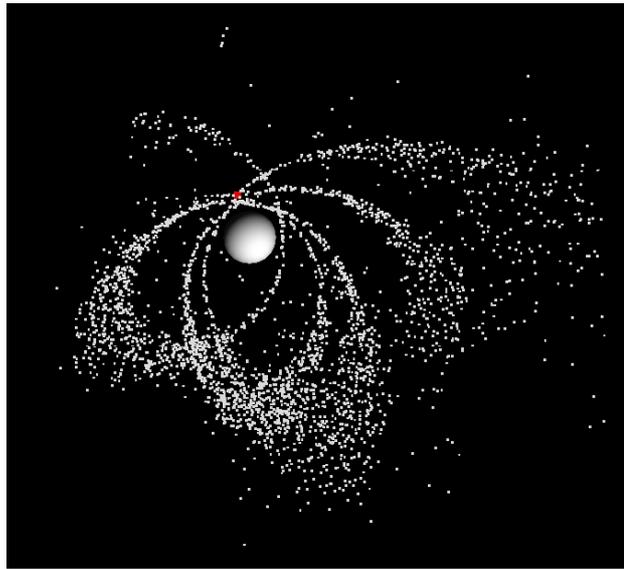




ircam
Centre
Pompidou

MODÈLES PHYSIQUES PARTICULAIRES EN ENVIRONNEMENT
TEMPS-RÉEL : APPLICATION AU CONTRÔLE DES
PARAMÈTRES DE SYNTHÈSE.



NICOLAS MONTGERMONT 7 mars 2005 - 7 juillet 2005

Université Pierre et Marie Curie : MASTER Science et Technologie, mention Mécanique et Ingénierie des Systèmes, parcours Acoustique, Traitement du Signal et Informatique Appliqués à la Musique.

Responsable : Cyrille Henry

Mots-clés : modèle physique, oscillateur mécanique, système masse-ressort-amortisseur, mapping, paramètres de synthèse, Pure Data, Max/MSP

*”La musique électronique nous
contraint à assembler chaque
note selon notre besoin”.*
- Pierre Boulez, 1955

Table des matières

1	Remerciements	6
2	Introduction	7
2.1	Objectif du stage	7
2.2	La kitchen	8
2.2.1	Lieu de création et de développement artistique	8
2.2.2	La recherche dans les domaines de l'innovation et des hautes technologies	9
3	Contexte scientifique	10
3.1	Modèle masse ressort amortisseur	10
3.1.1	Équation de la masse	11
3.1.2	Equation de la liaison	11
3.1.3	Oscillation d'un système masse-ressort-amortisseur	12
3.2	Modèles physiques particuliers	14
3.3	Mapping entre les paramètres de contrôle et les paramètres de synthèse	16
3.3.1	Définition et importance du mapping	16
3.3.2	État de l'art	18
3.3.3	Extension au mapping de la synthèse vidéo	23
3.4	Environnements temps-réel pour le traitement du signal : Pure Data et Max/MSP	23
4	Implémentation numérique : la collection d'objets externes MSD	26
4.1	Présentation du modèle	26
4.2	Equation de l'élément masse	27
4.2.1	Système à 1 degré de liberté	27
4.2.2	Extension du modèle à 2 et 3 dimensions	28
4.3	Equation de l'élément liaison	28
4.3.1	Système à 1 degré de liberté	28
4.3.2	Extension du modèle à 2 et 3 dimensions	29
4.4	Introduction de non linéarités	29
4.4.1	Limites du travail de la force de rappel	29
4.4.2	Puissance du travail de la force de rappel	30

4.4.3	Liaisons orientées	30
4.5	Utilisation de la librairie MSD	32
4.5.1	Messages de création	34
4.5.2	Message de calcul	34
4.5.3	Messages attributs	34
4.5.4	Messages de paramétrage	35
4.6	Résultats	35
4.6.1	Oscillation du système masse - ressort - amortisseur . .	35
4.6.2	Performances	36
5	Les modèles physiques pour le contrôle des paramètres de synthèse	37
5.1	Utilisation des modèles physiques pour le mapping	37
5.1.1	Métaphores du réel	37
5.1.2	Exemple de la synthèse additive	39
5.1.3	Exemple de la synthèse granulaire	40
5.2	Descripteurs de haut niveau	42
5.3	Modèles physiques à N dimensions	43
5.3.1	Proposition : Utilisation des modèles physiques à N dimensions	43
5.3.2	Exemple de gamme évolutive	44
6	Conclusion	46
6.1	Bilan du travail réalisé	46
6.2	Perspectives	47
7	Annexes	48
.1	Fichiers d'aide de l'objet msd2D pour Pure Data	48
.2	Téléchargement de la librairie MSD	51

Table des figures

2.1	<i>Intime</i> , pièce interactive réalisée par La kitchen.	8
3.1	Système simple représentant deux masses liées par une liaison ressort-amortisseur	10
3.2	Système masse-ressort-amortisseur idéal	13
3.3	Exponentielle amortie pour 3 valeurs de la constante d'amor- tissement.	13
3.4	Réponse en fréquence théorique du système masse ressort amor- tisseur pour 3 valeurs de la constante d'amortissement.	14
3.5	GENESIS en mode lutherie [CC02].	15
3.6	CYMATIC [HR03].	16
3.7	Schéma classique de dispositif de synthèse sonore et visuelle. . .	18
3.8	Forme virtuelle utilisée pour le mapping [MFM97].	20
3.9	ESCHER [WSR98]	22
3.10	Capture d'écran d'un patch Max/MSP de synthèse FM. Les lignes noires représentent les signaux de contrôle, les lignes rayés en jaune, les signaux audios.	24
3.11	Capture d'écran d'un patch Pure Data de synthèse granulaire. Les lignes fines représentent les signaux de contrôle, les lignes épaisses, les signaux audios.	25
4.1	Algorithme de résolution	27
4.2	Limites de l'élongation : travail de la force de rappel illimité et avec bornes.	30
4.3	Puissance de l'élongation : <i>Haut</i> : travail de la force de rappel pour $P = 1, 2$ et -2 . <i>Bas</i> : Rigidité classique et implémentation de l'équation de Cooper [CM97] à l'aide de deux liaisons.	31
4.4	Liaison orientée par rapport au plan xy	32
4.5	Exemple d'utilisation complète de l'objet <code>msd2D</code> dans un patch Pure Data : création, calcul, attributs, paramétrage.	33
4.6	Exemple d'utilisation des messages <i>bang</i> et <i>attributs</i> suivis d'un routage vidéo et son dans Pure Data.	35
4.7	Réponse en fréquence théoriques (-) et pratiques (- -) pour $M =$ 100 et $K = 10$ et trois valeurs de la constante d'amortissement μ	36

5.1	Utilisation des modèles physiques pour le mapping [MH05]. . .	39
5.2	Synthèse additive : utilisation du modèle physique pour le mapping	40
5.3	Synthèse additive : Interface d'utilisation.	41
5.4	Synthèse additive : <i>Haut</i> : Spectrogramme du son généré. <i>Bas</i> : Vitesse en x et en y de la masse 25	41
5.5	Synthèse granulaire : Représentation du modèle	42
5.6	Synthèse granulaire : Spectrogramme du son généré	43
5.7	Modèle 13D : <i>Haut</i> : Spectrogramme du son généré par la gamme évolutive. <i>Bas</i> : Spectrogramme du son généré par son pur.	45
1	Fichier d'aide principal de l'objet msd2D.	48
2	Fichier d'aide des messages de création.	49
3	Fichier d'aide des messages de paramétrages.	49
4	Fichier d'aide des messages attributs	50
5	Fichier d'aide d'informations générales.	50

Chapitre 1

Remerciements

Je tiens en premier lieu à remercier mon maître de stage Cyrille Henry. Il a su me communiquer sa passion pour Pure data. Son attrait et ses idées pour la recherche musicale ont été très motivant durant ces 4 mois de stage.

Je remercie également Thierry Coduys de m'avoir accueilli avec chaleur à La kitchen. C'est une des rares entreprises privées française à investir dans la recherche musicale, cela mérite donc d'être signaler. Je remercie et salue aussi toute l'équipe de La kitchen et de Kitchen.kom, pour l'ambiance de travail unique!

Je tiens aussi à remercier deux développeurs de Pure Data qui ont passé du temps et de la sueur sur MSD. En premier Thomas Grill qui m'a fortement aidé à optimiser le code mais aussi Frank Barknecht pour son éditeur graphique dédié à msd et ses nombreux conseils.

Et merci à Elodie pour le soutien moral quotidien, important dans 20m²!

Chapitre 2

Introduction

2.1 Objectif du stage

Dès l'apparition de l'ordinateur, les modèles physiques ont été utilisés pour le calcul numérique. Ils ont pour objectif la simulation de situations réelles par résolution des équations simplifiées correspondantes. Ce formalisme concerne tous les domaines de la physique, mais une avancée particulière a été effectuée dans les domaines de la synthèse audio et vidéo par Claude Cadoz [C79] durant son doctorat. Celui-ci a formalisé le système masse-ressort, appelé particule, pour une utilisation numérique. Ce système présente l'avantage d'être à la fois très simple, d'un point de vue calculatoire, mais aussi complexe puisqu'il permet de coupler les particules entre elles pour réaliser des structures. Cadoz effectuait déjà ce couplage et faisait ensuite résonner l'ensemble en lui appliquant des forces.

L'objectif de ce stage est d'utiliser ces modèles pour le contrôle des paramètres de synthèse audio ou vidéo. Le monde de l'informatique musicale possède aujourd'hui de nombreux et performants algorithmes de synthèse. Plus de cinquante ans de travaux ont en effet permis d'arriver à des résultats convaincants. L'un des problèmes majeurs reste l'utilisation de ces algorithmes en temps réel. Le nombre de paramètres à contrôler est en effet gigantesque pour un seul instrumentiste, il est donc nécessaire d'avoir une étape de *mapping*, durant laquelle le facteur de l'instrument électronique décide des rapports entre l'interface de l'instrumentiste et l'algorithme de synthèse. Ce document présente l'utilisation des modèles physiques dans le but de réaliser ce mapping.

Les logiciels PD [P96] et Max/MSP [PZ90] offre un environnement de travail adapté à nos besoins. Ils permettent l'intégration d'objets externes simplement. Ils sont en temps-réel et possèdent de nombreux outils afin de manipuler son et vidéo. Ce travail s'est particulièrement déroulé sous Pure Data pour des questions de puissance, de stabilité et de philosophie.

2.2 La kitchen

2.2.1 Lieu de création et de développement artistique

La kitchen est une entreprise privée fondée en 1999 par Thierry Co-duys. C'est un lieu de rencontres, d'échanges et de discussions, un espace de création s'appuyant sur des ustensiles toujours à la pointe de la technologie.

La kitchen est avant tout un lieu de recherche et de création artistique. Elle est une plate-forme technologique, un lieu de questionnement dédié en large partie à la collaboration et à la production artistique. La kitchen veut permettre une nouvelle relecture du mot "Multi-Art" : la musique, la vidéo, les arts vivants, les arts plastiques, la technologie et la recherche sont pensés et intégrés comme un unique paradigme.

Composée de quelques employés pluridisciplinaire, la kitchen propose un savoir-faire en informatique, en électronique ainsi qu'en ingénierie sonore et visuelle, et le met à disposition des créateurs dont les oeuvres sont de plus en plus tournées vers la haute technologie. Elle s'investit dans des projets très variés comme la captation gestuelle pour le spectacle vivant, les installations sonores interactives ou le développement du logiciel IanniX inspiré de l'UPIC de Iannis Xenakis.

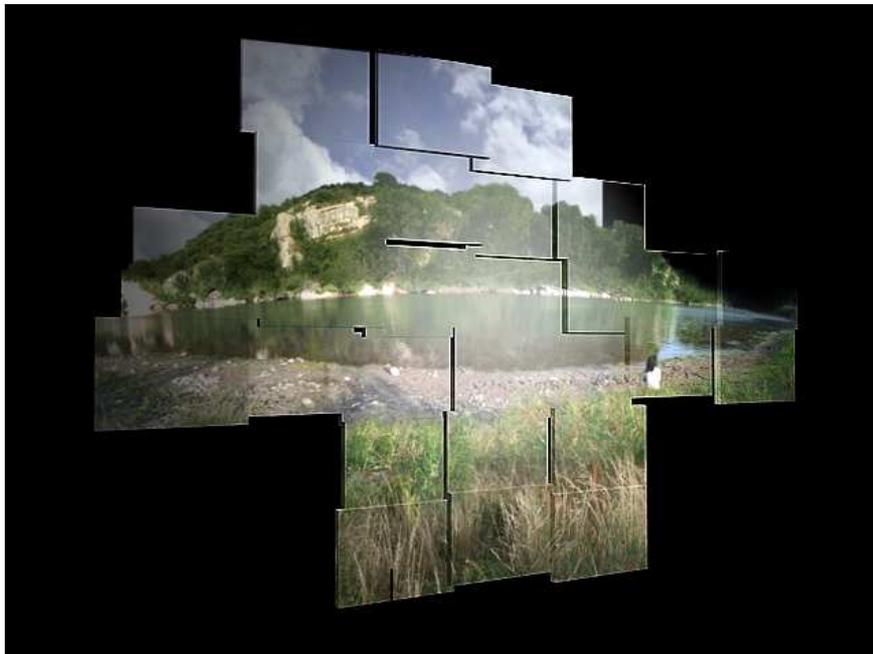


FIG. 2.1 – *Intime*, pièce interactive réalisée par La kitchen.

2.2.2 La recherche dans les domaines de l'innovation et des hautes technologies

La kitchen est profondément impliquée dans la recherche de nouveaux métissages entre structures électroniques et corporelles. Ses ingénieurs recherchent de nouveaux modes de communication dans notre univers technologique quotidien.

Dans une optique d'échanges et d'interactions, La kitchen développe de nouvelles plateformes artistiques, et oeuvre vers une relation constamment renouvelée avec des artistes en leur apportant des solutions techniques innovantes.

Kitchen.Lab, le laboratoire de la kitchen, met en oeuvre sur demande toute la partie technique nécessaire à la réalisation de projets artistique tournés vers les nouvelles technologies.

Tout en développant divers systèmes de captation sur mesure, Kitchen.Lab fabrique et commercialise aujourd'hui en France et à l'étranger deux types d'interfaces gestuelles pour des applications temps-réel (le Kroonde Gamma et le Toaster Gamma).

Chapitre 3

Contexte scientifique

3.1 Modèle masse ressort amortisseur

Le modèle physique particulière est défini autour de deux éléments :

- la *masse* : la masse est un élément ponctuel, auquel on associe la notion de particule. C'est un système qui possède une entrée en force F et une sortie en position x . Elle est décrite par un seul paramètre physique : sa masse M .
- la *liaison* : la liaison assemble les particules selon une certaine topologie. Physiquement, elle représente la mise en parallèle d'un élément de rappel de constante de raideur k et d'un élément de frottement de constante μ . Elle admet deux positions en entrée et résulte en deux forces opposées.

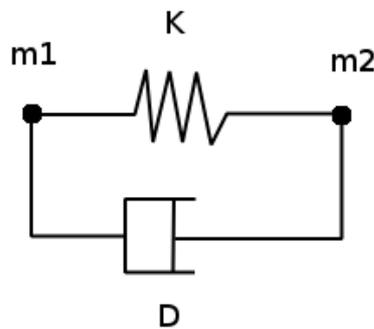


FIG. 3.1 – Système simple représentant deux masses liées par une liaison ressort-amortisseur

3.1.1 Équation de la masse

Système à 1 degré de liberté

Si les masses possèdent un seul degré de liberté, l'entrée du système est alors une force $F_{x_i}(t)$ selon l'axe x à l'instant t et la sortie est la position $x_i(t)$ de la masse i . En partant de l'équation fondamentale de la dynamique :

$$F_i = m_i \gamma_i \quad (3.1)$$

où F_i est la résultante des forces s'exerçant sur la masse i de masse m_i et d'accélération γ_i , qui s'écrit en projetant sur l'axe x :

$$F_{x_i}(t) = m_i \frac{d^2 x_i(t)}{dt^2} \quad (3.2)$$

où $x_i(t)$ est la position de la masse i à l'instant t .

Extension du modèle à 2 et 3 dimensions

L'extension du modèle à 2 ou 3 dimensions se fait simplement en remplaçant la force F_{x_i} par la projection de la force F sur les axes y et z soit :

$$F_{y_i}(t) = m_i \frac{d^2 y_i(t)}{dt^2} \quad (3.3)$$

$$F_{z_i}(t) = m_i \frac{d^2 z_i(t)}{dt^2} \quad (3.4)$$

3.1.2 Equation de la liaison

Système à 1 degré de liberté

L'entrée de l'élément liaison à une seule dimension est une position $x_i(t)$ à un instant t et la sortie est une force $F(t)$ qui est égale à la projection sur l'axe x de cette même force. La liaison possède trois propriétés physiques :

- la longueur au repos de l'élément ressort : L_0
- la constante de raideur de l'élément ressort : k
- la constante de viscosité de l'élément de frottement : μ

Sur une dimension la longueur de la liaison est donnée à l'instant t par :

$$L(t) = |x_1(t) - x_2(t)| \quad (3.5)$$

où $x_1(t)$ et $x_2(t)$ sont les positions des masses liées. La force de rappel sur la masse 1 est proportionnelle à l'élongation du ressort

$$F_{k1}(t) = -k(L(t) - L_0) \quad (3.6)$$

et la force de frottement est proportionnelle à la variation de longueur

$$F_{\mu 1}(t) = -\mu \frac{dL(t)}{dt} \quad (3.7)$$

La mise en parallèle des éléments ressort et amortisseur résulte en une force F_1 égale à la somme des deux forces soit :

$$F_1(t) = -k(L(t) - L_0) - \mu \frac{dL(t)}{dt} \quad (3.8)$$

tandis que la force exercée sur la masse 2 est l'opposée de la force F_1

$$F_2(t) = -F_1(t) \quad (3.9)$$

Extension du modèle à 2 et 3 dimensions

Le calcul de la longueur à deux ou trois dimensions se fait avec la distance euclidienne, soit :

$$L(t) = \sqrt{(x_1(t) - x_2(t))^2 + (y_1(t) - y_2(t))^2} \quad (3.10)$$

en 2 dimensions, et :

$$L(t) = \sqrt{(x_1(t) - x_2(t))^2 + (y_1(t) - y_2(t))^2 + (z_1(t) - z_2(t))^2} \quad (3.11)$$

en 3 dimensions. Les force F_{x1}, F_{y1}, F_{z1} , sont alors égales aux projections de la force $F_1 = F_{K1} + F_{\mu1}$ sur les axes correspondants. Soit

$$F_{x1}(t) = F_1(t) \frac{x_1(t) - x_2(t)}{L(t)} \quad (3.12)$$

$$F_{y1}(t) = F_1(t) \frac{y_1(t) - y_2(t)}{L(t)} \quad (3.13)$$

$$F_{z1}(t) = F_1(t) \frac{z_1(t) - z_2(t)}{L(t)} \quad (3.14)$$

3.1.3 Oscillation d'un système masse-ressort-amortisseur

Oscillations libres

Le système masse-ressort-amortisseur idéal est régi par :

$$m \frac{d^2x}{dt^2} + \mu \frac{dx}{dt} + kx = 0 \quad (3.15)$$

où $x = L - L_0$ est l'élongation. Cette équation différentielle du second degré admet des solutions de la forme

$$x = e^{-\alpha t} A \cos(\omega_d t + \phi) \quad (3.16)$$

où

$$\alpha = \frac{\mu}{2m}$$

$$\text{et } \omega_d = \sqrt{\omega_0^2 - \alpha^2} \text{ tel que } \omega_0 = \sqrt{\frac{k}{m}}$$

correspondent à la constante d'amortissement et à la pulsation caractéristique du système. A et ϕ dépendent des conditions initiales d'excitation. Le système est oscillant si ω_d est réel, ie $k/m > \alpha^2$.

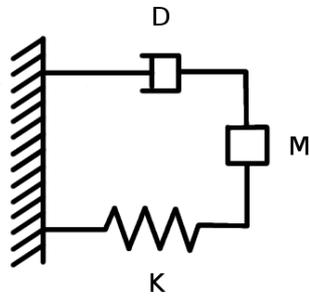


FIG. 3.2 – Système masse-ressort-amortisseur idéal

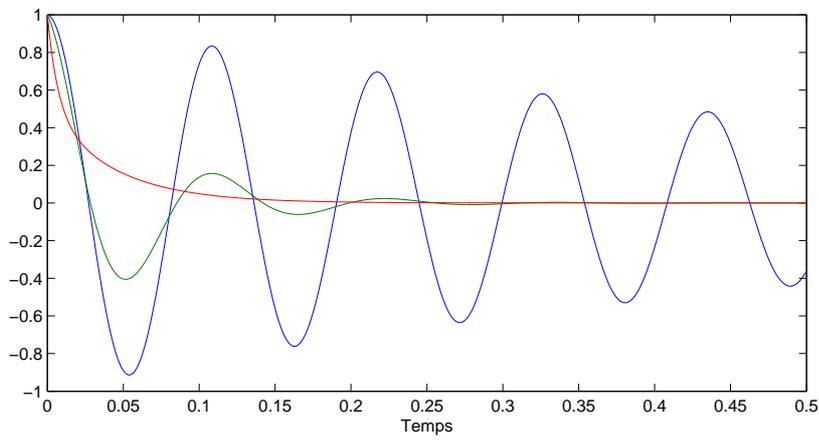


FIG. 3.3 – Exponentielle amortie pour 3 valeurs de la constante d'amortissement.

Oscillations forcées

Lorsque le système est soumis à une force en entrée $f(t)$, l'équation devient :

$$m \frac{d^2x}{dt^2} + \mu \frac{dx}{dt} + kx = f(t) \quad (3.17)$$

Le système peut alors être décrit fréquentiellement par la transformée de Laplace, qui nous donne (si $x(0) = 0$ et $v(0) = 0$) :

$$X(s)[ms^2 + \mu s + k] = F(s) \quad (3.18)$$

La réponse du système en déplacement par rapport à la force est alors (fig 3.4) :

$$\frac{X(s)}{F(s)} = \frac{1}{ms^2 + \mu s + k} \quad (3.19)$$

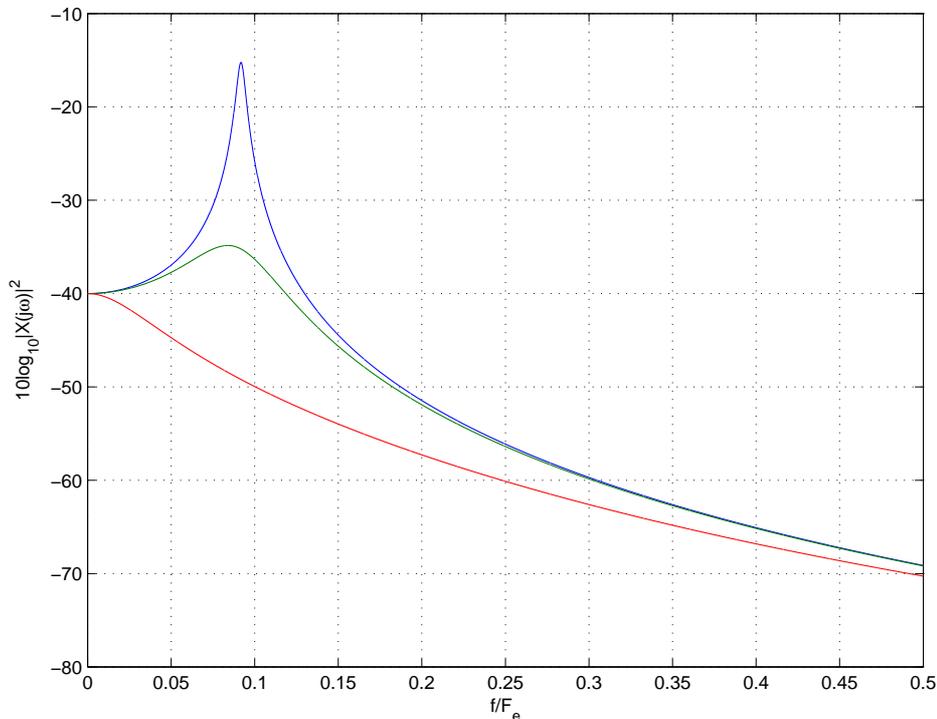


FIG. 3.4 – Réponse en fréquence théorique du système masse ressort amortisseur pour 3 valeurs de la constante d'amortissement.

3.2 Modèles physiques particuliers

Les recherches sur les modèles physiques particuliers ont débutées avec les travaux de Cadoz sur CORDIS-ANIMA [CLF93] et son environnement GENESIS [CC02] (fig. 3.5). C'est un système de modélisation et de simulation

numérique d'objets en temps-réel. Ce système a été construit à l'ACROE en plusieurs étapes depuis 1978. Il constitue le noyau d'un outil de création pour la musique sur ordinateur (Cordis) et les images animées (Anima). Il a été développé pour fournir des reproductions d'objets physiques issus de la réalité, qui peuvent être des sources sonores aussi bien que des objets visibles. L'objectif du système est la simulation complète. C'est à dire que la simulation doit non seulement rendre compte des aspects extérieurs de l'objet (aspects visuels, acoustiques, tactiles et gestuels) mais aussi de sa structure et de son comportement. Le modèle implémenté n'est donc pas une synthèse des effets de l'objet mais bien une synthèse de son état interne par modélisation physique. Le moteur physique de Cordis-Anima est basé sur deux éléments : les masses et les interactions. Les masses sont les éléments ponctuels qui permettent de définir la forme de la structure, tandis que les interactions représentent les forces qui s'exercent sur ces masses. Genesis est un environnement qui permet la manipulation d'objets masses-interactions dans le système Cordis-Anima à plus ou moins bas-niveau. Il fonctionne dans deux modes : le mode lutherie et le mode jeu. Le mode lutherie permet de construire une structure en assemblant des particules. Le mode jeu est dédié au "jeu" en lui-même, il consiste donc en l'édition de signaux temporels.

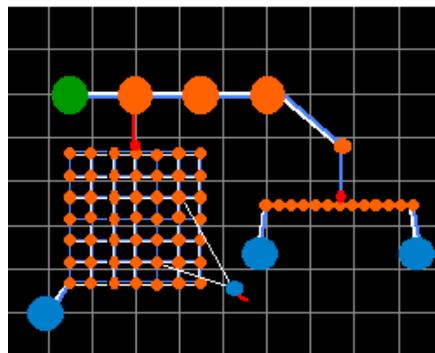


FIG. 3.5 – GENESIS en mode lutherie [CC02].

Dans le paysage des modèles physiques particulières pour la synthèse, on trouve également le logiciel TAO, qui a évolué vers CYMATIC [HR03] (fig. 3.6).

Cymatic est lui aussi un environnement qui permet l'édition d'objets masses-particules. L'utilisateur construit un instrument virtuel en plaçant graphiquement des éléments individuels comme des feuilles, des cordes ou des solides. Ces éléments sont interconnectés en liant individuellement les masses ; les extrémités peuvent également être fixées. Les paramètres de tension et de masses de chaque élément sont définis à l'initialisation ou changent dynamiquement pendant la synthèse. Des signaux d'excitations sont alors appliqués individuellement sur les masses et le signal audio peut être récupéré sur n'importe laquelle d'entre elle. Cymatic a été développé pour permettre l'utilisation des modèles physiques selon deux directions :

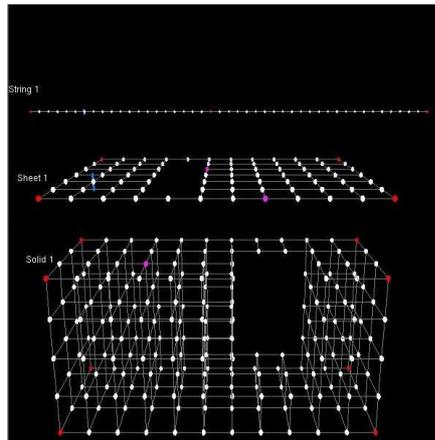


FIG. 3.6 – CYMATIC [HR03].

- l'utilisation pour la synthèse sonore afin de créer des sons qui sont plus proches des entités physiques.
- l'utilisation pour l'interface de contrôle afin de permettre au musicien d'interagir avec l'instrument électronique de manière plus intuitive et plus approfondie.

Un contrôle à retour de force de Cymatic est possible, notamment grâce aux interfaces développées pour les jeux vidéos qui sont supportées par le système.

3.3 Mapping entre les paramètres de contrôle et les paramètres de synthèse

3.3.1 Définition et importance du mapping

Dans un instrument acoustique, l'interface de jeu est intrinsèquement liée à la source sonore. Le résonateur de la flûte, par exemple, fait partie du mécanisme de contrôle aussi bien que du générateur sonore. Les liens qui déterminent l'influence de l'un sur l'autre sont déterminés par des lois physiques complexes que l'instrumentiste apprend à contrôler durant ses années d'études. Le son produit porte en lui les caractéristiques du geste instrumental qui en est la cause.

Avec les instruments de musiques électroniques et informatiques, la situation est radicalement différente. L'interface est généralement un équipement indépendant de la source sonore. Les rapports entre eux doivent alors être explicitement définis. Ce domaine de recherche peut être divisé en deux champs : le contrôle gestuel et le mapping.

Les études sur le contrôle gestuel cherche à comprendre et à interpréter le geste instrumental. Le développement historique des contrôleurs est lié au standard MIDI d'échanges de données musicales. Ce format est basé sur le

système du clavier et permet de coder simplement la hauteur, la durée et la force initiale des notes jouées. Ce système est donc adapté, par exemple, au contrôle du piano et il a permis le développement de nombreux contrôleurs de ce type appelés claviers MIDI. Le problème se pose lorsque ces claviers sont utilisés pour synthétiser des sources dont le mécanisme de production sonore est fondamentalement différent de celui de la corde frappée : les instruments à vent ou les cordes frottées par exemple.

Dans le piano, les notes sont des événements isolés les uns des autres et dépendent de gestes différents associant une note par doigt. Au contraire dans les instruments à sons entretenus, une phrase musicale est souvent exécutée dans le même geste [PO03]. Cette continuité du geste permet non-seulement aux notes d'être jouées liées, mais elle est aussi déterminante pour le mode de jeu. A la différence des pianistes, ces instrumentistes sont en contact direct avec la structure vibrante génératrice : corde, anche ou colonne d'air, ce qui leur permet un contrôle fin du phénomène acoustique à sa source.

Pour simuler ce type de situation, les chercheurs, et maintenant les industriels, travaillent sur les systèmes à retour de force. Ces contrôleurs disposent d'un ou plusieurs moteurs qui permettent de résister ou d'aider le mouvement de l'utilisateur. Leur efficacité pour le contrôle musical réside dans le fait qu'en plus du retour acoustique qu'a un musicien sur son instrument, il existe également un retour physique aussi important dans les données qui lui permettent de contrôler l'instrument. Le contrôle d'un être humain est beaucoup plus précis en force qu'en position, un tel système qui répond à la force exercée peut donc être utilisé avec plus de finesse.

Cette problématique de recherche s'intéresse donc à la pertinence d'un contrôleur par rapport à l'instrumentiste. Un bon contrôleur doit permettre un contrôle expressif et précis sur un large éventail de synthèse. On s'intéressera dans ce document principalement à l'étape suivante dans la chaîne de synthèse, le mapping.

Le mapping est la manière de définir les rapports entre paramètres de contrôle et paramètres de synthèse (fig. 3.7). Il peut être de 3 types selon Rovin et al.[RWDD97] :

- *Un-à-Un* : Chaque paramètre de contrôle est assigné à un paramètre de synthèse. C'est le schéma de mapping le plus simple, mais généralement le moins expressif.
- *Mapping Divergent* : Un paramètre de contrôle est utilisé pour contrôler plusieurs paramètres de synthèse. Ce mapping permet un contrôle au niveau macroscopique du son, mais appliqué seul, il ne permet pas un accès aux paramètres internes du son.
- *Mapping Convergent* : Plusieurs paramètres de contrôles sont utilisés pour produire un paramètre musical. Ce schéma nécessite une expérimentation préalable pour arriver à des résultats pertinents à l'utilisation.

Nous utiliserons une terminologie plus générale, celle de mapping complexe. Cette appellation regroupe les 3 mappings précédemment cités mais également

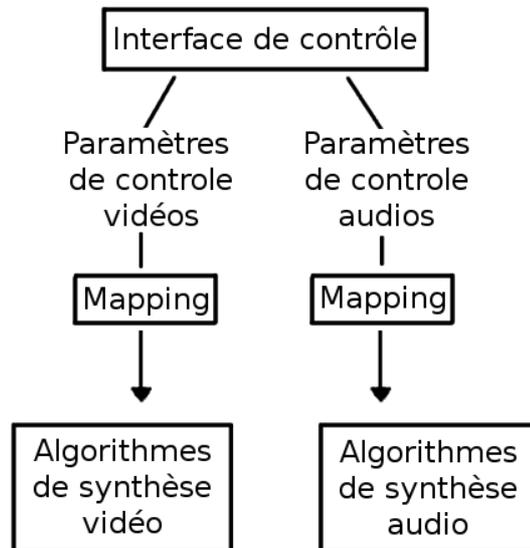


FIG. 3.7 – Schéma classique de dispositif de synthèse sonore et visuelle.

tous les mappings moins spécifiques de type *plusieurs-vers-plusieurs*.

Le mapping a une importance déterminante sur le sentiment et les émotions qu'a un instrumentiste en jouant, ce qui signifie que le mapping peut changer radicalement un instrument. Les expériences réalisées par Hunt et al. [HWP02] montre qu'avec un instrument très simple (2 paramètres, 2 potentiomètres et un oscillateur) et des mappings différents, les mêmes sujets ont des sensations différentes. Avec un mapping simple, un potentiomètre est relié à chacun des paramètres, les sujets comprennent le fonctionnement de l'instrument rapidement mais celui-ci les lasse vite. Avec un mapping complexe, les paramètres renvoyés par les potentiomètres affectent le générateur après un traitement, les sujets ont beaucoup de mal à le manier au début de l'expérience. Cependant, après un temps d'adaptation ils "apprennent" à utiliser l'instrument, point important détaillé plus loin dans le document.

3.3.2 État de l'art

Les travaux sur le thème du mapping peuvent être classés en deux catégories en fonction de la spécificité de l'algorithme de synthèse.

Mapping pour la simulation d'instruments traditionnels

Rovan *et al.* [RWDD97] ont montré que dans le contexte d'une interaction musicale experte, le dispositif d'entrée doit reproduire certaines caractéristiques de l'instrument original. L'exemple spécifique présenté est l'embouchure de clarinette. En effet, l'anche d'une clarinette a des propriétés non-linéaires bien connues des physiciens. Pour contrôler l'algorithme de synthèse de clarinette, ils expérimentent trois mappings différents, un mapping un-a-

un, un mapping convergent simple et un mapping convergent complexe appliqué entre un contrôleur pour instrument à vent et une synthèse additive. En étudiant les potentiels de chaque mapping, il en ressort que le mapping convergent complexe qui implémente le plus de caractéristiques d'une vraie clarinette, possède le plus haut niveau d'expressivité. La caractéristique la plus importante à reproduire est la déviation initiale d'un paramètre qui correspond dans la réalité à la pression qu'exerce le clarinettiste sur l'anche avant de produire un son.

Wessel a proposé d'utiliser un espace de timbre [W79], selon la métaphore simple qui place les instruments aux timbres similaires de manière rapprochée dans cet espace, la distance exacte pouvant être définie par des expériences de jugements de similarités. Le mapping est appliqué entre le contrôle et les axes de l'espace. En réalisant ces expériences, Wessel s'est rendu compte que deux dimensions étaient principalement pertinentes au niveau du timbre : l'attaque et la richesse spectrale. Ce résultat permet donc d'envisager un contrôle en 2 dimensions de l'espace timbral. L'auteur montre que l'expressivité dépend énormément de la stratégie de mapping employée. En utilisant la même configuration, des différences d'expressivité ont été remarquées, en fonction du mapping choisi.

Un point intéressant suggéré dans cet article est l'adaptation de la stratégie de mapping au niveau de l'utilisateur. Des débutants pourrait tirer profit d'un mapping simple, alors que des musiciens confirmés auraient avantage à utiliser des mappings plus complexes.

Comme conclusion directe de cette étude, on peut considérer le mapping comme fournissant *le degré d'interactivité* entre l'utilisateur et la machine. On peut alors définir une hiérarchie des possibilités offertes par le mapping employé allant d'un contrôle macroscopique (phrases, rythme) à un contrôle microscopique (caractère précis du timbre).

On notera que les recherches sur ce thème fonctionne bien mieux sur les instruments initialement interfacés. La baguette de batterie ou l'archet des cordes frottées permettent la captation d'un geste musical adapté à l'analyse et au routage vers un algorithme de synthèse.

Mapping générique

Dans le cas d'instruments interactifs qui n'ont aucun équivalent acoustique, des formalismes doivent être proposés.

Mulder *et al.* [MFM97] ont proposé d'utiliser les multiples capacités de la main pour générer des paramètres de synthèse. Ils postulent que les sensations fournies par le toucher peuvent être remplacées par un feedback acoustique. Ils génèrent alors une surface virtuelle (existante uniquement au niveau des données), que l'instrumentiste va pouvoir manipuler selon les techniques ha-

bituelles de la main dont les paramètres de forme, de position et d'orientation sont relevés grâce à des gants. Les paramètres de la forme virtuelle contrôlent alors les paramètres de synthèse.

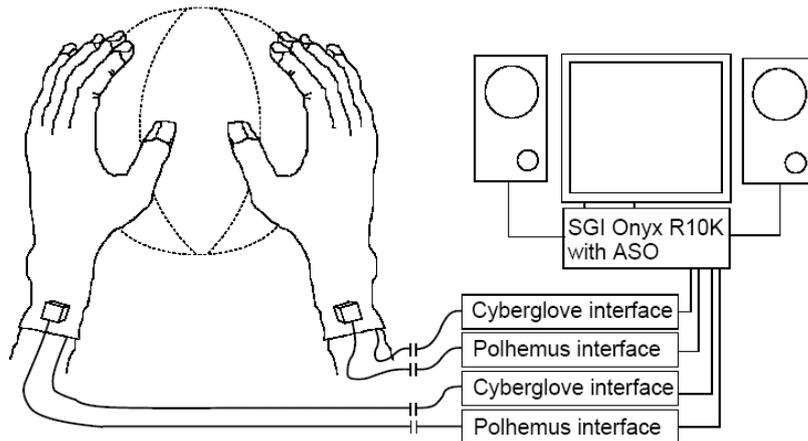


FIG. 3.8 – Forme virtuelle utilisée pour le mapping [MFM97].

Wessel & Wright [WW02] listent un certain nombre de métaphores gestuelles ou cognitives qui permettent à un instrumentiste de mieux se situer dans le contrôle abstrait de processus :

- Espaces Musicaux de sons et de processus : selon cette métaphore, les sons les plus similaires sont placés côte à côte dans une représentation spatiale. Un espace à une dimension permet par exemple de se déplacer selon une hauteur, ce qui est réalisé de manière discrète sur le piano mais des schémas à plusieurs dimensions ont été proposées pour ce type de représentations.
- "Glisser déposer" : cette métaphore exprime une forme de contrôle souvent rencontrée chez les usagers informatiques : un objet est sélectionné, attrapé, et déposé sur un processus. Les objets peuvent être une collection d'échantillons sonores et les processus une manière de les jouer : en boucle, à l'envers, transposé, ... mais les processus peuvent également être des traitements musicaux : un filtrage, une enveloppe.
- Frottement : Selon Wessel & Wright, ce type de métaphore est intéressante pour une manipulation temporelle. Sur des modèles qui permettent une manipulation d'échelle temporelle sans changement de hauteur (synthèse granulaire, sinus), un frottement peut être mappé sur l'indice temporel du processus. Il nécessite une très grande précision sur une dimension.
- Immersion : dans la métaphore immersive, l'ordinateur génère en permanence tout le matériel musical, mais ce matériel est silencieux par défaut. L'instrumentiste contrôle le volume de chaque processus. Par

exemple, en utilisant un capteur de pression multi-point où chaque zone correspond au volume d'un processus. Ce type d'interface est très intéressant pour orchestrer une composition en temps-réel en sélectionnant quel processus va être entendu.

- "Attraper et Lancer" : cette métaphore implique de piéger sélectivement du matériau musical au cours d'une performance, de le traiter, et de le relancer dans la performance. Le matériau peut être de différents types : données de contrôles comme du MIDI ou audios comme la densité spectrale dans une certaine bande de fréquence. Le traitement appliqué est critique pour ce genre de performance, un simple écho devenant vite redondant.

Momeni & Wessel [MW03] utilisent plusieurs espaces afin de contrôler du matériel musical en temps-réel. Dans le logiciel Max, ils réduisent cet espace de paramètres à 2 dimensions afin de permettre à l'utilisateur d'évoluer de manière intuitive dans l'espace associé au matériel.

Wanderley *et al.* propose un système générique, ESCHER, comprenant toute la chaîne de traitement et de synthèse suivant le contrôleur [WSR98]. Dans ESCHER, l'utilisateur choisit un contrôleur et une méthode de synthèse, puis il sélectionne une stratégie de mapping appropriée. Pour réaliser cette polyvalence, le système utilise un mapping à deux couches. La première couche est une couche d'adaptation entre les données de l'interface de contrôle et un jeu de paramètres abstraits définis par l'utilisateur durant la création de l'instrument. La deuxième couche de mapping transforme ces D paramètres abstraits en paramètres de synthèse en utilisant une interpolation dans un espace à D dimensions.

Hunt a prouvé à plusieurs reprises [HWP02] [HWK00] que les mappings complexes pour les interfaces génériques sont plus intéressants à long terme que les mappings simples. Il étudie l'utilisation de mêmes interfaces avec des mappings différents et analyse les résultats obtenus, par exemple, en donnant pour tâche au sujet d'imiter un son et en mesurant l'écart entre le son produit et l'original. Trois points ressortent de ses travaux :

- les mappings complexes ont des scores plus élevés que les autres exceptés pour les tests les plus simples.
- le score des utilisateurs de mappings complexes augmente rapidement avec le temps.
- les scores sont meilleurs pour les tests complexes.

Les complexités des tests exprimés ici se réfèrent à la complexité du son à obtenir. Un sinus pur est qualifié de test simple alors qu'un son créé par synthèse FM évoluant temporellement est qualifié de test complexe. En dehors des aspects quantitatifs de ces travaux, un point qualitatif important est analysé par Hunt : l'augmentation du score avec le temps. Ce phénomène rapproche en effet l'instrument électronique de ses homologues acoustiques. L'instrument nécessite une étape d'apprentissage durant laquelle l'utilisateur

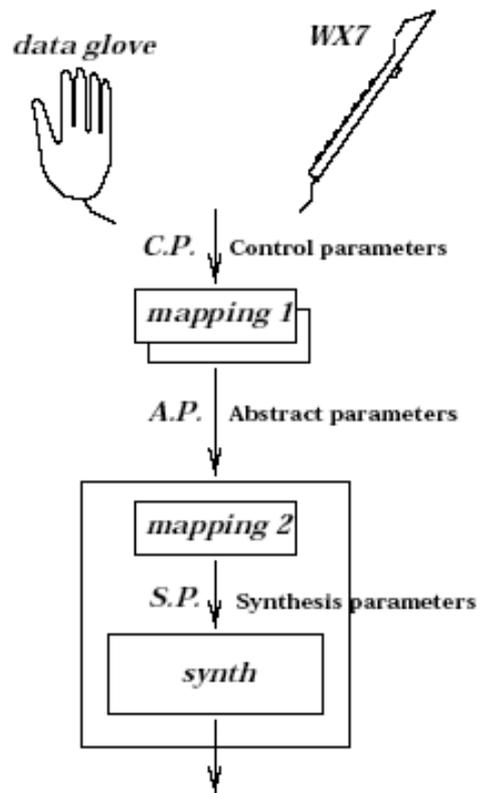


FIG. 3.9 – ESCHER [WSR98]

apprend à manier les contrôles pour obtenir les sons désirés. Ce processus d'apprentissage est gratifiant puisqu'il récompense l'apprenti en lui permettant de plus en plus de possibilités. Il l'incite à poursuivre son travail afin d'approfondir sa compréhension de l'instrument au contraire d'un mapping un-à-un où les sujets devinent le rôle de chacun des paramètres et se lassent vite de l'utiliser.

3.3.3 Extension au mapping de la synthèse vidéo

Depuis longtemps, les chercheurs et les artistes travaillent à coupler le son et l'image. Ce couplage est détecté dès l'enfance comme présentant un intérêt supérieur à une source audio ou visuelle isolée. Il présente un intérêt majeur puisqu'il crée un tout différent de la somme du son et de l'image. Michel Chion [C98] écrit à ce sujet :

Par de nombreux exemples, des analyses et des expériences, nous avons démontré qu'on ne peut pas étudier le son d'un film séparément de son image et inversement. En effet, leur combinaison produit quelque chose d'entièrement spécifique et nouveau, analogue à un accord ou un intervalle en musique.

C'est la notion de *valeur ajoutée*. Il semble donc intéressant d'appliquer nos réflexions sur le mapping audio directement au mapping vidéo (fig. 3.7).

3.4 Environnements temps-réel pour le traitement du signal : Pure Data et Max/MSP

Pure Data et Max/MSP sont deux logiciels provenant du même noyau. Ce noyau a été développé par Miller Puckette dans les années 1980. Aujourd'hui, les deux logiciels suivent des évolutions différentes. Max/MSP est un logiciel commercial et donc fini tandis que Pure Data est un logiciel libre en perpétuel développement, adapté à la recherche. Bien que les définitions formelles de Max et PD soient rares (excepté [S04]), il est communément admis que ce sont des environnements de programmation graphique dédiés au temps-réel.

L'environnement de travail s'appelle un *patch*. C'est une feuille, initialement vierge, dans laquelle l'utilisateur dépose des objets virtuels représentant des fonctions. Ces objets sont ensuite reliés entre eux afin de faire transiter des données. Ils sont de trois types :

- primitive / externe : ce sont des objets codés en C, qui accomplissent une fonction bien définie. Le terme primitive désigne les objets inclus dans le logiciel tandis que les externes sont rajoutés par des programmeurs.
- abstraction : c'est un objet qui correspond à une fonction programmée dans un autre patch. On peut donc imbriquer les patches les uns dans les autres.

- sous-patch : c'est seulement une partie du patch rendue invisible pour des questions de clarté.

Ces données peuvent soit provenir d'un élément extérieur (autre logiciel, interface) soit être directement synthétisées dans le logiciel. Deux types de données sont utilisées :

- les signaux de contrôle, qui permettent l'édition de paramètres. Ils transitent de manière ponctuelle ou à une fréquence lente qui dépasse rarement 1 kHz.
- les signaux audios, qui transportent un flux continu de données au format audio de la carte son utilisée. Un format standard étant une fréquence de travail de 44.1 kHz sur des données de 16 bits.

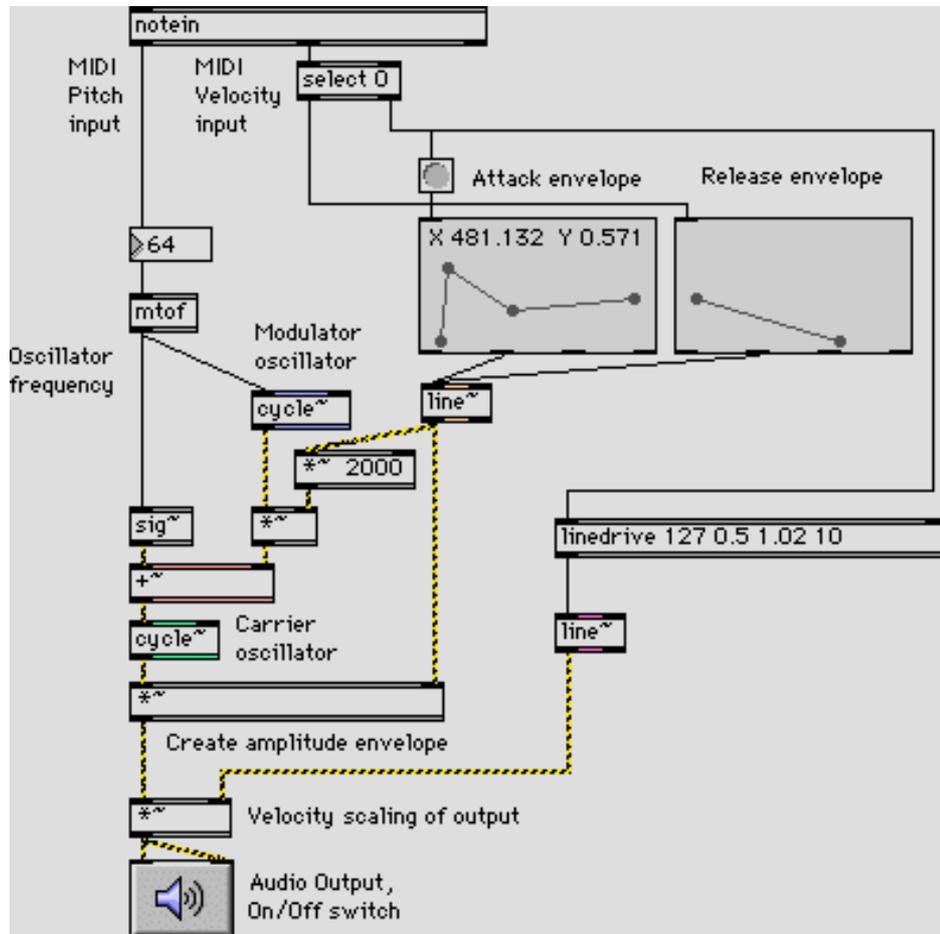


FIG. 3.10 – Capture d'écran d'un patch Max/MSP de synthèse FM. Les lignes noires représentent les signaux de contrôle, les lignes rayés en jaune, les signaux audios.

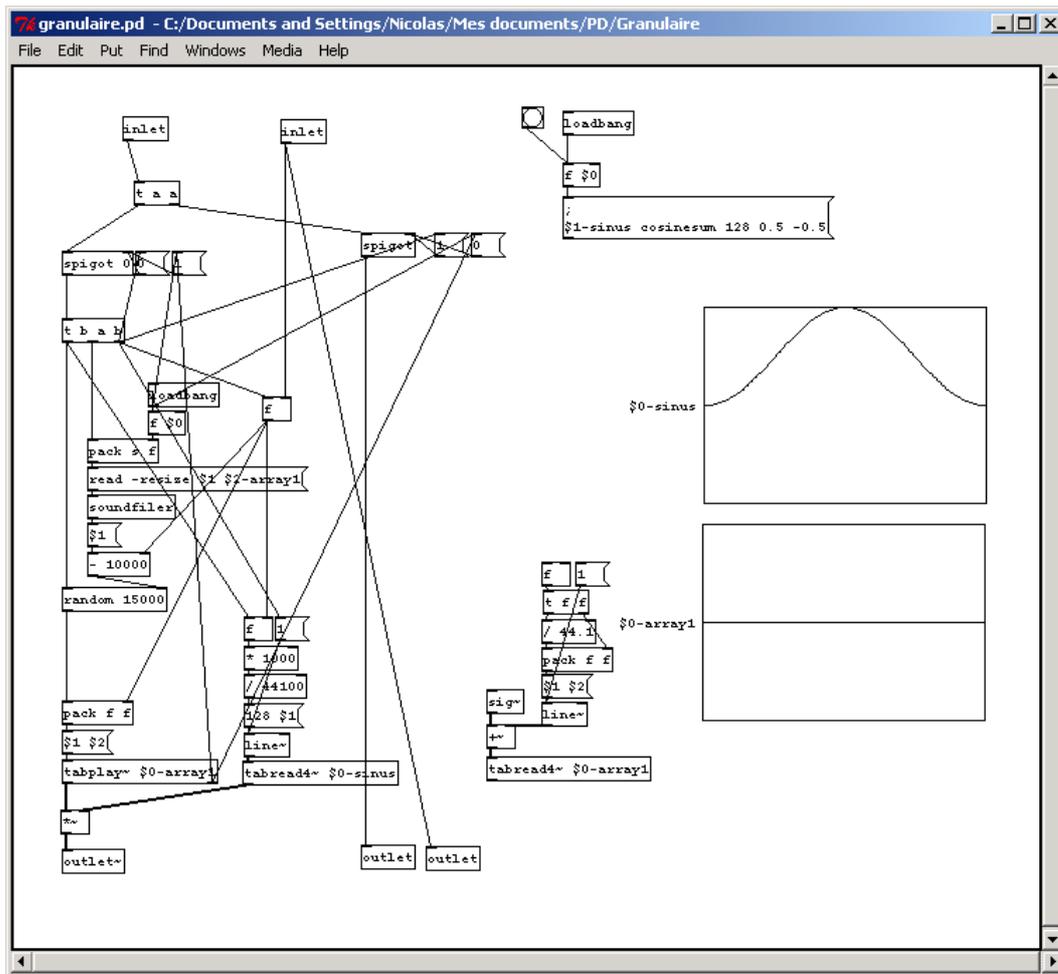


FIG. 3.11 – Capture d'écran d'un patch Pure Data de synthèse granulaire. Les lignes fines représentent les signaux de contrôle, les lignes épaisses, les signaux audios.

Chapitre 4

Implémentation numérique : la collection d'objets externes MSD

”When new instrument will allow me to write music as I conceive it, taking the place of the linear cunterpoint, the movement of sound masses, or shifting planes, will be clearly perceived. When these sound masses collide the phenomena of penetration or repulsion will seem to occur.”
 - Edgard Varèse, 1962

Pour pouvoir expérimenter l'utilisation des modèles physiques pour le mapping, il a été nécessaire d'implémenter une librairie spécifique : la librairie MSD, mass-spring-damper.

4.1 Présentation du modèle

Le modèle particulaire est défini autour des éléments de masse et de liaisons.

La *masse* : la particule est un système qui possède une entrée en force F et une sortie en position X . En trois dimensions, elle est décrite à l'instant n par :

- deux identifiants : un numéro et un nom
- sa masse M
- sa position $X[n] = (x[n] \ y[n] \ z[n])^t$
- sa vitesse $V[n] = (v_x[n] \ v_y[n] \ v_z[n])^t$

La *liaison* : la liaison associe deux masses, elle possède donc deux entrées en positions et deux sorties en forces. Elle est décrite à l'instant n par :

- deux identifiants : un numéro et un nom
- sa rigidité k
- sa longueur au repos L_0
- sa constante d'amortissement μ

- les 2 numéros des masses m_1 et m_2
- sa longueur minimale L_{min} et maximale L_{max}
- son orientation $X_a = (x_a[n] \ y_a[n] \ z_a[n])^t$, $X_b = (x_b[n] \ y_b[n] \ z_b[n])^t$
- sa puissance P

Tous ces paramètres sont détaillés dans le présent chapitre.

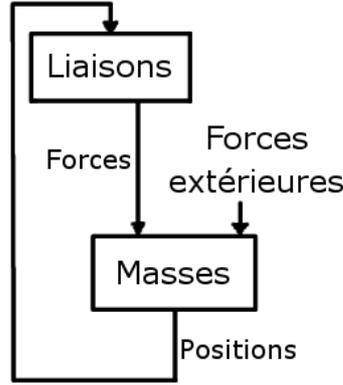


FIG. 4.1 – Algorithme de résolution

4.2 Equation de l'élément masse

4.2.1 Système à 1 degré de liberté

Si les masses possèdent un seul degré de liberté, l'entrée du système est alors une force $F_{x_i}(t)$ selon l'axe x à l'instant t et la sortie est la position $x_i(t)$ de la masse i . En partant de l'équation fondamentale de la dynamique :

$$F_i = m_i \gamma \quad (4.1)$$

où F_i est la résultante des forces s'exerçant sur la masse i de masse m_i et d'accélération γ , qui s'écrit en projetant sur l'axe x :

$$F_{x_i}(t) = m_i \frac{d^2 x_i(t)}{dt^2} \quad (4.2)$$

où $x_i(t)$ est la position de la masse i à l'instant t . On peut alors discrétiser l'équation selon le schéma des différences finies, qui donne pour la dérivée :

$$\frac{dx(t)}{dt} = \frac{x(nT_e) - x((n-1)T_e)}{T_e} \quad (4.3)$$

et donc pour l'accélération

$$\frac{d^2 x_i(t)}{dt^2} = \frac{x(nT_e) - 2x((n-1)T_e) + x((n-2)T_e)}{T_e^2} \quad (4.4)$$

où T_e est la période d'échantillonnage. Si l'on écrit $x[n] = x(nT_e)$ l'équation 4.1 devient

$$x_i[n] = -T_e^2 \frac{F_{x_i}[n]}{m_i} + 2x_i[n-1] - x_i[n-2] \quad (4.5)$$

Cette équation est la base de l'algorithme utilisé pour le calcul de la position de la masse i à l'instant n . En pratique le système est implémenté avec $T_e = 1$. A l'utilisateur de normaliser la valeur de la masse avec le carré de la fréquence d'échantillonnage utilisée par :

$$M_i = \frac{m_i}{T_e^2}$$

4.2.2 Extension du modèle à 2 et 3 dimensions

L'extension du modèle à 2 ou 3 dimensions se fait simplement en remplaçant la force F_{x_i} par la projection de la force F sur les axes y et z soit :

$$y_i[n] = \frac{-F_{y_i}[n]}{M_i} + 2y_i[n-1] - y_i[n-2] \quad (4.6)$$

$$z_i[n] = \frac{-F_{z_i}[n]}{M_i} + 2z_i[n-1] - z_i[n-2] \quad (4.7)$$

4.3 Equation de l'élément liaison

4.3.1 Système à 1 degré de liberté

Les entrées de l'élément liaison à une seule dimension sont deux positions $x_1(t)$ et $x_2(t)$ à un instant t et les sorties deux forces $F_1(t)$ et $F_2(t)$. La liaison possède, dans un premier temps, trois propriétés physiques :

- la longueur au repos de l'élément ressort : L_0
- la constante de raideur de l'élément ressort : k
- la constante de raideur de l'élément de frottement : μ

Sur une dimension la longueur de la liaison est donnée à l'instant $n = nT_e$ par :

$$L[n] = |x_1[n] - x_2[n]| \quad (4.8)$$

où $x_1[n]$ et $x_2[n]$ sont les positions des masses liées. La force de rappel sur la masse 1 est proportionnelle à l'élongation du ressort

$$F_{k1}[n] = -k(L[n] - L_0) \quad (4.9)$$

et la force de frottement est proportionnelle à la variation de longueur

$$F_{\mu 1}[n] = -\mu(L[n] - L[n-1]) \quad (4.10)$$

La mise en parallèle des éléments ressort et amortisseur résulte en une force F_1 égale à la somme des deux forces soit :

$$F_1[n] = -k(L[n] - L_0) - \mu(L[n] - L[n-1]) \quad (4.11)$$

tandis que la force exercée sur la masse 2 est l'opposée de la force F_1

$$F_2[n] = -F_1[n] \quad (4.12)$$

4.3.2 Extension du modèle à 2 et 3 dimensions

Le calcul de la longueur à deux ou trois dimensions se fait avec la distance euclidienne, soit :

$$L[n] = \sqrt{(x_1[n] - x_2[n])^2 + (y_1[n] - y_2[n])^2} \quad (4.13)$$

en 2D, et :

$$L[n] = \sqrt{(x_1[n] - x_2[n])^2 + (y_1[n] - y_2[n])^2 + (z_1[n] - z_2[n])^2} \quad (4.14)$$

en 3 dimensions. Les force F_{x1}, F_{y1}, F_{z1} , sont alors égales aux projections de la force $F_1 = F_{k1} + F_{\mu1}$ sur les axes correspondants. Soit

$$F_{x1}[n] = F_1[n] \frac{x_1[n] - x_2[n]}{L[n]} \quad (4.15)$$

$$F_{y1}[n] = F_1[n] \frac{y_1[n] - y_2[n]}{L[n]} \quad (4.16)$$

$$F_{z1}[n] = F_1[n] \frac{z_1[n] - z_2[n]}{L[n]} \quad (4.17)$$

4.4 Introduction de non linéarités

Pour le mapping audio, il peut être intéressant d'introduire des non-linéarités dans le calcul des forces. Le comportement de la masse est alors plus complexe et moins prévisible que dans le cas d'une liaison linéaire.

Après l'introduction de non-linéarités, le système ne peut plus être décrit comme un modèle masse-ressort-amortisseur. On lui préfère alors le terme de masse-interaction.

4.4.1 Limites du travail de la force de rappel

Les premières non-linéarités introduites sont sur la longueur de l'élongation. Le travail de la force de rappel est borné à deux valeurs limites L_{min} et L_{max} . En dehors de ces bornes le travail de la force est nul, la liaison admet alors pour équation :

$$F_1[n] = \begin{cases} -k(L[n] - L_0) - \mu(L[n] - L[n-1]) & \text{si } L_{min} < L[n] < L_{max} \\ -\mu(L[n] - L[n-1]) & \text{sinon} \end{cases} \quad (4.18)$$

On peut alors définir simplement une liaison linéaire par morceau (fig 4.2). Il suffit pour cela de découper une liaison en plusieurs avec la longueur minimale de l'une égale à la longueur maximale de l'autre. La rigidité sera toutefois toujours antisymétrique par rapport à L_0 .

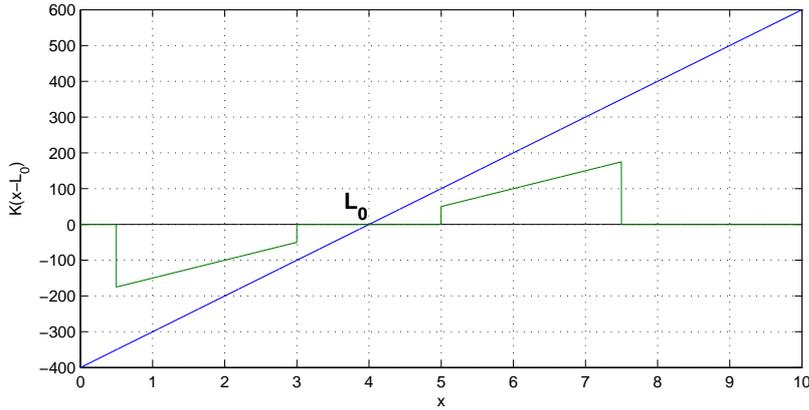


FIG. 4.2 – Limites de l'élongation : travail de la force de rappel illimité et avec bornes.

4.4.2 Puissance du travail de la force de rappel

Dans l'objectif de pouvoir implémenter une plus grande variété de situations physiques, un paramètre a été ajouté à la liaison, la puissance de l'élongation P . On rajoute cette puissance dans l'équation de calcul de la force qui devient :

$$F_1[n] = -k(L[n] - L_0)^P - \mu(L[n] - L[n-1]) \quad (4.19)$$

Cette puissance P peut être n'importe quel nombre réel. On peut par exemple définir des liaisons en $1/R^2$ avec $P = -2$ et $L_0 = 0$ qui modélisent les champs de potentiels (fig. 4.3, voir couverture) ou implémenter des équations plus complexes pour le calcul d'une rigidité, comme celle de Cooper & Maddock [CM97] qui ont défini une équation qui évite l'effondrement de structure masse-ressort-amortisseur :

$$F_{k1}[n] = -k \left(L[n] - \frac{L_0^2}{L[n]} \right) \quad (4.20)$$

4.4.3 Liaisons orientées

Le dernier type de non-linéarités implémenté est la liaison orientée. On définit deux types de liaisons orientées : les liaisons tangentielles et les liaisons normales. Les liaisons orientées fonctionnent comme des liaisons classiques excepté le fait que l'élongation est projetée sur un espace vectoriel. Cet espace est un vecteur dans le cas des liaisons tangentielles et un hyperplan pour les liaisons normales.

On obtient alors les équations suivantes en 3 dimensions :

Pour une liaison tangentielle au vecteur $X_a = (x_a \ y_a \ z_a)^t$

$$F_{x1}[n] = F_1[n] \frac{x_1[n] - x_2[n]}{L[n]} x_a \quad (4.21)$$

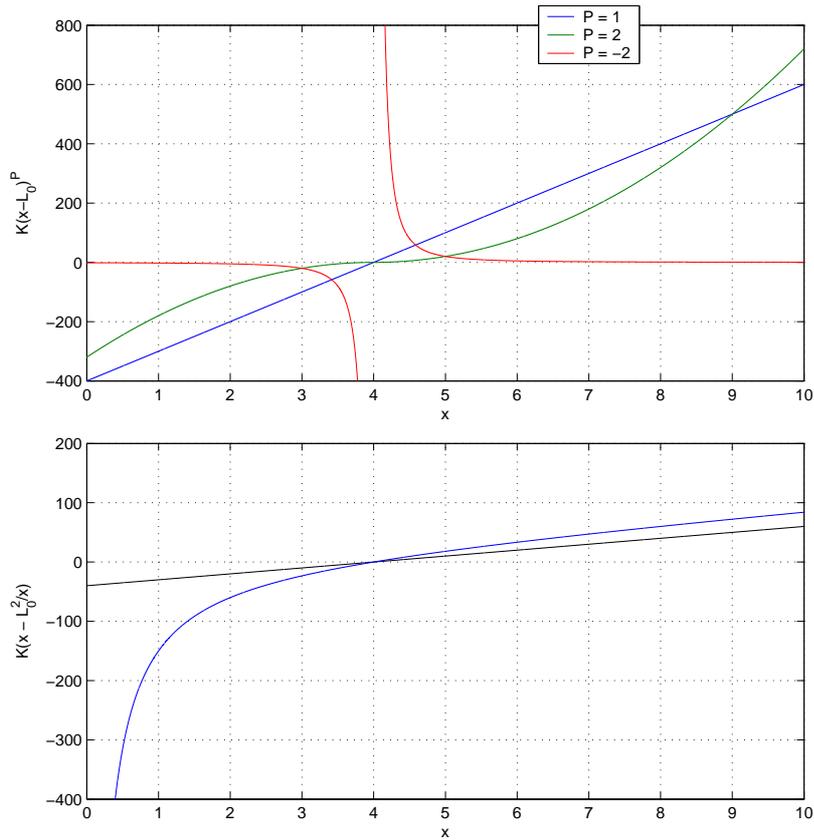


FIG. 4.3 – Puissance de l'élongation : *Haut* : travail de la force de rappel pour $P = 1, 2$ et -2 . *Bas* : Rigidité classique et implémentation de l'équation de Cooper [CM97] à l'aide de deux liaisons.

$$F_{y1}[n] = F_1[n] \frac{y_1[n] - y_2[n]}{L[n]} y_a \quad (4.22)$$

$$F_{z1}[n] = F_1[n] \frac{z_1[n] - z_2[n]}{L[n]} z_a \quad (4.23)$$

Pour une liaison normale au vecteur X_c tel que $\vec{X}_c \cdot \vec{X}_b = \vec{X}_c \cdot \vec{X}_a = \vec{X}_b \cdot \vec{X}_a = 0$

$$F_{x1}[n] = F_1[n] \frac{x_1[n] - x_2[n]}{L[n]} (x_a + x_b) \quad (4.24)$$

$$F_{y1}[n] = F_1[n] \frac{y_1[n] - y_2[n]}{L[n]} (y_a + y_b) \quad (4.25)$$

$$F_{z1}[n] = F_1[n] \frac{z_1[n] - z_2[n]}{L[n]} (z_a + z_b) \quad (4.26)$$

Ceci permet de définir des interactions spécifiques. Deux exemples simples :

- En 3D, si on réalise une liaison normale à l'axe z de longueur au repos constante L_0 entre toutes les masses et une masse située au centre de l'espace, on réalise un cylindre d'axe z et de rayon L_0 (fig. 4.4).

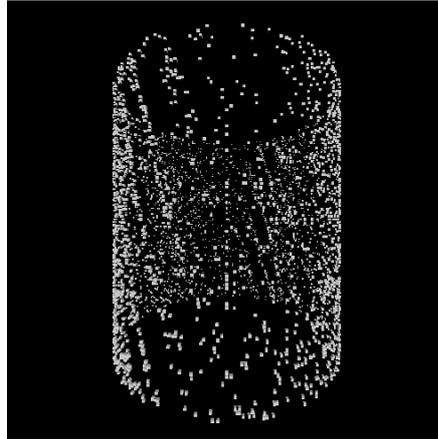


FIG. 4.4 – Liaison orientée par rapport au plan xy .

- En 2D, si on ajoute deux liaisons tangentielles aux axes x et y de raideur nulle $k = 0$ entre toutes les masses et une masse centrale, on réalise un amortissement général simulant le frottement ou la viscosité du milieu. Ceci évite d'avoir un mouvement de rotation infini, qui n'est jamais amorti si $L[n]$ est constant.

4.5 Utilisation de la librairie MSD

MSD, pour Mass-Spring-Damper, est la collection d'objets externes implémentée en C++. Elle peut être compilée pour Pure Data ou Max, sous les systèmes les plus courants (Windows, Linux et Mac Os X). Elle est constituée d'un

fichier en-tête msd.h qui est un modèle pour un objet à N dimensions, et d'instanciations de cet objet pour les différents espaces envisagés(1D, 2D, 3D ou ND).

L'utilisateur utilise donc un objet correspondant à l'espace choisi et portant sa terminologie, par exemple msd2D (fig. 4.5). La gestion du modèle se fait par l'envoi de messages à l'objet msd. Ces messages sont de 4 types :

- création : permettent de définir la structure en créant des liens et des masses
- bang : calcul du nouvel état du modèle à partir de l'instant précédent.
- attributs : permettent de récupérer les données internes du modèle.
- paramétrage dynamique : permettent de modifier les valeurs des paramètres internes du modèle.

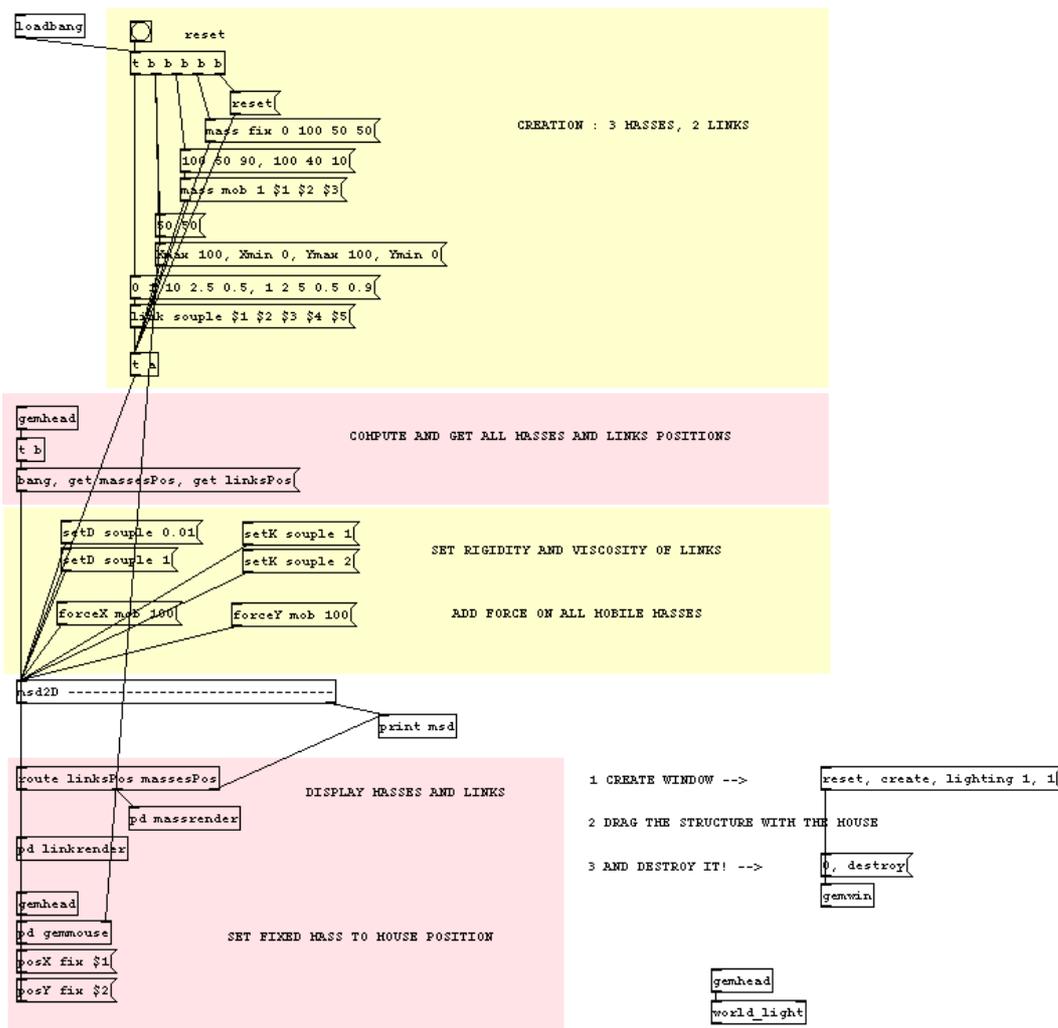


FIG. 4.5 – Exemple d'utilisation complète de l'objet msd2D dans un patch Pure Data : création, calcul, attributs, paramétrage.

Les possibilités de ces messages sont détaillées dans les paragraphes suivants.

La liste exhaustive des messages et de leur paramètres est présentée en annexe .1.

4.5.1 Messages de création

Les messages de création permettent de définir la structure du modèle physique. Ils servent donc à créer les masses et les liens utilisés.

Les *masses* créées peuvent être de deux types : mobiles ou immobiles. Les masses immobiles servent à attacher la structure à un point fixe de l'espace, les forces appliquées à ces masses ne sont pas calculées. L'utilisateur donne un nom, une masse et une position initiale à toutes les masses créées, tandis qu'un numéro unique leur est attribué.

Les *liaisons* sont de trois types à partir de la dimension 2 : non orientées, tangentielles ou normales. Les liens non orientés sont calculés selon les équations du paragraphe 4.3. Pour les liens orientés, les forces exercées par les liaisons sont projetées selon un vecteur pour les liens tangentiels, ou un hyperplan pour les liens normaux. Le vecteur permettant de définir ces espaces vectoriels est donné par l'utilisateur. Il définit en plus les propriétés physiques de la liaison : masses reliées, rigidité, amortissement, et si besoin puissance, longueurs minimales et maximales ainsi qu'un nom.

Des messages permettent en plus de détruire une masse ou une liaison afin de créer des structures dynamiques.

4.5.2 Message de calcul

Le message *bang* demande au modèle de calculer son nouvel état $n + 1$ à partir de l'état n . Le modèle fonctionnant ici à une fréquence de contrôle et non à une fréquence audio, on fera bien attention au fait que la fréquence d'échantillonnage du modèle correspond à la fréquence d'envoi du message *bang*.

4.5.3 Messages attributs

Les messages attributs permettent à l'utilisateur de récupérer des informations de l'état interne du modèle. Ils sont nombreux pour permettre une utilisation variée des objets *msd*. Il existe par exemple un message qui renvoie l'état de toutes les masses et de tous les liens du modèles. D'autres messages sont plus spécifiques et permettent de demander un paramètre particulier, par exemple la vitesse des masses nommées *Id1* ou alors les forces exercées sur les masses 20 à 50... Des descripteurs de plus haut niveau ont également été implémentés afin de caractériser l'ensemble de l'objet. On a par exemple accès à la moyenne des positions des masses ou à l'écart type des forces exercées dans la structure. Le modèle sort alors les paramètres demandés avec un identifiant permettant de router les paramètres vers des algorithmes

de synthèses qui servent ensuite à la création de vidéo et/ou d'audio (fig. 4.6).

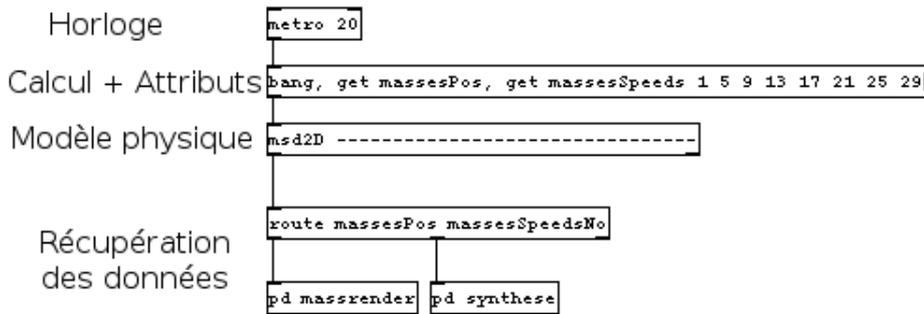


FIG. 4.6 – Exemple d'utilisation des messages *bang* et *attributs* suivis d'un routage vidéo et son dans Pure Data.

4.5.4 Messages de paramétrage

Les messages de paramétrages dynamiques permettent de redéfinir explicitement un paramètre du modèle. Là encore, ils sont nombreux. On peut citer en exemple la possibilité de changer la rigidité de tous les liens nommés Id , de redéfinir la longueur L_0 d'une liaison, d'envoyer une force en x et en y à la masse numéro 2, ou encore de déplacer une masse.

Ces messages sont importants car ils sont utilisés pour interagir avec le modèle. C'est grâce à eux que l'utilisateur va pouvoir "jouer" avec le modèle comme expliqué dans le chapitre 5.

4.6 Résultats

4.6.1 Oscillation du système masse - ressort - amortisseur

Le premier élément à valider est la réponse en fréquence du système masse-ressort amortisseur. Le test a été effectué directement dans Pure Data à l'aide de la fonction implémentée `[rfft]` qui calcule la transformée de Fourier pour les fréquences allant de 1 à $F_e/2$. Avec un dirac en force en entrée du système, on obtient en sortie la réponse présentée sur la figure 4.7. On voit sur la figure que les résultats sont proches. Les fréquences de résonance sont similaires en théorie et en pratique. Cependant il existe une différence d'amplification dans les hautes fréquences, celle-ci pouvant atteindre 15 dB dans le cas d'un amortissement élevé.

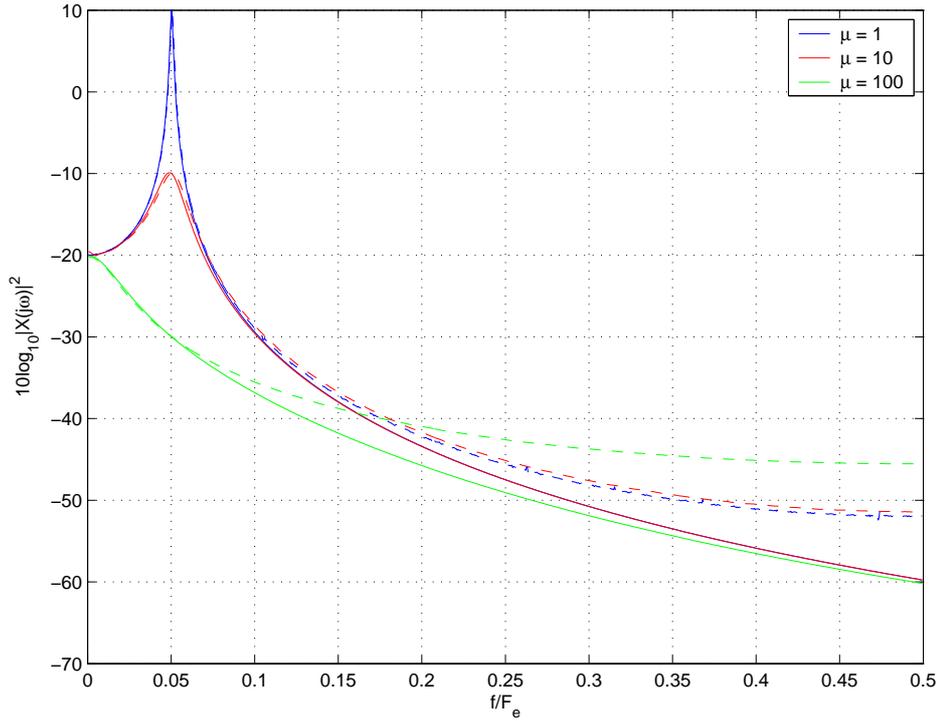


FIG. 4.7 – Réponse en fréquence théoriques (-) et pratiques (- -) pour $M = 100$ et $K = 10$ et trois valeurs de la constante d'amortissement μ .

4.6.2 Performances

L'implémentation actuelle permet le calcul d'environ 5000 masses et de 5000 liaisons à une fréquence de 50Hz sur un ordinateur individuel. Ces performances sont atteintes dans PD en représentant le modèle dans un espace à trois dimensions avec GEM à 50 Hz également. Ces résultats nous permettent d'implémenter des mappings sans se préoccuper des limitations de calcul puisque le nombre de masses utilisées pour les réaliser est rarement supérieur à une cinquantaine. la puissance de calcul peut donc être réservée pour les algorithmes de synthèse audio.

Chapitre 5

Les modèles physiques pour le contrôle des paramètres de synthèse

*”With the aid of electronic computers,
the composer becomes a sort of pilot :
he presses the buttons, introduces coordinates
and supervises the control of a cosmic vessel
sailing in the space of sound”.*
- Iannis Xenakis, 1963

5.1 Utilisation des modèles physiques pour le mapping

Les recherches menées sur le mapping entre les paramètres de contrôle et les paramètres de synthèse montrent qu’un mapping complexe est plus intéressant du point de vue de l’expressivité artistique ([RWDD97],[HWK00]). Hunt et. al écrivent :

*Les stratégies de mapping qui ne sont pas un-à-un peuvent être plus séduisantes aux utilisateurs que les mappings un-à-un*¹

5.1.1 Métaphores du réel

Le problème principal du mapping audio est qu’il n’a *a priori* aucun lien avec la réalité physique. Grâce à notre implémentation des modèles particuliers, nous pouvons à cette étape simuler un comportement ”naturel”. Alors que les modèles physiques sont souvent utilisés directement pour *la synthèse audio*, nous proposons de les utiliser pour *la création des paramètres*

¹”Mapping strategies which are not one-to-one can be more engaging to users than one-to-one mappings.”

de synthèse d'après le formalisme d'Henry [H04] (fig. 5.1). Ce paradigme présente plusieurs avantages :

- les modèles physiques permettent un contrôle naturel et expressif. En effet, ceux-ci permettent de s'imaginer l'espace dans lequel ils évoluent si celui-ci est en 1,2 ou 3D. Par exemple, lorsque l'utilisateur lance une balle 2D, il se représente parfaitement le moment où elle va atteindre un bord. On peut également facilement représenter le modèle en associant une sphère à une masse et une ligne à une liaison, ce qui permet à l'instrumentiste d'avoir un retour visuel sur ses actions. La représentation peut être augmentée en jouant sur les paramètres de représentation (taille, couleur) selon les désirs de l'utilisateur.
- les modèles particuliers sont adaptés pour recréer des structures issues de la réalité, aussi bien que des structures inventées. Les objets issus de la réalité sont intéressants à manipuler, car n'importe quel utilisateur saura anticiper, en se basant sur son expérience, la réaction et l'évolution de la structure. De plus, on peut étendre ce principe à des structures irréelles mais composées d'éléments de nature physique. On peut ainsi envisager la description du comportement d'un violon, aussi bien qu'une corde étonnamment souple.
- le modèle évolue dans le temps. Le mapping est donc dynamique et dépend de l'instant. Ce phénomène est important pour réussir à imiter le comportement des instruments acoustiques. En effet, le son produit par un instrumentiste ne dépend pas uniquement du jeu de paramètres d'entrée mais également de l'état interne du système. Un flûtiste, par exemple, réalisera une note ou son octave avec les mêmes paramètres de contrôles selon ce qui a été joué précédemment. Grâce à un mapping dynamique, on peut reproduire ce type de situation.
- la fréquence de travail d'un signal de contrôle étant peu élevée, le modèle implémenté peut être très complexe et cependant calculé en temps réel. Une fréquence de 50 Hz suffit pour un retour visuel fluide, tandis que les fréquences couramment utilisées pour les données de contrôle sont du même ordre de grandeur (100 Hz).
- on peut intégrer une interface à retour de force au modèle. Ces interfaces prennent en entrée une force appliquée par l'utilisateur et envoi au système une position. Cela correspond à l'élément masse de notre modèle qui pourra donc être associé à ce type d'interface.

Comme présenté sur la figure 5.1, l'utilisation du modèle physique pour le mapping peut se coupler de manière très intéressante au mapping vidéo. Alors que la plupart des systèmes de génération sonore et vidéo en temps-réel s'appuie soit sur la vidéo pour créer le son, soit sur le son pour créer la vidéo, il est possible d'utiliser la même source de donnée, l'interface et de mapper le même modèle physique pour la synthèse sonore et pour la synthèse vidéo. Ceci permet d'obtenir une cohésion forte entre l'image et le son, tout en jouant sur les décalages entre les paramètres fournis à l'algorithme vidéo et ceux fournis à l'algorithme audio. Les deux synthèses font alors parties du

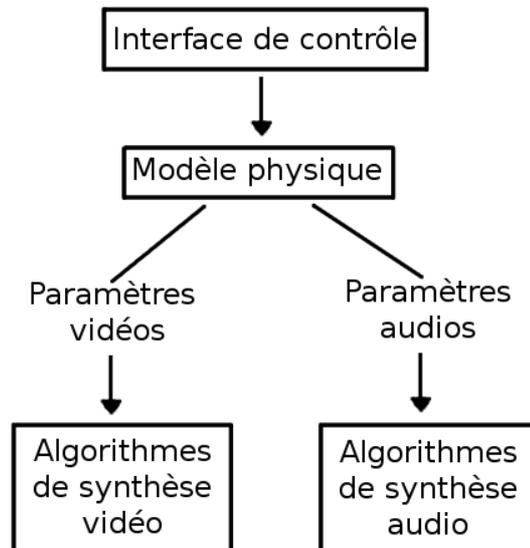


FIG. 5.1 – Utilisation des modèles physiques pour le mapping [MH05].

même instrument.

5.1.2 Exemple de la synthèse additive

Nous présentons ici un exemple de mapping divergent utilisant les modèles physiques particulières.

L’algorithme de synthèse utilisé est celui de synthèse additive. Le son est créé en ajoutant des sinus dont l’amplitude évolue au cours du temps (métaphore immersive).

Le modèle utilisé est constitué de 30 masses initialement positionnées aléatoirement. Chaque masse est reliée à la masse créée juste avant et juste après elle.

L’interface de contrôle de l’utilisateur est la souris et l’écran représentant la position actuelle des masses et leurs liens (fig 5.3).

Le modèle sort en permanence les positions de toutes les masses et de tous les liens afin d’actualiser les données à l’écran. Il sort également les vitesses en x et en y de 8 masses choisies arbitrairement. Les vitesses de ces masses contrôlent directement l’amplitude de 16 générateurs sinusoïdaux de fréquences différentes, chaque masse contrôlant 2 générateurs.

L’utilisateur peut ”attraper” à l’aide de la souris n’importe laquelle des masses et lui imprimer un mouvement qui va se transmettre par les liaisons à toutes les autres masses. Pour donner plus de choix sur le son généré, les masses les plus rouges sont associées aux générateurs les plus graves, tandis que les masses les plus blanches sont associées aux générateurs les plus aigus.

L’analyse d’un mouvement est présentée à la figure 5.4. A l’instant $t = 1s$, l’utilisateur attrape la masse 25 et la lance. On voit que l’énergie initialement

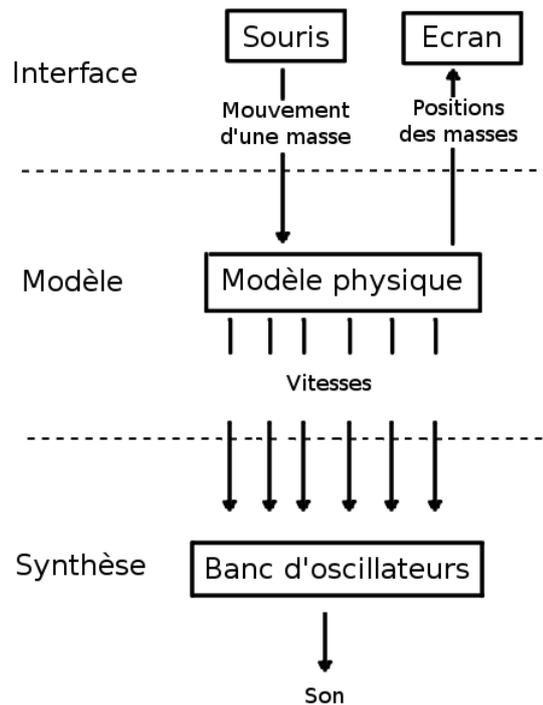


FIG. 5.2 – Synthèse additive : utilisation du modèle physique pour le mapping

transmise à la masse déplacée se propage par les liens aux masses les plus proches, puis aux masses plus éloignées. A l'instant $t = 5s$, la masse a cognée une des limites en x définie par l'utilisateur, la vitesse en x s'annule temporairement, et le son produit par le générateur associé à cette vitesse également.

5.1.3 Exemple de la synthèse granulaire

Un autre exemple de synthèse audio est proposé. A la différence du précédent, cet exemple s'appuie sur une utilisation de paramètres de contrôle macroscopique.

L'objectif de ce patch de synthèse granulaire est la création et le contrôle de textures sonores. L'utilisateur peut ici aussi visualiser l'état du modèle. Celui-ci est composé de 25 masses 2D disposées sous la forme d'un filet. Les masses rouges représentent les masses qui sont utilisées pour la synthèse sonore.

L'instrumentiste peut influencer sur le modèle en modifiant les paramètres physiques des liaisons.

Le modèle est représenté en temps-réel pour aider l'instrumentiste à jouer (fig. 5.5). Les 8 masses rouges sont des têtes de lecture pour la synthèse granulaire. Leur position en x représente la position du grain dans le fichier

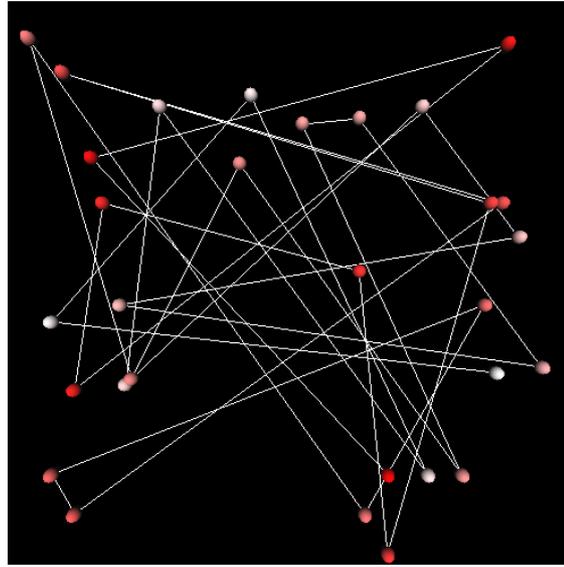


FIG. 5.3 – Synthèse additive : Interface d'utilisation.

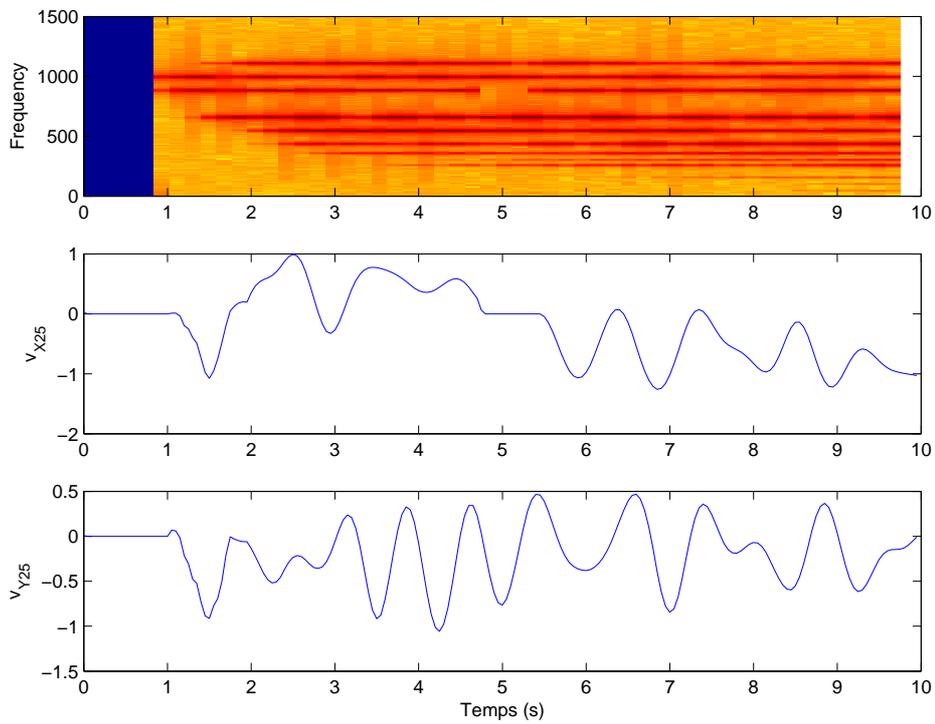


FIG. 5.4 – Synthèse additive : *Haut* : Spectrogramme du son généré. *Bas* : Vitesse en x et en y de la masse 25

audio, tandis que leur position en y est associée au volume du grain.

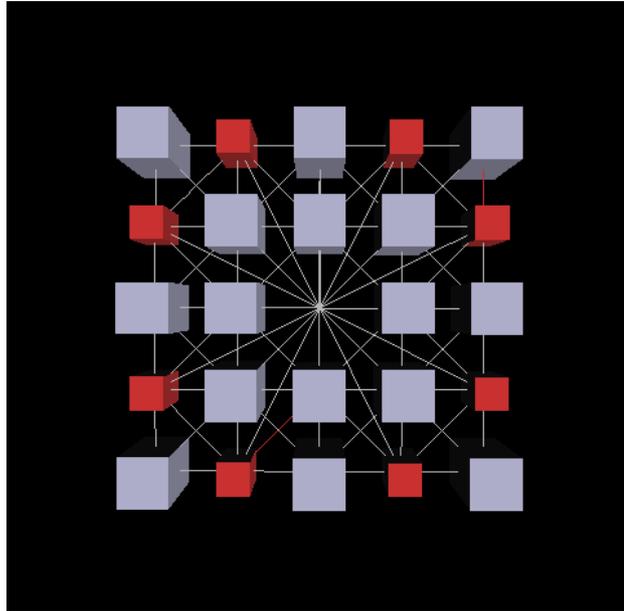


FIG. 5.5 – Synthèse granulaire : Représentation du modèle

Une analyse temps-fréquence d'un exemple d'utilisation de l'instrument est présentée sur la figure 5.6. L'interprète agit sur les paramètres physiques de la structure. A l'instant t_1 , il augmente la valeur de la longueur au repos L_0 de tous les liens. A l'instant, t_2 , il augmente la rigidité des liens tandis qu'à l'instant t_3 , il diminue la valeur L_0 . Le spectre de la texture est modifié sans pour autant changer la hauteur fondamentale de la nappe. Le contrôle du timbre est dissocié du contrôle de la mélodie.

5.2 Descripteurs de haut niveau

On peut également utiliser des paramètres macroscopiques pour décrire la structure. Au lieu de récupérer les forces et les positions individuelles des éléments, on a aussi accès à des valeurs statistiques pour l'ensemble de l'objet. On réduit ainsi le nombre d'informations sorties par le modèle.

Les fonctions implémentées permettent le calcul de la moyenne et de l'écart-type des paramètres de l'objet. On peut ainsi caractériser la déformation de l'objet avec la valeur de l'écart-type des positions des masses ou déterminer l'état de stabilité de l'objet avec la valeur moyenne des forces.

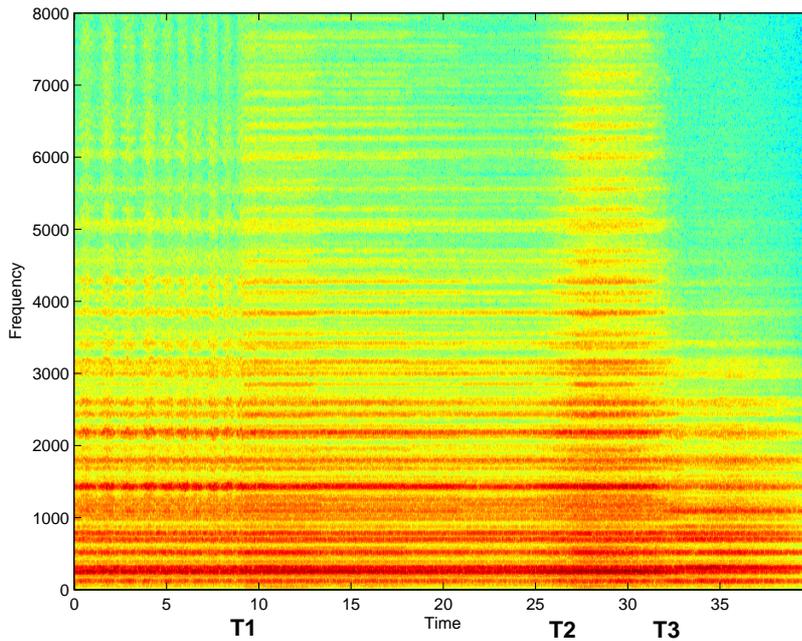


FIG. 5.6 – Synthèse granulaire : Spectrogramme du son généré

5.3 Modèles physiques à N dimensions

5.3.1 Proposition : Utilisation des modèles physiques à N dimensions

Comme on l’a vu dans le paragraphe 3.3, les mappings complexes sont plus intéressants du point de vue de l’utilisateur. Le modèle physique permet au contraire de réduire la charge cognitive de l’utilisation d’un algorithme de synthèse en temps-réel. Ce mapping n’a aucune raison d’être contraint à 3 degrés de liberté. On se propose donc de généraliser leur utilisation à N dimensions.

Ce formalisme nous paraît particulièrement intéressant dans le cadre du contrôle audio. Castagne & Cadoz ont défini 10 critères qui permettent d’évaluer la pertinence des modèles physiques pour la création musicale [CC03]. Parmi eux :

PM7 : A quel degré le modèle mental associé est-il intuitif et efficace ?²

Par modèle mental, ils entendent la représentation que construit l’utilisateur dans son esprit en interagissant avec le système. Un bon modèle mental doit permettre à l’utilisateur d’anticiper le résultat de ses actes et faciliter l’expérimentation. Castagne & Cadoz propose donc de mesurer ce critère par rapport à l’impression de réalité que l’utilisateur ressent en utilisant ce système.

²PM7 : How Intuitive and Effective is the Associated Mental Model ?”

Le modèle à N dimensions nous semble un bon compromis entre la persistance de l'image mentale et la complexité du mapping. La N -ième dimension est difficile à se représenter mentalement. Cependant, les rapports entre la N ème et la 1ère dimension sont régis par les mêmes lois qu'entre la deuxième et la première.

Ce type de relation augmente la dynamique temporelle du mapping. Les relations qui existent entre les différentes dimensions dépendent de l'état du modèle à l'instant précédent. En effet pour un système linéaire, une dimension ne sera jamais excitée par les autres si sa position initiale est nulle. Lorsque l'instrumentiste excite une dimension les non-linéarités du système engendrent un mouvement de toutes les masses précédemment excitées (qui ont été déplacées par rapport à leur point d'équilibre).

Ce modèle est particulièrement adapté pour une gestion des mouvements plutôt que des positions. On assimile l'application d'une force dans une dimension à l'ajout d'énergie au système.

5.3.2 Exemple de gamme évolutive

Pour permettre de comprendre la spécificité des liens qui existent entre les dimensions, on réalise ici un modèle constitué d'une masse à 13 dimensions reliée à un point centré dans l'espace par une liaison masse-ressort-amortisseur.

Cet exemple a pour objectif une gamme qui évolue dans le temps. Treize liaisons définissant des amortissements spécifiques à chaque dimension sont ajoutées afin d'avoir un temps d'amortissement différent pour chaque dimension.

Chaque dimension est associée à une hauteur de la gamme tempérée relative à une hauteur fondamentale ou tonique, définie par l'utilisateur.

L'utilisateur peut appliquer une force sur chacune des dimensions, la hauteur correspondant à cette dimension va être excitée mais également les hauteurs précédemment excitées qui ne sont pas retournées à leur état de repos. Après excitation, le son s'éteint rapidement. Une analyse temps-fréquences d'une utilisation du patch est présentée sur la figure 5.7 et est comparée à une synthèse additive associant un seul sinus par fréquence. On lit le même fichier MIDI dans les deux patchs. Sur le spectrogramme du son de la gamme évolutive, on voit que les fréquences générées dépendent de celles qui ont été précédemment excitées. Plus une dimension a reçu de l'énergie et plus elle est éloignée de son état initial, l'excitation des autres dimensions aura pour effet de créer un mouvement ramenant les autres dimensions à leur état de repos.

Cet exemple nous montre qu'il est facile de réaliser des systèmes à mémoire avec les modèles physiques. Grâce à cette gamme, on conserve la prépondérance de la hauteur originale sur les autres, ce qui permet de reconnaître aisément la mélodie. Cependant, on enrichit le spectre, il devient donc intéressant d'explorer l'instrument.

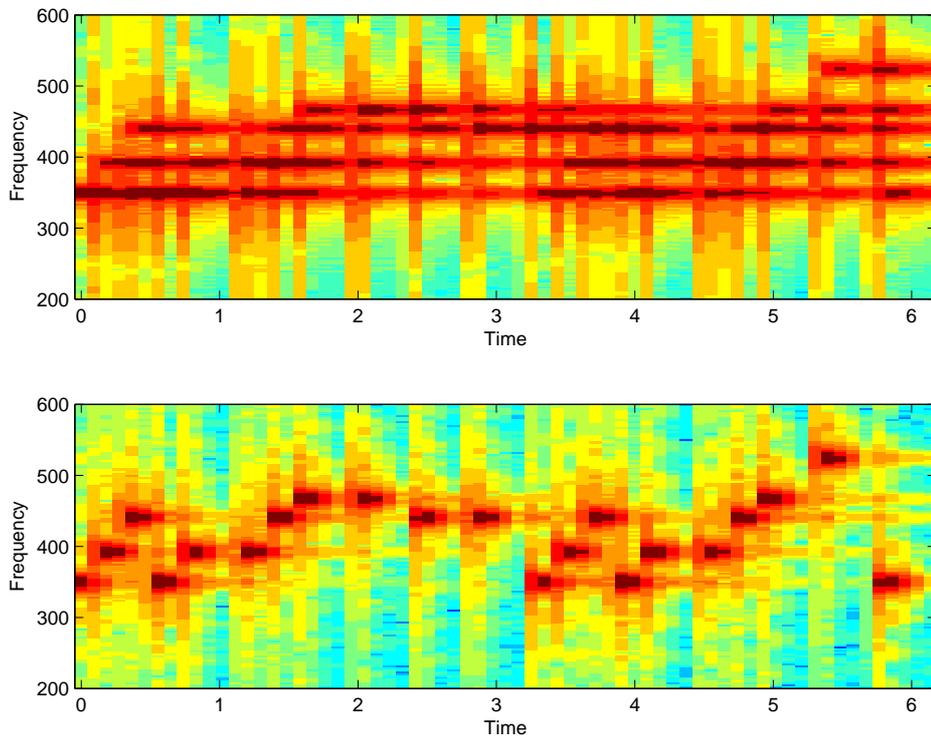


FIG. 5.7 – Modèle 13D : *Haut* : Spectrogramme du son généré par la gamme évolutive. *Bas* : Spectrogramme du son généré par son pur.

Chapitre 6

Conclusion

6.1 Bilan du travail réalisé

Dans ce document, nous reprenons le formalisme défini par Cadoz. Les modèles particuliers permettent une grande modularité et la simulation de nombreux phénomènes. Nous avons implémenté une collection d'objet pour les environnements PD et Max qui permettent de réaliser ces calculs en temps-réel. Ces logiciels permettaient d'avoir un support pour l'interface des objets MSD : leur utilisation, le rendu graphique, le lien avec l'audio. Ces objets ont été développés en C++ sur le système FlexT de Thomas Grill qui permet de compiler les sources à la fois pour Pure Data et pour Max/MSP.

Nous avons utilisé les modèles physiques afin de relier paramètres de contrôle et paramètres de synthèse. Ce formalisme présente des avantages sur plusieurs aspects du mapping. Les modèles définissent un rapport complexe, cette complexité étant indispensable pour un contrôle expressif. Ils permettent de modéliser une réalité au sein de l'ordinateur. L'expérience que possède tout un chacun sur les lois physiques environnantes assurant une bonne compréhension de l'évolution du modèle, ils facilitent la représentation mentale que se fait l'instrumentiste de l'espace de paramètres. Ils réalisent un mapping dynamique qui permet d'adopter un comportement plus proche des instruments acoustiques traditionnels. Enfin, ils permettent également de synthétiser audio et vidéo à partir des mêmes données, créant une cohésion entre les deux médias.

Nous avons généralisé les modèles physiques à la dimension N pour cette utilisation. Les rapports qui existent entre les paramètres de synthèse sont alors régis non plus par les liaisons entre les masses mais principalement par les interactions entre les dimensions et les non-linéarités qui transfèrent l'énergie d'une dimension à une autre.

6.2 Perspectives

Les perspectives de ce travail sont nombreuses et variées. Sur un aspect physique, on peut envisager de nouvelles implémentations des équations de liaisons et de masses. Pour la liaison, l'objectif serait de pouvoir définir la caractéristique en rigidité du ressort non plus avec des paramètres (K,P,L...) mais avec une véritable caractéristique que l'objet msd irait lire dans un fichier texte. Par ce moyen, le système pourrait facilement imiter des caractéristiques complexes comme le chevalet d'un violon ou la table d'harmonie d'un piano, si on prend des exemples dans l'univers des instruments de musique.

Il faut aussi étudier l'implémentation de liaisons ressort et amortisseur en série. Ce type de liaison permettraient de décrire des phénomènes physiques différents, comme des liaisons avec seuils.

D'un point de vue interface, on peut intégrer l'utilisation d'un joystick à retour de force assez simplement. Il suffit d'associer le joystick à une masse, ses mouvements seront alors considérés comme des forces appliquées à cette masse, tandis que les forces appliquées sur cette masse seront envoyées au servo-moteurs du joystick.

Pour poursuivre ce travail, il faudrait arriver à caractériser des classes de structure qui permettent un mapping intéressant pour la synthèse : comment réussir à faire une structure qui permettent de contrôler une synthèse additive qui produise un son de clarinette. Si on arrive à catégoriser les structures, on peut envisager d'étudier la construction du modèle physique en fonction des contraintes liées à la synthèse et à l'interface. L'objectif final étant un schéma de mapping universel.

Pour les applications futures, on peut envisager d'implémenter un patch de "vol d'oiseaux" qui posséderait les propriétés typiques de ces systèmes. Les oiseaux seraient représentés par des masses orientées. On pourrait ainsi à chaque instant repérer les 3 masses les plus proches dans le champ de vision de la masse afin de créer des liaisons correspondantes et de détruire celle de l'instant précédent. On peut également envisager un choix entre différents comportements comme décrits dans les études sur les vols d'oiseaux ou les bancs de poissons.

Des systèmes externes de générations de maillage peuvent également être utilisés dans la librairie implémentée. Frank Barknecht a utilisé le système Triangle afin de générer des maillages triangulaires de Delaunay.

Chapitre 7

Annexes

.1 Fichiers d'aide de l'objet msd2D pour Pure Data

```

HELP: msd2D
DESCRIPTION: Mass spring damper physical modeling in 2D.

Full Name: msd2D

Examples:
Sorry, you need GEN for this example...

Reset
pd init
ON/OFF
pd compute

1 Create window --> reset, create, 1(
2 Drag the structure with the mouse
3 Destroy the window --> destroy(
gemwin

r $0-in
msd2D
s $0-out

r $0-out
route massesPos linksPos
pd gemrender

genhead
pd gemmouse
s $0-in

Inlets:
- Left: Bang - A bang at the left inlet compute the new model state based on previous instant.
Messages - Different messages are used to control the msd object. They are of three types :
pd creation To create the model masses and links.
pd dynamic settings To set the model parameters after creation.
pd attributes To get the model parameters

Arguments:
None

Outlets:
- Left: Outputs the model parameters asked with the attributes messages.
- Right: Outputs information on model when creation messages are send or with the special message [info$() which dump the complete state of the model.

See Also: 01_msd2Dtest.pd editor/msd2d-editor.pd
CATEGORY: control
KEYWORDS: physical model mass spring damper link
pd More_Info pd Related_Objects - Nicolas Montgermont, May 12, 2005

```

FIG. 1 – Fichier d'aide principal de l'objet msd2D.

CREATION Messages

Creation messages are used to define the structure of the model. Messages create links and masses or destroy them.

Reset :

`reset` Delete all masses, links and internal variables

Masses :

`mass $1 $2 $3 $4 $5` Add a mass
 \$1 : Id (symbol)
 \$2 : fixed or mobile (0/1)
 \$3 : mass
 \$4, \$5 : initial position

`deleteMass $1` Delete a mass and associated links
 \$1 : Creation No of mass

Links :

`link $1 $2 $3 $4 $5 ($6 $7 $8)` Add link(s)
 \$1 : Id (symbol)
 \$2 : creation No/Id of mass1
 \$3 : creation No/Id of mass2
 \$4 : rigidity
 \$5 : damping
 (\$6) : Power of the rigidity distance
 (\$7) : minimum lenght of link
 (\$8) : maximum lenght of link

`deleteLink $1` Delete a link
 \$1 : Creation No of link

Oriented links :

In 2D (and 3D) there are two specials links : oriented links. They works as general links excepts their calculation is made following a vector.

`tLink $1 $2 $3 $4 $5 $6 $7 ($8 $9 $10)` Add tangential link(s)
 \$1 : Id (symbol)
 \$2 : creation No/Id of mass1
 \$3 : creation No/Id of mass2
 \$4 : rigidity
 \$5 : damping
 \$6, \$7 : tangential vector
 (\$8) : Power of the rigidity distance
 (\$9) : minimum lenght of link
 (\$10) : maximum lenght of link

`nLink $1 $2 $3 $4 $5 $6 $7 ($8 $9 $10)` Add normal link(s)
 \$1 : Id (symbol)
 \$2 : creation No/Id of mass1
 \$3 : creation No/Id of mass2
 \$4 : rigidity
 \$5 : damping
 \$6, \$7 : normal vector
 (\$8) : Power of the rigidity distance
 (\$9) : minimum lenght of link
 (\$10) : maximum lenght of link

FIG. 2 – Fichier d'aide des messages de création.

DYNAMIC SETTINGS Messages

Dynamic settings messages allows the user to redefine internal parameters of links and masses.

Masses :

`posX $1 $2` Set position of fixed mass(es)
 \$1 : Id (symbol) or No
 \$2 : Value

`posY $1 $2`

`forceX $1 $2` Add force on mass(es)
 \$1 : Id (symbol) or No
 \$2 : Value

`forceY $1 $2`

`xmin $1 xmax $1` Set minimum and maximum position of all masses
 \$1 : Value

`ymin $1 ymax $1`

`setMobile $1` Set mass to mobile or fixed
 \$1 : Id (symbol) or No

`setFixed $1`

`grabMass $1 $2 $3` Grab nearest mass
 \$1, \$2 : position
 \$3 : grab or not (0/1)

Links :

`setK $1 $2` Set rigidity of link(s)
 \$1 : Id (symbol) or No
 \$2 : New value

`setD $1 $2` Set damping of link(s)
 \$1 : Id (symbol) or No
 \$2 : New value

`setL $1 $2` Set initial lenght of link(s)
 \$1 : Id (symbol) or No
 \$2 : New value

FIG. 3 – Fichier d'aide des messages de paramétrages.

ATTRIBUTES Messages

The attributes messages ask the object to output some of his internal parameters. They can be output by lists for positions and forces of masses.

General :

`infosl()` Get infos on all masses and links on right outlet

Lists :

`massesPosl()` Output all masses positions in a list on outlet No 1
`massesPosXl()` Output all masses x or y in a list on outlet No 1
`massesPosYl()`
`massesForcesl()` Output all forces applied on masses in a list on outlet No 1

Specific :

`get $1 ($2)` Get specific attribute on specific element

`$1` : Attribute type (massesPos / massesSpeeds / massesForces / linksPos / linksLengths)
`($2)` : - If not defined all the attributes are send for all the elements. - Ids or/and creations No

The get message return the asked attribute preceded by an identifier and the creation No of the element. The identifier is made of the asked parameter and the way you asked for it.

Examples with 3 masses numbered 0, 1 and 2 and named mas:

```

message                                response
[get massesPos( -----> [massesPos 0 x0(
                                [massesPos 2 x2(
                                [massesPos 1 x1(
[get massesPos 1( -----> [massesPosNo 1 x1(
[get massesPos mas( -----> [massesPosId 0 x0(
                                [massesPosId 2 x2(
                                [massesPosId 1 x1(
    
```

FIG. 4 – Fichier d'aide des messages attributs

MSD : mass - spring - damper model

MSD is the 2D object of the msd objects collection.
 It is designed to implement particules physical model in PD. The model is based on two elements type : mass and link. The msd masses are the principals objects of the model. They got only one physical parameter, the value of their mass. They can be mobile or fixed, in this case forces applied on them automatically, by links, or manually, by messages, don't do anything.
 Be careful : if masses are deleted, lists messages won't work
 Links connect masses two by two. They got 4 physicals parameters : length, rigidity, damping and power.
 Rigidity, damping and power are defined by the creation message. The lenght is initialised to the distance between the two masses at the creation.
 Links can be created in one shot between mutiples masses, instead of creation number, the masses linked are defined with their Id.

The equations are :

$$\text{if } L_{min} < |L[n] - L[0]| < L_{max}$$

$$F[n] = K(L[n] - L[0])^P + D(L[n] - L[n-1])$$

else

$$F[n] = D(L[n] - L[n-1])$$

For oriented links, the force $F[n]$ is projected onto a vector which is given during the creation of the link using $xlyl$ coordinates.
 For normal vector, the direction is calculated using the scalar product :

$$x_1x_2 + y_1y_2 = 0$$

You can build specific links using different links messages defining the characteristic step by step.

`pd figure`

FIG. 5 – Fichier d'aide d'informations générales.

.2 Téléchargement de la librairie MSD

La librairie MSD est disponible sur le CVS de Pure Data à l'adresse :

<http://cvs.sourceforge.net/viewcvs.py/pure-data/externals/nusmuk/>

Les fichiers sont répartis dans 3 répertoires `msd`, `msd2D` et `msd3D` correspondant aux objets 1, 2 et 3D. Un répertoire `/editor` a été ajouté par Frank Barknecht dans lequel un éditeur graphique pour `msd` est disponible.

Le téléchargement se fait à l'aide d'un `cvs`, les instructions de téléchargement sont disponibles à l'adresse :

https://www.puredata.org/docs/developer/sourceforge_cvs

Il est nécessaire d'installer FlexT pour compiler la librairie. FlexT est également disponible sur le CVS de PD à l'adresse :

<http://cvs.sourceforge.net/viewcvs.py/pure-data/externals/grill/flex/>

La librairie MSD est disponible sous licence LGPL : *Lesser General Public License*, ce qui signifie approximativement qu'elle est libre mais peut être incluse dans un logiciel propriétaire sous forme de librairie externe.

Bibliographie

- [C79] C. Cadoz. Synthèse Sonore par Simulation de Mécanismes vibratoires, Application aux Sons Musicaux. *Thèse à l'Institut National Polytechnique de Grenoble*, 1979.
- [C98] M. Chion. Le Son. Paris, *Editions Nathan*, 1998.
- [C02] J. Chadabe. The Limitations of Mapping as a Structural Descriptive in Electronic Instruments. In *Proceedings of the Conference on New Instruments for Musical Expression*, 2002.
- [CC02] N. Castagne & C. Cadoz. L'Environnement GENESIS : Créer avec des Modèles Physiques Masse-Interaction. In *Journées d'Informatique Musicale*, 2002.
- [CC03] N. Castagne & C. Cadoz. 10 Criteria for Evaluating Physical Modelling Schemes for Music Creation. In *Proceedings of the 6th International Conference on Digital Audio Effects*, 2003.
- [CLF93] C. Cadoz, A. Luciani & J.L. Florens. CORDIS-ANIMA : A Modelling System for Sound and Image Synthesis, the General Formalism. In *Computer Music Journal* **17**,(1), 19-29, 1993.
- [CM97] L. Cooper & S. Maddock. Preventing Collapse within Mass-Spring-Damper models of Deformable Objects. In *Proceedings of the Fifth International Conference in Central Europe on Computer and Graphics Visualization*, 1997.
- [G04] T. Grill. FLEXT, A C++ Programming Layer for Cross-Platform developments of PD and Max/MSP externals. 2004.
- [H04] C. Henry. Physical Modeling for Pure Data (pmpd) and Interaction with a Real-Time Audio Synthesis. In *Proceedings of the Sound and Music Computing*, 2004.
- [HR03] D. M. Howard & S. Rimell. Gesture-tactile Controlled Physical Modelling Music Synthesis. In *Proceedings of the Stockholm Music Acoustics Conference*, 2003.
- [HWP02] A. Hunt, M.M. Wanderley & M. Paradis. The importance of parameter mapping in electronic instrument design. In *Proceedings of the Conference on New Instruments for Musical Expression*, 2002.
- [HWK00] A. Hunt, M.M. Wanderley & R. Kirk. Towards a Model for Instrumental Mapping in Expert Musical Interaction. In *International Computer Music Conference*, 2000.

- [MA93] J.D Morrison & J.M. Adrien. MOSAIC : A Framework for Modal Synthesis. In *Computer Music Journal* **17**,(1), 45-56, 1993.
- [MFM97] A. G. E. Mulder, S.S. Fels & K. Mase. Empty-handed Gesture Analysis in Max/FTS. In *Proceedings of the Kansei Workshop*, 1997.
- [MH05] A. Momeni & C. Henry. Dynamic Autonomous Mapping Layers for Concurrent Control of Audio and Video Synthesis. Submitted to *Computer Music Journal* , 2005.
- [MW03] A. Momeni & D. Wessel. Characterizing and Controlling Musical Material Intuitively with Geometric Models. In *Proceedings of the Conference on New Instruments for Musical Expression*, 2003.
- [P96] M. Puckette. Pure Data. In *Proceedings of the International Computer Music Conference* , 1996.
- [PO03] J.A. Paradiso & S. O'Modhrain. Current Trends in Electronic Music Interface. In *Journal for New Music Research* **32**,(2), 2003.
- [PZ90] M. Puckette & D. Zicarelli. Max : An Interactive Graphic Programming Environment. 1990.
- [RWDD97] J. Rován, M.M. Wanderley, S. Dubnov & P. Depalle. Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance. In *Proceedings of the Kansei Workshop*, 1997.
- [S04] C. Seleborg. Interaction Temps-Réel/Temps Différé : Elaboration d'un Modèle Formel de Max et Implémentation d'une Bibliothèque OSC pour OpenMusic *Mémoire de DEA ATIAM*, 2004.
- [W79] D. Wessel. Timbre Space as a Musical Control Structure. In *Computer Music Journal* **3**,(2), 1979.
- [WSR98] M.M. Wanderley, N. Schnell & J. Rován. ESCHER : Modeling and Performing Composed Instruments in real-time. In *Proceedings of the IEEE Symposium on System, Man and Cybernetics*, 1998.
- [WW02] D. Wessel & M. Wright. Problems and Prospects for Intimate Musical Control of Computers. In *Computer Music Journal* **26**,(3), 11-22, 2002.