ; 1[ un profil correspondant à une articulation intermédiaire. Un exemple d'application est donné dans la partie 6.2.

#### 4.4. Morphing formantique

En parallèle, nous avons implémenté une procédure permettant de générer des états hybrides entre stabilité et transition. Bien que cela n'entre pas directement dans le cadre de notre modélisation de départ, cette procédure permet d'étendre les possibilités compositionnelles offertes par le système.

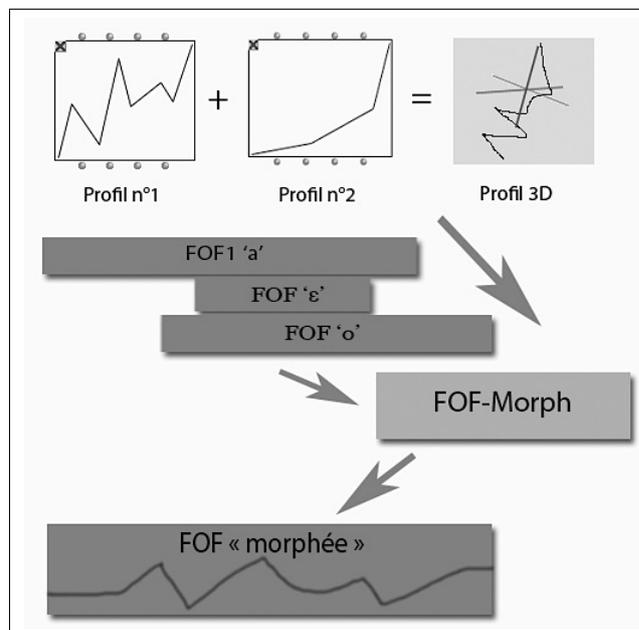
Nous appellerons cette procédure le « morphing formantique ». En effet, elle permet de prendre en entrée plusieurs états stables, et de pondérer leurs paramètres, afin d'obtenir un état hybride. La fonction *FOF-morph* a été implémentée selon ce principe et permet, à partir d'un ensemble d'événements FOF et d'une courbe, d'effectuer une pondération dans le temps entre les événements superposés, suivant un profil donné. En fonction du nombre d'événements, une courbe supplémentaire peut être spécifiée : le premier profil sera utilisé si deux FOF se superposent, le second si une troisième FOF est donnée (on parlera alors de « morphing 3D »). Une illustration de ce mécanisme est proposée sur la figure 4.

Cette procédure de morphing peut servir, entre autres, à modéliser des chants diphoniques.

### 5. ENVIRONNEMENT GRAPHIQUE

Pour proposer à l'utilisateur un environnement de composition plus efficace et intuitif, nous avons intégré ces procédures de contrôle aux *maquettes* OpenMusic. Les maquettes sont des interfaces graphiques permettant de représenter et d'agencer des structures musicales dans le temps. Elles sont composées d'une interface principale, comprenant une « timeline », sur laquelle il est possible de déposer ou de créer des patches ; ainsi qu'un patch interne d'évaluation [4]. Ce dernier permet de définir le processus produisant le résultat de la maquette (dans notre cas, un rendu sonore), après évaluation des patches contenus dans celle-ci.

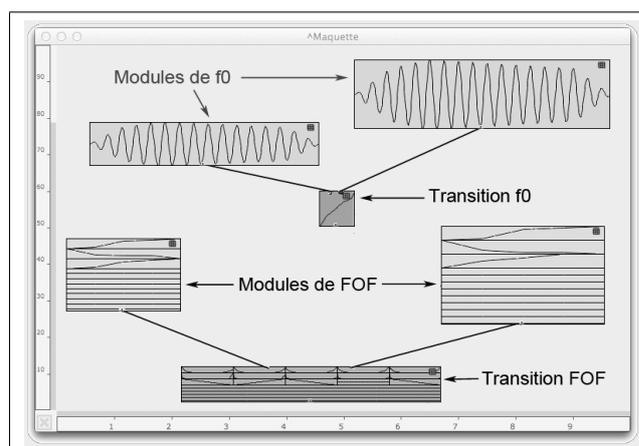
Grâce aux maquettes, il devient possible de représenter symboliquement les éléments de contrôle sur un espace 2D, et d'éditer graphiquement les phrases musicales. Concrètement, l'utilisateur peut créer et manipuler divers modules (patches) représentant des événements, qui sont classés dans différentes catégories :



**Figure 4.** Illustration du fonctionnement de *FOF-morph*, en effectuant une pondération entre trois états stables superposés, représentant les phonèmes 'ə', 'a' et 'o', en suivant deux profils (représentés en vue 3D).

- modules de FOF, qui représentent les états stables ;
- modules de f0 (avec ou sans vibrato) ;
- modules de transition (f0 et FOF) ;
- modules de morphing de FOF.

Un exemple d'utilisation est donné sur la figure 5. On peut observer sur cette figure que la phrase vocale a été construite en connectant différents modules : les modules de FOF sont connectés en entrée à un module de transition, de même que les modules de f0. Ainsi, les transitions sont générées automatiquement en fonction des états stables lors de l'évaluation de la maquette.



**Figure 5.** Maquette contenant deux modules de f0 avec vibrato et un module de transition f0 (en haut), ainsi que deux modules de FOF séparés par un module de transition FOF (en bas). Les modules de transition ont été légèrement déplacés pour rendre visibles les connexions entre modules.

L'ensemble des objets produits par évaluation des différents modules sont de type homogène : le résultat d'un morphing ou d'une transition entre deux modules de FOF sera un objet du même type que celui produit par l'évaluation d'un module de FOF simple. Cela permet de définir des événements complexes tels que des morphings de morphings, ou des morphings de transitions...

Nous pouvons aussi noter que le système de base de données de phonèmes, qui vient enrichir le concept de dictionnaire de transitions évoqué en partie 4.2, permet de concevoir des « banques de chanteurs », qui contiennent les articulations ainsi que les phonèmes propres aux formants et au style de jeu d'un chanteur virtuel.

Pour accélérer la création d'une phrase musicale, nous avons implémenté plusieurs fonctions permettant de remplir automatiquement une maquette avec des modules de FOF et de f0. Ces fonctions prennent en entrée :

- une structure temps fréquence <sup>9</sup>, représentant les différentes notes jouées (on peut par exemple partir d'un objet *chord-seq* <sup>10</sup>).
- une suite de symboles, désignant les clés pointant vers des modules d'états stables et de transitions.
- une base de données de phonèmes, effectuant la relation clés/modules de FOF (instance de la classe *phoneme-DB*).
- une liste spécifiant les durées des différentes transitions.

Après analyse des entrées, et vérification de leur cohérence (rapport entre le nombre d'états stables et le nombre de notes, etc.), ces fonctions génèrent les modules qui viendront remplir une maquette cible. D'autres traitements particuliers sont effectués, comme dans le cas de suites de consonnes, où un état stable de courte durée (20 ms) est ajouté entre chaque « consonne ». Un exemple est visible sur la figure 6, et le résultat de son évaluation, sous forme de maquette, sur la figure 7.

Il devient alors possible de travailler à deux niveaux : à partir de structures musicales (*chord-seq*, « séquences de phonèmes »), en travaillant à partir de données musicales précises ; et directement à partir de la maquette, en agencant de manière intuitive des événements de contrôle dans le temps.

## 6. APPLICATIONS MUSICALES

L'introduction de ces nouvelles procédures de contrôle dans l'environnement OpenMusic a permis de créer des processus musicaux originaux. Nous allons détailler deux d'entre eux, qui ont été utilisés à l'occasion de la création de l'opéra *Re Orso* de Marco Stroppa. <sup>11</sup>

9. Structure sous la forme d'une liste d'éléments (fréquence, date, durée).

10. Dans OpenMusic, les instances de la classe *chord-seq* permettent de représenter des suites de notes ou d'accords.

11. Création en mai 2012 à l'Opéra Comique de Paris.

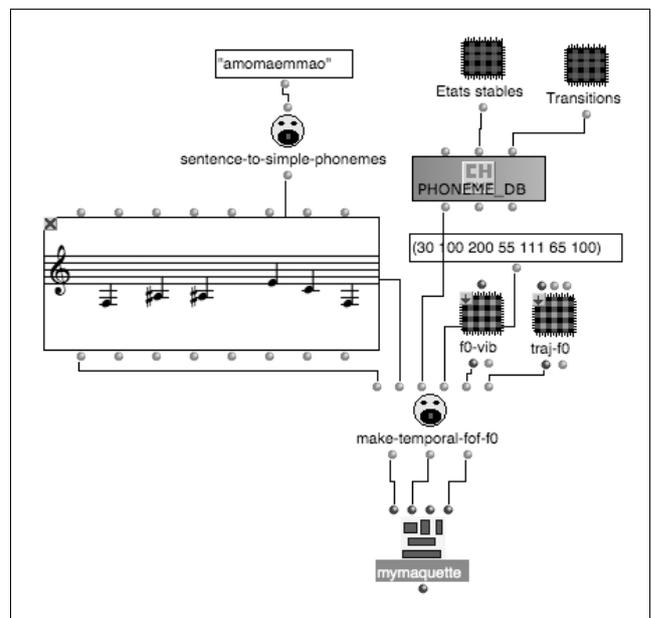


Figure 6. Patch illustrant le remplissage automatique d'une maquette à partir d'un objet *chord-seq*, d'une base de données de phonèmes, d'une chaîne de caractères et d'une liste de durées de transition.

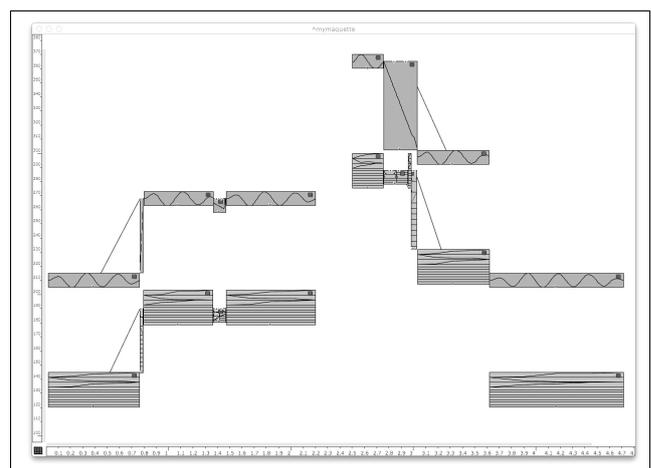


Figure 7. Maquette remplie automatiquement après évaluation du patch de la figure 6.

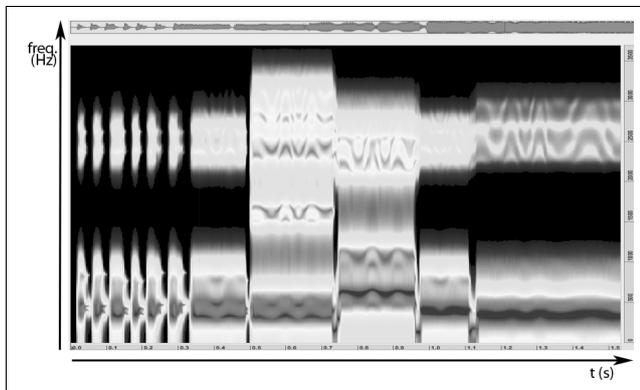
### 6.1. Roulement de 'r' à période variable

Contrairement au 'r' tel qu'on le prononce en français <sup>12</sup>, qui peut être synthétisé à l'aide de variations aléatoires agissant sur les différents paramètres formantiques, le 'r' roulé présente un motif périodique. En effet, en fonction de la morphologie de l'appareil vocal du chanteur, la langue vient frapper le palais à une certaine fréquence <sup>13</sup>. Au regard de cette particularité, nous avons choisi de modéliser le 'r' roulé à l'aide d'une suite de transitions, chacune correspondant à une période, soit à un claquement de

12. Il s'agit, en phonétique, d'une consonne *uvulaire*.

13. La moyenne se situe approximativement autour de 20Hz, cependant, il existe des individus capables de rouler les 'r' à des fréquences dépassant les 30Hz.

langue. Un ensemble d'états stables courts permettra de séparer les différentes transitions. Il devient alors possible de paramétrer la durée de chaque claquement, et donc d'effectuer des *accelerando* ou des *rallentendo* au niveau des roulements, chose impossible à effectuer pour un chanteur. Une analyse temps fréquence du résultat d'un tel processus est visible sur la figure 8.



**Figure 8.** Analyse temps fréquence d'un roulement de 'r', synthétisé par CHANT, dont la fréquence décroît au cours du temps

Ce processus de synthèse est un exemple de ce qu'offre cette implémentation de CHANT en termes de créativité : il devient possible de dépasser et de détourner le modèle de production de la voix.

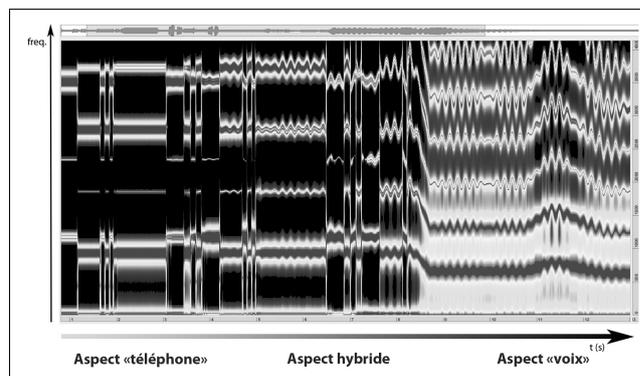
## 6.2. Morphings timbraux

Un autre exemple, tiré lui aussi de *Re Orso*, est celui du morphing entre un matériau « non-organique » et une voix. Nous avons ici voulu créer une phrase de quelques secondes au sein de laquelle une sonnerie de téléphone (sonnerie monophonique et monotimbrale) est transformée progressivement en une voix de chanteur lyrique. Pour cela, nous avons travaillé à partir de plusieurs éléments :

- les paramètres des états stables : en effectuant une interpolation entre les paramètres de formants reproduisant le timbre de téléphone, et ceux issus de l'analyse de la voix d'une chanteuse d'opéra <sup>14</sup>.
- la nature des articulations, en passant progressivement d'un staccato pur à un legato (en utilisant des transitions paramétrées).
- la durée des transitions, en partant de durées inférieures à 25 ms, pour aboutir à des durées supérieures à 50 ms.
- le vibrato, nul au début et dont l'amplitude augmente en fonction du temps (à noter que le vibrato suit une enveloppe sur chaque note).

Sur la figure 9 se trouve l'analyse spectrale du rendu sonore d'un tel processus, sur laquelle on peut observer l'évolution de ces paramètres.

14. Analyse d'un extrait de « La Reine de la Nuit », Mozart, interprétation de Erika Miklósa.



**Figure 9.** Analyse spectrale du rendu sonore du processus « téléphone vers voix », identification des trois « aspects » sonores (timbres).

## 7. CONCLUSION, PERSPECTIVES

Nous avons introduit dans l'environnement OpenMusic plusieurs méthodes permettant l'élaboration de procédures de contrôle originales du synthétiseur CHANT, en partant d'un modèle de base pour la voix chantée, que nous avons étendu afin de maximiser les capacités expressives du système. Deux mécanismes ont été mis en place : le premier permet de gérer une séquence d'événements en les traitant deux à deux ; un second permet de manipuler les événements de synthèse de manière intuitive grâce à une interface graphique. Ces outils sont intégrés dans OM-Chant et documentés dans le manuel utilisateur en ligne de la bibliothèque : <http://support.ircam.fr/docs/om-libraries/om-chant/>. De plus, des rendus sonores issus des processus décrits dans cet article sont disponibles : <http://repmus.ircam.fr/foulon/om-chant-exemples>.

Afin de terminer notre modèle de production vocale, il faudrait effectuer un travail en profondeur sur l'ajout de composantes bruitées. Ces fonctionnalités n'ont pas été proposées : nous nous sommes concentrés sur l'étude des sons voisés, qui constitue à elle seule un sujet de recherche très vaste. Cependant, les procédures proposées par notre système pourront intégrer le contrôle de la synthèse de sons non-voisés, notamment grâce aux modules de génération de bruit et de filtrage disponibles.

La synthèse par FOF est dotée d'un potentiel créatif très large, notamment de par la richesse des gammes de sons qu'elle est capable de produire. Si nous nous sommes souvent limités à la production de signaux inspirés de la voix chantée, des procédures de contrôle permettant de reproduire (et de détourner !) d'autres types de sons pourront à l'avenir être imaginées. Parmi eux, certains sons instrumentaux qui se prêtent bien à la synthèse par formants (percussions, cloches, cuivres...) ou encore d'autres sons issus de techniques originales, telles que la synthèse granulaire [7], ou les synthèses hybrides <sup>15</sup>.

15. Nous avons travaillé sur un mélange entre synthèse par FOF et synthèse additive, en modélisant un glas à l'aide de 171 formants, pour lui faire subir un morphing timbral de manière analogue qu'en partie 6.2.

## 8. REFERENCES

- [1] Baisnée P.-F. and the CHANT Group, « CHANT Manual », IRCAM, 1985.
- [2] Barrière J.-B., « Chréode, un chemin vers une nouvelle musique avec l'ordinateur », *IRCAM : une pensée musicale*, ed. Tod Machover, InterEditions, 1984.
- [3] Bogaards N., Röbel A., « An interface for analysis-driven sound processing », *AES 119th Convention*, New York, USA, 2004.
- [4] Bresson J., Agon C., « Temporal Control over Sound Synthesis Processes », *Proc. Sound and Music Computing Conference*, Marseille, France, 2006.
- [5] Bresson J., Michon, R., « Implémentations et contrôle du synthétiseur CHANT dans OpenMusic », *Actes des Journées d'Informatique Musicale*, Saint-Etienne, France, 2011.
- [6] Bresson J., Stroppa, M., « The Control of the CHANT Synthesizer in OpenMusic : Modelling Continuous Aspects in Sound Synthesis », *Proc. Int. Computer Music Conference*, Huddersfield, Royaume Uni, 2011.
- [7] Liuni M., Gentilucci M., « Multifog : a Multi-Level Control Device for FOG Granular Synthesis in Max/MSP », *Computer Music Modeling and Retrieval*, Malaga, Espagne, 2010.
- [8] Rodet X., « Time-domain Formant-wave Function Synthesis », *Computer Music Journal*, 8(3), Autumn 1984.
- [9] Rodet X., Potard Y., Barrière J.-B., « The CHANT Project : From the Synthesis of the Singing Voice to Synthesis in General », *Computer Music Journal*, The MIT Press, 8(3), 1984.
- [10] Rodet X., Cointe P., « Formes : Composition and Scheduling of Processes », *Computer Music Journal*, 8(3), Autumn 1984.
- [11] Rodet X., Depalle P., « High quality synthesis-by-rule of consonants », *Proc. International Computer Music Conference*, Vancouver, Canada, 1985.
- [12] Rodet X., Lefevre A., « The Diphone Program : New Features, New Synthesis Methods and Experience of Musical Use », *Proc. Int. Computer Music Conference*, Thessaloniki, Grèce, 1997.
- [13] Sundberg J., « The KTH synthesis of singing », *Advances in Cognitive Psychology*, Versita, 2006.