

# LA COMPOSITION ÉLECTROACOUSTIQUE POUR INTERFACE INVENTÉE

Martin Marier

Université de Montréal

Centre interdisciplinaire de recherche en musique, médias et technologie (CIRMMT)

## RÉSUMÉ

L'auteur explique son approche de la composition et de l'interprétation de la musique électroacoustique. Il cherche à retrouver le côté ludique du jeu instrumental et à améliorer l'interaction avec le public en utilisant une nouvelle interface musicale pour interpréter ses oeuvres. L'interface qu'il a conçue s'appelle l'éponge et est, en quelque sorte, un coussin muni de capteurs qui en détectent les déformations. Les stratégies de *mapping* qui ont été développées sont décrites. Les difficultés et enjeux de la composition électroacoustique pour nouvelles interfaces sont abordés et discutés.

## 1. INTRODUCTION

Dans les années '40, les nouvelles technologies permirent à Pierre Schaeffer de découvrir la musique concrète [13], celle qu'on appela plus tard musique acousmatique ou musique électroacoustique. Il était désormais possible de composer des musiques qui n'agençaient plus uniquement les sons instrumentaux, mais aussi tous les sons susceptibles d'être enregistrés ou synthétisés.

Cet art des sons fixés a donné naissance à l'art de la diffusion qui est toujours pratiqué aujourd'hui. Dans la salle de concert, où une multitude de haut-parleurs sont disposés, l'interprète spatialise le son, mais n'est responsable ni de sa génération, ni du déroulement temporel. C'est la fixation du son sur support qui, à l'origine, rendait tous les sons accessibles au musical. Mais de nos jours, la technologie permet une manipulation en direct non seulement de l'espace et de la dynamique, mais aussi d'une multitude d'autres paramètres du son. Il serait donc possible d'interpréter réellement la musique électroacoustique ; de la jouer de la même façon qu'une pièce instrumentale.

Mais l'accès à cette technologie ne règle pas tous les problèmes. Il faut aussi disposer d'une interface gestuelle qui permettra un contrôle expressif de tous ces paramètres, et c'est là le coeur de ma recherche. À cet effet, j'ai conçu une interface musicale que j'ai appelée l'éponge.

Afin de mieux comprendre ce qu'est l'expressivité et comment il est possible de l'obtenir, j'ai développé les notions de « contrôle musical » et de « contrôle paramétrique » que je décrirai plus précisément à la section 2. Je présenterai ensuite l'interface musicale que j'ai conçue et fabriquée : l'éponge. Après ce survol technique, j'élaborerai les différents enjeux et difficultés associés à cette

approche : la pérennité des oeuvres, la multidisciplinarité et surtout, les stratégies qui permettent d'établir un rapport geste-son clair. Je parlerai ensuite des compositions qui ont été écrites pour l'éponge et, pour finir, les avenues possibles pour le futur seront présentées.

## 2. CONTRÔLE PARAMÉTRIQUE VS CONTRÔLE MUSICAL

Au cours de mes recherches, j'ai développé les notions de « contrôles musical » et de « contrôle paramétrique ». Je les définies comme étant les deux extrémités d'un continuum sur lequel pourrait se trouver une grande variété de types de contrôle.

Le contrôle musical permet le transfert direct du caractère du geste vers le caractère du son, mais ne permet pas de contrôler précisément la valeur d'un paramètre. Sur une guitare acoustique, par exemple, il est relativement facile de rejouer une même phrase musicale en lui donnant chaque fois un caractère différent (agressif, bondissant, sombre, brillant, etc.). Par contre, il est impossible, et ce même pour un interprète aguerris, de jouer deux notes rigoureusement identiques, mais dont l'une serait précisément quatre décibels plus forte que l'autre.

Le contrôle paramétrique, quant à lui, permet ce degré de précision, mais ne permet pas le transfert du caractère du geste. Si, par exemple, on utilise le clavier alphanumérique d'un ordinateur pour entrer la fréquence exacte d'un oscillateur, on est bien en présence d'un contrôle paramétrique. À ce niveau extrême, le geste est complètement divorcé du son.

Une fois ces deux extrêmes établis, il est relativement aisé de classer les différents contrôles dans le continuum.

Le potentiomètre d'une table de mixage offre un contrôle légèrement plus musical que le clavier d'ordinateur puisqu'un aspect du geste est relié à un aspect du son. Cependant, l'énergie du geste n'est pas transférée à l'énergie du son. Si c'était la vitesse de déplacement du potentiomètre qui affectait l'amplitude, on se rapprocherait encore du contrôle musical.

La référence en ce qui concerne le contrôle musical, c'est le corps sonore. Tout objet qu'on peut manipuler physiquement et qui produit un son acoustique offre forcément un contrôle musical. Paradoxalement, ce qui différencie les instruments acoustiques de tous les autres corps sonores, c'est qu'ils permettent le contrôle précis de cer-

tains paramètres musicaux, comme par exemple, celui de la hauteur.

Le défi des luthiers traditionnels, c'est de fabriquer des instruments acoustiques qui offrent un contrôle paramétrique, alors que la difficulté en lutherie numérique, c'est de fabriquer des instruments qui offrent un contrôle musical.

## 2.1. Les problèmes du contrôle paramétrique

Le contrôle paramétrique est très utile et souhaitable lors d'un certain type de travail en studio comme, par exemple, les étapes de mixage et de mastering qui ont bien besoin de ce genre de précision. Par contre, ce type de contrôle est mal adapté à l'interprétation ou à l'improvisation. Il comporte trois principales faiblesses.

### 2.1.1. Le côté ludique est atrophié

Comme il n'y a pas de transfert direct du caractère du geste au caractère du son, le contrôle paramétrique requiert une étape d'analyse avant l'intervention. Il est évident que jouer d'un instrument acoustique requiert aussi un travail intellectuel, mais insuffler un caractère au son ne se fait qu'en un seul geste. Inversement, insuffler un caractère en utilisant un séquenceur audio requiert la coordination et l'ajustement précis d'une multitude de courbes d'automatisation. Cette tâche laborieuse, recherchée avec soin, est longue à mettre en oeuvre et empêche donc l'approche impromptue du jeu musical traditionnel.

### 2.1.2. Obtenir des caractères musicaux est très difficile

Si nos outils n'offrent que des contrôles proches du contrôle paramétrique, transformer un caractère *agressif* en un caractère *scintillant*, par exemple, sera extrêmement difficile. Si on veut arriver à ce résultat en utilisant un séquenceur audio traditionnel, il faudra encore une fois ajuster et coordonner une multitude de courbes d'automatisation.

Malheureusement, en musique électroacoustique, le contrôle musical pur ne se trouve qu'à l'étape de la prise de son, lors de la manipulation des corps sonores. La grande majorité des outils logiciels disponibles à ce jour n'offrent que des contrôles plus proches du contrôle paramétrique.

### 2.1.3. L'interaction avec le public est minimale

Dans le cadre d'une expérience, Wanderley et collab. [15] ont demandé à des participants de regarder, d'écouter ou de regarder *et* écouter la performance d'un clarinetiste. La perception que les participants avaient de la tension musicale et des phrasés musicaux était enregistrée en temps réel. En outre, Wanderley et collab. [15] ont observé que les gestes de l'interprète ne faisaient pas que suivre les phrases musicales, ils les prolongeaient dans le silence. Les résultats de cette expérience ont montré que les participants qui voyaient l'interprète comprenaient mieux les messages musicaux qui étaient véhiculés. Les

informations sur la tension et les phrasés seraient donc communiquées autant par le geste que par le son.

À cela, me basant sur mon expérience personnelle, j'ajouterais que la seule présence d'un interprète ne suffit pas. Quand le contrôle qu'exerce le performeur est trop loin du contrôle musical, ses gestes sont divorcés du son et ne contribuent pas à communiquer le message. Lors d'une performance de *live coding*<sup>1</sup>, par exemple, les gestes et les expressions faciales du codeur sont complètement détachés du son produit.

## 3. L'ÉPONGE : VERS UN CONTRÔLE MUSICAL DE LA MUSIQUE ÉLECTROACOUSTIQUE



Figure 1. L'éponge.

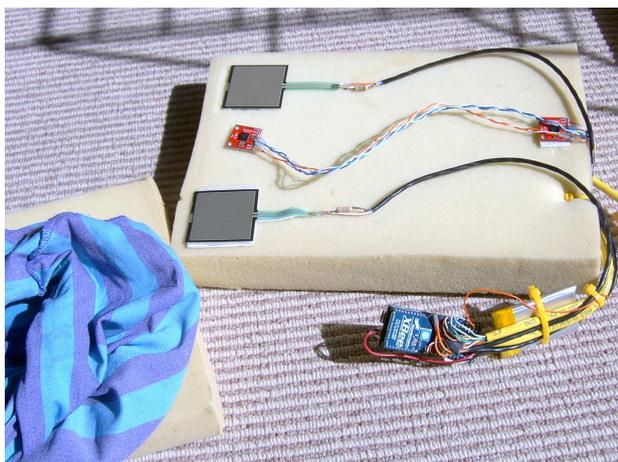
L'éponge est une interface musicale que j'ai développée dans le but de permettre l'interprétation devant public de pièces électroacoustiques. Elle ressemble en tout point à un coussin rayé (voir la figure 1), sauf que des capteurs installés à l'intérieur détectent les chocs qu'elle subit, son inclinaison, ainsi que ses déformations (compression, torsion, pliage). Afin d'exploiter le potentiel qu'offre le médium électroacoustique, j'ai développé une interface dont la forme et l'aspect ne rappellent aucun instrument acoustique. Une interface calquée sur un instrument acoustique existant aurait permis d'exploiter l'expertise d'un instrumentiste aguerri [2]. Par contre elle aurait aussi conditionné le travail de création en imposant son mode de jeu, ce qui aurait considérablement biaisé les choix esthétiques.

En résumé, l'éponge a été conçue pour permettre d'interpréter la musique électroacoustique en permettant un contrôle musical.

### 3.1. Construction et capteurs

L'éponge comporte 11 capteurs analogiques : sept interrupteurs momentanés, deux capteurs de pression (FSR)

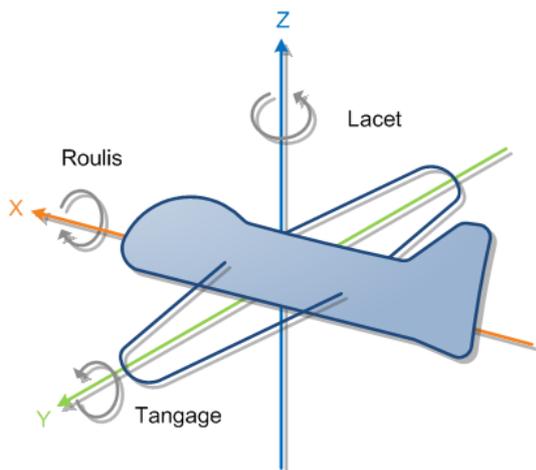
1. Le *live coding* est une pratique qui consiste à utiliser un langage de programmation dédié à la musique pour performer devant un public. Traditionnellement, le code du performeur est projeté sur grand écran pour que le public puisse voir et apprécier son travail.



**Figure 2.** L'éponge nue. On peut voir les deux capteurs de pression (carrés gris), les accéléromètres (composantes rouges), l'interface Arduino Fio et le module XBee. Les sept boutons ne sont pas visibles sur cette photo ; ils sont sous l'éponge, du côté droit.

et deux accéléromètres 3D. Comme les deux accéléromètres sont sensibles aux trois dimensions, l'éponge fournit un total de 15 signaux.

Les capteurs de pression permettent de détecter la pression appliquée sur l'éponge (les carrés gris visibles sur la figure 2). Les accéléromètres (les composantes rouges visibles sur la figure 2) captent une multitude de déformations et mouvements. Ils servent d'abord à détecter l'inclinaison globale de l'éponge sur les trois axes, soit le roulis, le tangage et le lacet (illustrés par la figure 3).<sup>2</sup>



**Figure 3.** Les notions de roulis, tangage et lacet sont empruntées au domaine de l'aéronautique.

Ensuite, en calculant les différences d'inclinaison entre les deux accéléromètres, il est possible de déduire la tor-

2. Les accéléromètres ne peuvent fournir que deux de ces rotations au même moment. Par exemple, lorsque l'éponge est tenue horizontalement, il est impossible de connaître le lacet. Il faudrait ajouter une boussole à l'éponge pour rendre cette donnée disponible.

sion et le pliage. Les accéléromètres sont aussi utilisés pour détecter les chocs et les vibrations.

Les six axes des accéléromètres ainsi que les deux capteurs de pression sont connectés aux huit entrées analogiques d'une interface *Arduino Fio*. Les sept boutons sont connectés à sept entrées numériques de la même interface.

Chaque signal analogique est quantifié à 10 bits approximativement 300 fois par secondes et est acheminé vers l'ordinateur via une interface *XBee* (sans fil). Si les pilotes adéquats sont installés, l'éponge apparaît au système d'exploitation comme un périphérique sériel standard (`/dev/ttyUSB0`, par exemple), lui permettant de fonctionner sur les trois systèmes d'exploitation principaux, soit Linux, Mac OSX et Windows.

### 3.2. Le mapping

#### 3.2.1. One-to-one ou many-to-many ?

Le *mapping* consiste à établir une correspondance entre les paramètres du geste (les signaux issus des capteurs) et les paramètres des algorithmes de traitement ou de génération du son. C'est donc une étape incontournable de tout travail avec une interface musicale. Lors de la conception d'un *mapping*, on peut choisir l'une de ces quatre approches [5] :

**one-to-one** Un signal de contrôle n'affecte qu'un seul paramètre.

**one-to-many** Un signal de contrôle affecte plusieurs paramètres.

**many-to-one** Plusieurs signaux de contrôle n'affectent qu'un seul paramètre.

**many-to-many** Une combinaison des options précédentes : plusieurs signaux de contrôle affectent une multitude de paramètres.

Les *mappings* implémentés dans les outils électroacoustiques traditionnels entrent habituellement dans la première catégorie. Sur une table de mixage, par exemple, le potentiomètre principal ne fait varier qu'un seul paramètre : l'amplitude.

Un bon exemple de *mapping one-to-many* est le potentiomètre d'interpolation de préréglage qu'on trouve dans les logiciels de la série GRM Tools (figure 4). En actionnant ce seul potentiomètre, tous les autres paramètres du traitement sont affectés.

Toutes les parties logicielles de l'éponge sont des outils qui servent de près ou de loin à concevoir des *mappings* qui contribuent à établir un contrôle musical.

### 3.3. Logiciel

La partie logicielle de l'éponge consiste en une bibliothèque implémentée dans l'environnement SuperCollider. Elle est composée d'une dizaine de classes qui implémentent ces trois fonctionnalités : un étage de réception des signaux, un étage d'extraction de traits caractéristiques et un interpolateur de préréglages. Elle est sans



**Figure 4.** Une capture d'écran du logiciel *Doppler* de la collection *GRM Tools*. Le long potentiomètre horizontal situé au bas de la fenêtre permet d'interpoler entre les préréglages numérotés de 1 à 8.

cesse en évolution. La version actuelle peut être téléchargée sur *github* en suivant ce lien :

<http://github.com/marierm/mmExtensions>.

### 3.3.1. Réception des signaux

La réception des signaux est implémentée dans la classe *Sponge* et permet à *SuperCollider* d'ouvrir le port sériel approprié et de rendre les données brutes des capteurs disponibles aux autres étages de *mapping*.

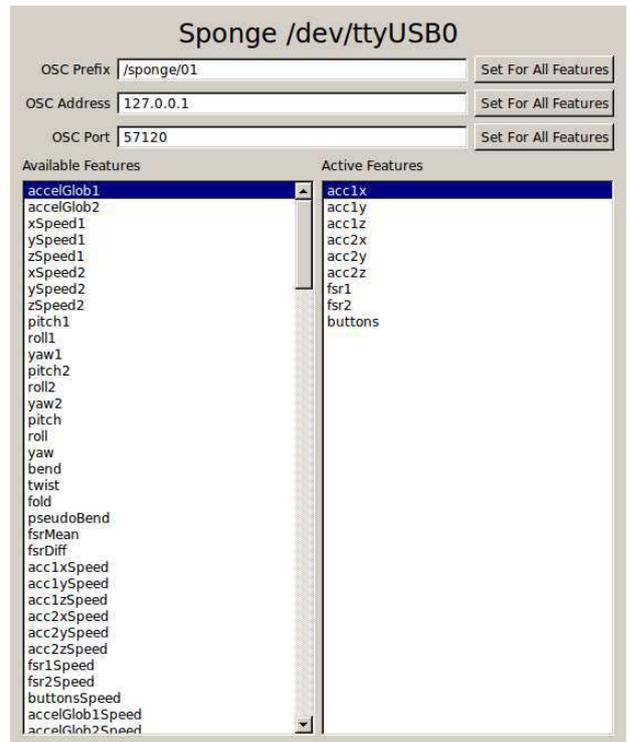
### 3.3.2. Extraction de traits caractéristiques

L'expression *extraction de traits caractéristiques* est une traduction de l'anglais *feature extraction*. Un trait caractéristique est un signal qui peut être déduit ou calculé à partir des signaux bruts des capteurs. Par exemple, il est possible d'extraire de l'éponge le trait caractéristique  $pitch_1$  (l'inclinaison de l'accéléromètre numéro un) qui est calculé à partir des axes  $x$  et  $z$  de l'accéléromètre 1 :

$$pitch_1 = \arctan\left(\frac{acc_1x}{acc_1z}\right).$$

Il est aussi possible d'extraire d'autres traits caractéristiques à partir de traits caractéristiques existants et ainsi de suite. Par exemple, les traits caractéristiques  $roll_1$  et  $roll_2$  sont obtenus respectivement avec les formules  $roll_1 = \arctan\left(\frac{acc_1y}{acc_1z}\right)$  et  $roll_2 = \arctan\left(\frac{acc_2y}{acc_2z}\right)$ . Par la suite, on peut obtenir le trait caractéristique *twist* en soustrayant  $roll_2$  de  $roll_1$ . Ces interdépendances sont gérées automatiquement par la classe *Feature* : si un utilisateur active le trait caractéristique *twist*, les traits caractéristiques  $roll_1$  et  $roll_2$  seront automatiquement activés.

L'extraction de traits caractéristiques permet de séparer le *mapping* en plusieurs étages, ce qui facilite la conception de *mappings* complexes [16]. De plus, cette stratégie rend disponibles des signaux qui sont plus représentatifs du geste que les signaux bruts des capteurs, ce qui aide à obtenir un contrôle musical.



**Figure 5.** L'interface graphique qui permet d'activer et de désactiver les nombreux traits caractéristiques préprogrammés.

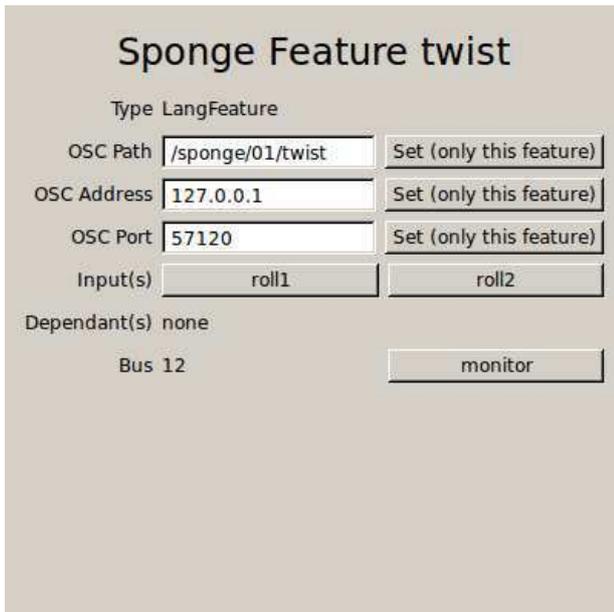
Il existe une multitude de traits caractéristiques (environ 200) préprogrammés pour l'éponge qui peuvent être activés ou désactivés très rapidement selon les besoins de l'utilisateur. La figure 5 est une capture d'écran de l'interface graphique qui permet de gérer l'activation des différents traits caractéristiques. Les signaux correspondants aux traits caractéristiques deviennent alors disponibles à l'utilisateur. Il est aussi possible de les visionner (figure 7) et de les acheminer vers d'autres applications ou d'autres appareils sous forme de messages OSC (figure 6 et 5).

### 3.3.3. Interpolation de préréglages

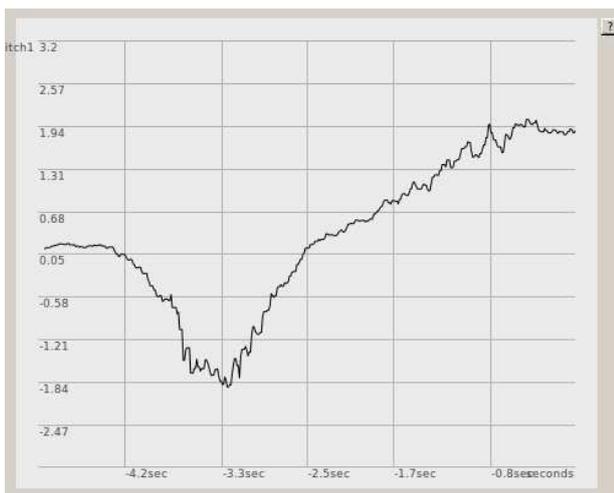
Un système d'interpolation de préréglages permet à l'utilisateur de faire varier une multitude de paramètres en n'agissant que sur un nombre limité de contrôles [3, 1, 14, 4, 10]. De par sa nature, un tel système facilite grandement la conception de *mappings* complexes de type *many-to-many*, ce qui permet aussi de se rapprocher d'un contrôle musical.

Le système d'interpolation de préréglages que j'ai développé prend la forme d'un espace virtuel (à une ou plusieurs dimensions) dans lequel il est possible de disposer une multitude de points représentant chacun un préréglage. Par la suite, l'utilisateur peut déplacer un curseur dans cet espace. La valeur des multiples paramètres de chacun des préréglages varie avec la position du curseur.

Le tableau 1 illustre le fonctionnement d'un interpolateur de préréglages à une dimension qui comporte simplement deux points. Dans cet exemple, deux paramètres



**Figure 6.** L'interface graphique du trait caractéristique *twist*. L'utilisateur peut choisir le format du message OSC qui sera envoyé.



**Figure 7.** Il est possible de visualiser les signaux des capteurs et des traits caractéristiques.

d'un filtre en cloche variant en fonction de la position du contrôle d'interpolation.

Position du contrôle d'interpolation	Poids des pré-réglages	Gain	Fréquence centrale
A	A	+12 dB	400 Hz
$\frac{1}{4}$ de course	$\frac{3}{4}A + \frac{1}{4}B$	+10.5 dB	800 Hz
$\frac{1}{2}$ de course	$\frac{1}{2}A + \frac{1}{2}B$	+9 dB	1200 Hz
$\frac{3}{4}$ de course	$\frac{1}{4}A + \frac{3}{4}B$	+7.5 dB	1600 Hz
B	B	+6 dB	2000 Hz

**Table 1.** Les valeurs que prennent les paramètres d'un filtre cloche en fonction de la position du contrôle d'interpolation.

Les classes *Interpolator* et *PresetInterpolator* implémentent un système d'interpolation de pré-réglage beaucoup plus complexe et beaucoup plus versatile. Il est possible d'y créer des espaces à un nombre arbitraire de dimensions et d'y placer un nombre infini de pré-réglages (l'utilisateur n'est limité que par les capacités de son ordinateur). Le nombre de dimensions de l'espace correspond au nombre de signaux de contrôles utilisés (le nombre de capteurs). Par exemple, on peut choisir de connecter un accéléromètre 3D à un interpolateur de pré-réglages à trois dimensions (une dimension pour chacun des axes de l'accéléromètre). Autrement dit, on navigue dans l'espace 3D à l'aide de l'accéléromètre. Les espaces à plus de trois dimensions sont plus difficile à imaginer, mais il s'agit d'avoir autant de dimensions que de contrôleurs continus.

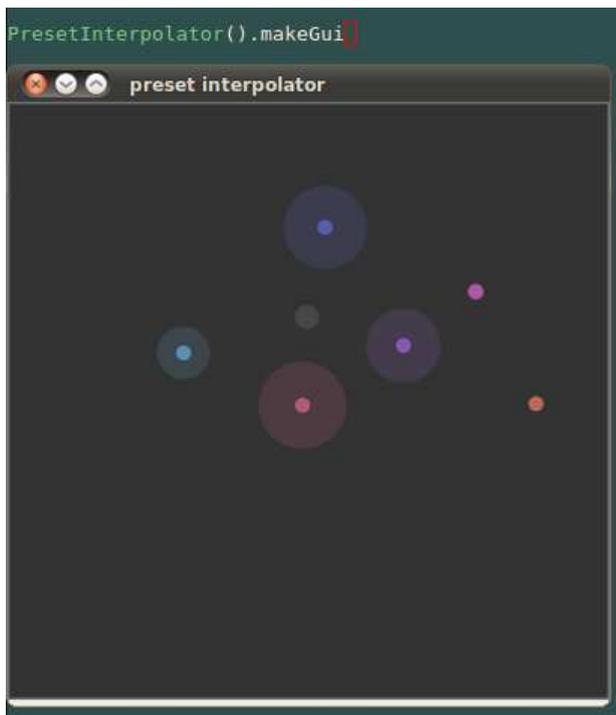
Pour utiliser l'interpolateur de pré-réglage avec l'éponge, on peut se créer un espace d'interpolation à huit dimensions auxquelles on connecte les huit signaux continus de l'éponge. Le nombre de paramètres que l'on peut contrôler est complètement indépendant du nombre de dimension.

## 4. ENJEUX ET DÉFIS

### 4.1. L'éponge : interface ou instrument ?

Les acteurs importants du domaine des nouvelles interfaces (ou nouveaux instruments) ne s'entendent toujours pas sur la définition exacte de ces deux termes. Pour les besoins de la discussion, j'adopterai la définition suggérée par [9] et illustrée par la figure 9. Un instrument complet est constitué de quatre étages successif : l'interface, la *mapping*, la génération du son et la diffusion. Comme on peut le voir, l'interface n'est en fait que le premier étage d'un instrument, et c'est exactement ce que l'éponge est.

Fait important à noter, sur un instrument de musique numérique, les différents étages sont discrets, alors que sur la majorité des instruments acoustiques, les éléments sont indissociables les uns des autres.



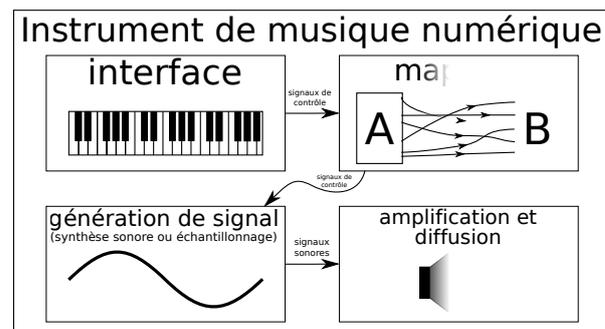
**Figure 8.** La fenêtre *preset interpolator* permet à l'utilisateur de visualiser et de modifier l'emplacement des pré-réglages dans l'espace. Si l'espace comporte plus de deux dimensions, il est évidemment impossible de toutes les visualiser. Cependant, il est possible d'ouvrir plusieurs fenêtres *preset interpolator* et de choisir quelles dimensions seront visualisées dans chacune d'elles. Chacun des points colorés correspond à un pré-réglage. Le point gris légèrement plus gros que les autres est le curseur. Le poids des points est représenté par des cercles transparents : plus le cercle autour d'un point est grand, plus le point a du poids.

#### 4.2. Établir un lien d'énergie

Plusieurs chercheurs sur les interfaces musicales [12, 7, 11] ont déjà tiré la conclusion que, si l'on désire un rapport geste-son clair, il doit y avoir un transfert de l'énergie du geste vers l'énergie du son. Pour établir ce transfert d'énergie, il est suggéré de faire correspondre la vitesse du geste à l'amplitude du signal [6]. Avec l'éponge et son système d'extraction de traits caractéristiques, il est possible d'avoir rapidement accès aux différentes *vitesse*s qui sont en jeu :

- la vitesse de pression (dérivée du signal des capteurs de pressions) ;
- la vitesse de rotation globale de l'éponge sur chacun des axes ;
- la vitesse de pliage et de torsion.

De plus, comme l'accélération est elle-même directement proportionnelle à l'énergie, il est tout à fait cohérent d'affecter l'amplitude d'un son avec les signaux qui proviennent directement des accéléromètres. D'ailleurs, dans la pièce *Clarinette* (cette pièce est discutée à la section 5), les signaux des accéléromètres (filtrés par un passe haut) sont utilisés comme excitateurs pour un synthétiseur



**Figure 9.** L'instrument de musique est un système complet alors que l'interface ne représente que son premier étage. Cette figure est une adaptation de celle qu'on retrouve dans le livre de [9, p. 3]

à guide d'ondes. Ce *mapping* a beaucoup de succès auprès du public : les gens comprennent intuitivement le lien qui existe entre le geste et le son.

#### 4.3. Gestion de la polyphonie

L'éponge a été conçue comme une interface monophonique mais, rapidement, le besoin de faire de la polyphonie s'est fait sentir. Mis à part la solution évidente qui consiste à utiliser plusieurs éponges, il existe quelques façons d'arriver à jouer plusieurs voix.

La première consiste à utiliser les boutons de l'éponge pour démarrer et arrêter différentes voix. La stratégie utilisée jusqu'à maintenant consiste à attribuer trois états différents à chaque bouton. Voici à quels comportements sont associés chacun des états.

1. Le processus est déclenché : une voix joue et on est en contrôle de ses différents paramètres avec l'éponge.
2. Le processus continue, mais l'éponge ne contrôle plus les paramètres de la voix. La voix est gelée.
3. La voix s'éteint doucement (en fondu).

De cette façon, il est possible d'avoir une voix de polyphonie par bouton, c'est-à-dire sept. Cela fonctionne bien, mais il est très difficile pour l'interprète de se souvenir de l'état de chacune des voix. En réalité, il est impossible de s'en sortir lorsque plus de trois voix dont le timbre est semblable sont utilisées en même temps.

La deuxième façon de faire est très semblable ; il n'y a qu'une exception : au lieu d'utiliser le deuxième état pour geler la voix, on l'utilise pour boucler la voix. Aussitôt que le boucler de geste décrit à la section 6.1 sera fonctionnel, cette stratégie remplacera la première.

#### 4.4. Composer avec la multidisciplinarité

Être à la fois compositeur, interprète et concepteur d'instrument pose certains problèmes qui peuvent être illustrés par l'exemple suivant.

Imaginons un compositeur-interprète-concepteur qui se retrouve face à un problème musical typique : il n'est pas satisfait du résultat sonore qu'il entend et veut le changer. Trois possibilités s'offrent alors à lui.

D'abord, il peut opter pour l'approche de l'interprète. Dans ce cas, il pratiquera jusqu'à ce qu'il ait développé les habiletés qui lui permettront de jouer la phrase musicale problématique de façon satisfaisante. Il pourrait aussi opter pour l'approche du compositeur, auquel cas il choisira de remettre en question la musique elle-même et de recomposer des phrasés mieux adaptés à l'instrument. Sa dernière option serait l'approche du concepteur. Dans ce dernier cas, il choisira de modifier l'instrument de façon à ce qu'il devienne possible de produire un résultat plus satisfaisant. Cette solution est la plus laborieuse puisqu'elle impose à l'interprète de réapprendre à jouer de l'instrument modifié.

Un instrument performant et expressif doit se plier aux besoins de l'interprète et du compositeur, mais le développement continu de l'instrument empêche de fixer les oeuvres et, pire encore, de rejouer des oeuvres composées pour une version antérieure de l'instrument.

Découvrir et développer des stratégies qui permettront de conjuguer développement continu et pérennité des oeuvres est une partie importante de cette recherche. Pour l'instant, des systèmes de gestion de versions comme Subversion ou Git (très utilisés par les programmeurs) sont utilisés pour pouvoir facilement jongler avec les multiples versions d'une pièce. Cette solution est très pratique pour la gestion de la partie logicielle, mais n'est d'aucune utilité pour gérer différentes versions d'une interface.

#### 4.5. La question de la pérennité

La plupart du temps, en musique électroacoustique, la partition ne sert que de support pour l'analyse ou d'aide à la diffusion. Comme il s'agit traditionnellement d'un art des sons fixés, cela ne pose pas de problème : la pérennité de l'oeuvre est assurée par son support. Mais lorsqu'on tombe dans le domaine des arts d'interprétation, une oeuvre n'est considérée pérenne qui si elle peut être réinterprétée. Donc, pour assurer une longue vie à une oeuvre pour éponge, on doit d'abord s'assurer que les appareils et logiciels utilisés pourront être remplacés au fur et à mesure que la technologie évolue. On doit aussi veiller à ce qu'un interprète puisse apprendre et rejouer l'oeuvre.

Il est possible de remplir la première condition en documentant clairement les processus utilisés de façon à ce qu'il soit possible de réimplémenter les mêmes algorithmes en utilisant la technologie du futur. Il est aussi possible d'utiliser des technologies que l'on croit plus durables que d'autres comme par exemples des logiciels dont le code source est ouvert.

Remplir la deuxième condition est délicat. On pourrait croire que la partition est la solution, mais ce n'est pas si simple. Un des objectifs de cette recherche est de permettre l'interprétation en exploitant le médium électroacoustique. Hors, il n'existe pas de système de notation standard pour ce genre de musique. Il est certes possible

d'en concevoir un, mais basé sur quoi ? Sur la notation traditionnelle ? Sur une notation du geste comme celle de Laban pour la danse ?

Thierry De Mey, pour sa pièce *Light Music*<sup>3</sup>, a opté pour une partition hybride. Certaines sections utilisent la notation musicale traditionnelle, alors que d'autres emploient des dessins qui évoquent les gestes que l'interprète doit faire. Cette seule partition ne suffirait pas à assurer la pérennité de cette oeuvre. D'ailleurs, le compositeur et son équipe ont dernièrement organisé un stage de formation dont le but était de former des interprètes pour cette oeuvre.

Pour l'instant, développer un système de notation rigoureux et systématique pour les oeuvres pour éponge n'est pas une priorité. Pour assister le travail sur la structure des pièces, un simple séquenceur est utilisé. Après avoir été enregistrées, des performances sont retravaillées dans le logiciel. Par la suite, j'essaie de rejouer à l'éponge la version retravaillée. Si la version est injouable (ce qui est habituellement le cas) je recommence.

Dans ce procédé, le logiciel constitue un substitut à la partition, mais uniquement pour le travail de composition ; il ne permet aucunement la transmission des oeuvres. Pour permettre une éventuelle transmission d'une oeuvre, une stratégie impliquant des captations vidéo de bonne qualité et de différents points de vue serait choisie plutôt qu'un système de notation.

## 5. LA COMPOSITION

### 5.1. *Cymbale*, la pièce morte

*Cymbale* avait été composée pour la toute première version de l'éponge, celle qui était construite à partir d'une interface Infusion Systems BlueTooth [8]. Cette technologie ne s'est pas avérée adéquate pour ce type d'interface et, peu de temps après la création de la pièce, l'éponge 2.0 a vu le jour. Adapter *Cymbale* à la nouvelle version de l'éponge aurait représenté un travail considérable et, comme elle n'était pas très satisfaisante esthétiquement, la pièce fut abandonnée.

La décision a été consciente, mais la mort de *Cymbale* n'était pas voulue. Cet événement m'a appris à quel point les pièces écrites pour nouvelles interfaces sont fragiles.

### 5.2. *Clarinette* et ses descendants

*Clarinette* a un destin différent : elle est toujours jouée et elle évolue sans cesse. En fait, elle évolue à un point tel qu'il est difficile de considérer qu'il s'agit encore de la même pièce !

*Clarinette* est donc une pièce laboratoire. À chaque nouvelle mouture, au fur et à mesure que l'éponge offre de nouvelles possibilités, de nouveaux éléments se greffent. Au départ, il n'y avait que des enregistrements de clarinette traités par granulation. Ensuite, des sons de synthèse

3. *Light Music* est une pièce pour « chef d'orchestre » et dispositif électronique. Les gestes de l'interprète sont suivis par des caméras vidéos et affectent le son

par guide d'onde se sont ajoutés. Parallèlement à cela, les stratégies de *mapping* se sont raffinées. L'interpolateur de pré-réglage s'est perfectionné et est devenu multidimensionnel, ce qui permet d'exploiter une plus grande variété de gestes et d'avoir plus de précision. Des boutons ont été ajoutés, ce qui permet d'avoir un contrôle sur la polyphonie, de déclencher des événements et de jouer avec des hauteurs discrètes. Enfin, des filtres formantiques et de la distorsion ont été intégrés.

Si on ajoute à cela le fait que la structure n'est toujours pas fixée, il n'est pas très étonnant que les différentes versions de *Clarinette* ne se ressemblent pas beaucoup. Pour cette raison, elle porte toujours un sous-titre. Il existe trois versions importantes de la pièce :

**Clarinette (2009)** La première version jouée sur l'éponge originale [8]. Les matériaux sonores étaient constitués uniquement de sons de clarinette granulés.

**Struggling (2010)** Cette version était jouée sur l'éponge 2.0 (identique à la version actuelle, mais sans boutons) et incorporait de la synthèse par guide d'onde. Pour la première fois, il était possible de jouer des percussions sur l'éponge sans problèmes de latence.

**Albino Butterfly (2011)** C'est la première version à exploiter les boutons. Ils permettent de gérer jusqu'à quatre voix de polyphonie. La distorsion et les filtres formantiques ont été intégrés.

*Clarinette* devrait se fixer avant la fin de 2012, mais pour l'instant, elle reste une pièce laboratoire grâce à laquelle il est possible d'expérimenter avec l'éponge devant public.

## 6. DÉVELOPPEMENTS FUTURS

### 6.1. Boucleur de geste

Depuis les expérimentations de [13] sur le sillon fermé, le concept de la boucle audio est bien connu : l'idée est simplement de répéter cycliquement un signal audio.

Le boucleur de geste est un système pratiquement identique sauf qu'il est conçu pour boucler des signaux de contrôle plutôt que des signaux audio.

Cette nuance peut sembler anodine, mais si on y regarde de plus près, on constatera que choisir de boucler le geste plutôt que le son a un impact important.

#### 6.1.1. Concept

Pour boucler un geste, il faut boucler un ou plusieurs signaux de contrôle. Il peut s'agir de signaux obtenus directement des capteurs ou encore de traits caractéristiques qui ont été préalablement extraits. Les signaux ainsi bouclés peuvent alors affecter les paramètres qui étaient contrôlés avec l'éponge. L'interprète, qui n'a plus à contrôler ces paramètres, peut alors décider de créer une nouvelle voix de polyphonie en se faisant accompagner par la boucle.

Un tel système n'offre pas d'avantages majeurs par rapport au boucleur de son traditionnel. Ce qui est différent,

c'est qu'il devient aussi possible de boucler un nombre limité de paramètres et de garder le contrôle manuel d'autres paramètres. Par exemple, si on enregistre une boucle dans laquelle les paramètres *fréquence*, *amplitude* et *amplitude du vibrato* varient, on peut boucler les deux premiers paramètres et garder le contrôle manuel l'*amplitude du vibrato*.

Il devient aussi possible d'appliquer le concept du mode *trim*<sup>4</sup> qu'on retrouve dans les séquenceurs audio numériques comme ProTools ou Digital Performer. En effet, plutôt que de prendre le contrôle d'un paramètre, on peut le boucler et venir l'altérer en temps réel. De cette façon, on répète le geste musical enregistré, mais on fait finement varier la boucle.

#### 6.1.2. Intégration avec l'interpolateur de pré-réglages

Le boucleur de geste en lui-même n'aide aucunement à établir un contrôle musical. Son utilité est à un autre niveau : il permet à un interprète de gérer plusieurs voix de polyphonie avec une interface qui, à prime abord, semble suggérer la monophonie.

Mais si on veut se rapprocher du contrôle musical, on peut l'utiliser de pair avec l'interpolateur de pré-réglage : plutôt que de boucler directement les paramètres, on peut boucler des trajectoires dans l'espace multidimensionnel. De cette façon, une seule boucle affecte une multitude de paramètres. On peut alors utiliser le mode *trim* pour venir modifier en temps réel la trajectoire bouclée.

#### 6.1.3. Implémentation

Au moment de l'écriture de ce document, le boucleur de geste n'est pas encore utilisable. Les fonctionnalités de bases sont implémentées dans les classes *Looper* et *FeatureLooper*, mais l'intégration avec l'interpolateur de pré-réglages et l'éponge elle-même n'est pas encore codée.

### 6.2. Projet de pièces finales

L'objectif est d'avoir quatre pièces pour éponge. Le but est d'écrire des œuvres dont la structure est claire et perceptible dès une première écoute. Le principe de tension-détente et la recherche d'un équilibre entre redondance et originalité guideront l'écriture.

Les premières pièces pour éponge (*Cymbale* et *Clarinette*) étaient plutôt minimalistes et *ambient*. Cela vient bien entendu d'un intérêt pour ce genre, mais les contraintes techniques liées à l'utilisation d'une nouvelle interface y étaient aussi pour beaucoup. Maintenant que l'éponge a atteint une certaine maturité et que les temps de latence ont été grandement réduits, il est désormais plus facile de jouer avec des matériaux plus articulés et plus dynamiques. Les nouvelles pièces comporteront des sections rythmiques dans lesquelles la pulsation sera l'ancrage du discours musical.

<sup>4</sup> . Le mode *trim* est un mode d'enregistrement d'automatisations qui additionne le geste enregistré à la courbe qui est déjà là.

### 6.3. Interface, mappings et pièces : des modules assemblables

À écrire de la musique pour une nouvelle interface, on est forcé de se demander si le *mapping* utilisé est associé à l'interface ou à la pièce. Lorsqu'un *mapping* est associé à l'interface, on se retrouve avec un instrument de musique numérique pour lequel on peut écrire plusieurs pièces. À l'inverse, si le *mapping* est associé à une pièce, on obtient une entité *pièce-mapping-interface* monolithique.

Dans le travail à venir, ces deux avenues seront explorées : il y aura deux pièces pour un même instrument et une autre pièce composée pour un *mapping* complètement différent.

De plus, une pièce pour deux éponges sera écrite. En plus de permettre une polyphonie plus évoluée, cette avenue permettra d'explorer les possibilités d'interaction avec d'autres musiciens.

## 7. CONCLUSION

Le travail fait au cours des dernières années a permis d'amener l'éponge à un certain niveau de maturité. L'interface est devenue à la fois plus robuste et plus efficace, et la couche logicielle rend maintenant la conception de *mappings* complexes beaucoup facile et rapide. Parallèlement à l'évolution de l'éponge, il y a eu mon évolution personnelle : j'ai énormément progressé comme interprète... comme épongist.

Il y aura toujours quelque chose à améliorer sur l'éponge, mais il semble que, pour la première fois depuis quatre ans, mon outil est enfin prêt et que je peux maintenant composer et jouer de la musique.

## 8. REMERCIEMENTS

Merci à Prof. Jean Piché, Annie Lalancette, Dr. Garth Paine et à Georges Forget. Nos échanges me nourrissent énormément. Sans votre support, l'éponge laverait encore de la vaisselle.

## 9. REFERENCES

- [1] Ross Bencina. The metasurface : applying natural neighbour interpolation to two-to-many mapping. In *Proceedings of the 2005 conference on New interfaces for musical expression*, pages 101–104. National University of Singapore, 2005.
- [2] Christopher Dobrian and Daniel Koppelman. The 'E' in NIME : musical expression with new computer interfaces. In *Proceedings of the 2006 conference on New interfaces for musical expression*, pages 277–282. IRCAM—Centre Pompidou, 2006.
- [3] Adrian Freed, John MacCallum, Andrew Schmeder, and David Wessel. Visualizations and Interaction Strategies for Hybridization Interfaces. In *NIME '10 : Proceedings of the 2010 conference on New interfaces for musical expression*, number Nime, pages 343–347, Sydney, 2010.
- [4] Camille Goudeseune. Interpolated mappings for musical instruments. *Organised Sound*, 7(02) :85–96, 2003.
- [5] Andy Hunt and Marcelo M. Wanderley. Mapping performer parameters to synthesis engines. *Organised Sound*, 7(02) :97–108, 2002.
- [6] Andy Hunt, Marcelo M. Wanderley, and Ross Kirk. Towards a Model for Instrumental Mapping in Expert Musical Interaction. In *Proceedings of the 2000 International Computer Music Conference*, pages 209–212, 2000.
- [7] Andy Hunt, Marcelo M Wanderley, and Matthew Paradis. The importance of parameter mapping in electronic instrument design. In *NIME '02 : Proceedings of the 2002 conference on New interfaces for musical expression*, pages 1–6, Singapore, Singapore, 2002. National University of Singapore.
- [8] Martin Marier. The Sponge : A Flexible Interface. In *NIME '10 : Proceedings of the 2010 conference on New interfaces for musical expression*, pages 356–359, Sydney, 2010.
- [9] Eduardo R. Miranda, Marcelo M. Wanderley, and Ross Kirk. *New digital musical instruments : control and interaction beyond the keyboard*. AR Editions, Inc., Middleton, 2006.
- [10] Ali Momeni and David Wessel. Characterizing and controlling musical material intuitively with geometric models. In *Proceedings of the 2003 conference on New interfaces for musical expression*, pages 54–62. National University of Singapore, 2003.
- [11] Garth Paine. Towards Unified Design Guidelines for New Interfaces for Musical Expression. *Organised Sound*, 14(02) :142–155, 2009.
- [12] Joel Ryan. Some remarks on musical instrument design at STEIM. *Contemporary Music Review*, 6(1) :3–17, 1991.
- [13] Pierre Schaeffer. *Traité des objets musicaux*. Éditions du Seuil, Paris, France, 1ère édition, 1966.
- [14] Martin Spain and Richard Polfreman. Interpolator : a two-dimensional graphical interpolation system for the simultaneous control of digital signal processing parameters. *Organised Sound*, 6(02) :147–151, February 2002.
- [15] M M Wanderley, B W Vines, N Middleton, C McKay, and W Hatch. The musical significance of clarinetists' ancillary gestures : An exploration of the field. *Journal of New Music Research*, 34(1) :97–113, 2005.
- [16] M.M. Wanderley, Norbert Schnell, and Joseph Rowan. Escher-modeling and performing composed instruments in real-time, 1998.