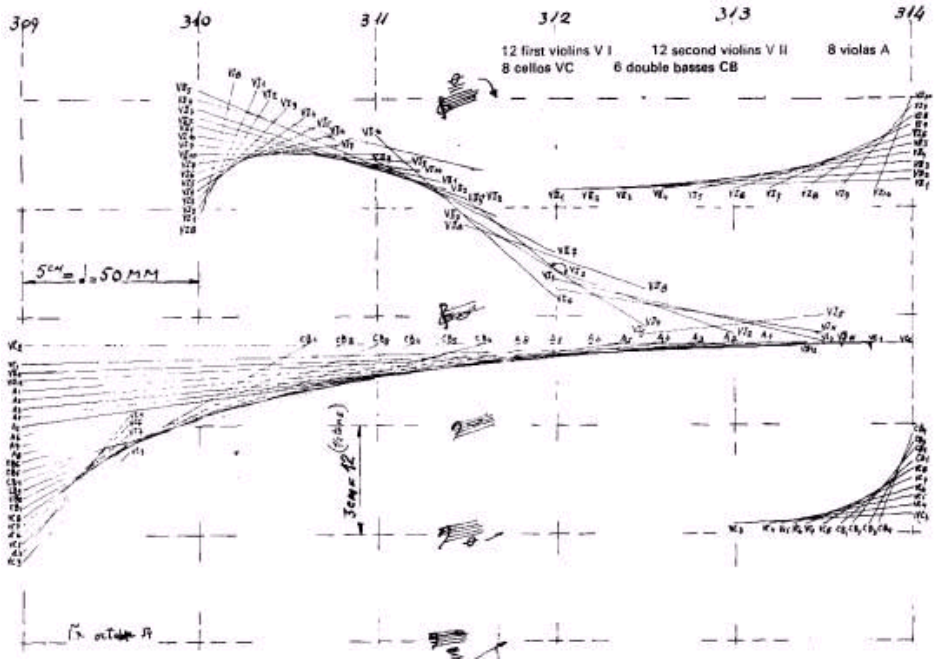
Contrôle de la synthèse granulaire



J.F. OLIVER

DEA ATIAM
(PARIS VI, ENST, AIX-MARSEILLE II, UJF GRENOBLE I)

Rapport du stage effectué au Centre de Création National G.M.E.M. (Groupe de Musique Electroacoustique de Marseille) sous la direction de L. POTTIER

TABLE DES MATIERES

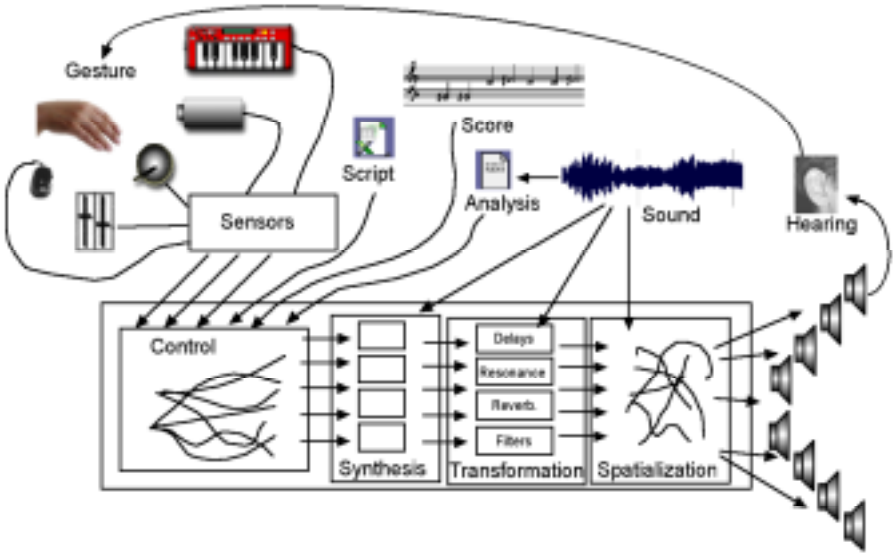
I Introduction

I.1 Le projet GMU (GMEM Microsound Universe)

Ce projet démarre cette année, c'est un nouveau programme de recherche à long terme lancé par le GMEM. Il vise à mettre au point un système intégré permettant la production et le contrôle de sons de synthèse pour des applications musicales.

Plusieurs points caractérisent ce système :

- Les techniques de synthèse choisies se démarquent des techniques de synthèse classiques généralement basées sur des oscillateurs continus. On privilégie ici le grain sonore et non le partiel sonore (voir annexes). Ces techniques sont destinées à la production de sons qui ne sont pas nécessairement harmoniques et doivent permettre d'explorer un domaine beaucoup plus large, plus proche des domaines de la musique concrète à laquelle la sensibilité du GMEM est rattachée.
- Le système est prévu pour intégrer aussi des techniques de contrôle appropriées, des outils de transformation du timbre et de spatialisation. Les variations des paramètres de synthèse peuvent induire des modifications corrélées des paramètres de traitement et de spatialisation.
- Les paramètres de contrôle de bas niveau seront regroupés et contrôlés par des paramètres perceptifs et continus (interpolations), permettant aisément aux compositeurs de créer de nouveaux types de sons.
- Le système sera opérationnel en temps-réel (environnement MAX/MSP)
- Des techniques d'analyse associées permettront de configurer automatiquement les différents modules du système (synthétiseurs, résonateurs).



Ma participation à ce projet est l'étude du contrôle d'un objet MAX/MSP implémenté en C++ qui est un générateur de grains sonores à partir de fichiers audios.
Ma tâche consiste à trouver des moyens de contrôler de manière ergonomique l'objet en temps-réel (utilisation de MAX/MSP) et à partir de générations automatiques de valeurs pour les paramètres. Pour ce deuxième point, j'ai travaillé dans l'environnement Open-Music pour développer des processus de génération de paramètres granulaires à partir de graphes ou de fonctions chaotiques et étudié le problème du transfert de données vers MAX.

I.2 Objectifs du document

Le projet du GMEM est rattaché à plusieurs notions qui m'ont été inculquées durant ma formation ATIAM de septembre 2002 à Février 2003. J'ai participé à l'élaboration d'outils d'informatique musicale principalement à partir de deux langages orientés objet (MAX et OM) et j'ai pu me faire une idée de l'enjeu de ces recherches d'une manière pratique mais aussi d'une manière théorique. La rédaction de ce mémoire est pour moi l'occasion de synthétiser les diverses considérations que j'ai pu avoir durant cette année dans ce domaine.
Je vais donc exposer en premier lieu un historique général des problématiques auxquelles j'ai été confronté, reposant sur les évolutions technologiques liées à la musique durant le vingtième siècle. Puis je parlerai de la place de la synthèse granulaire dans les différents modes de création sonore en m'appuyant sur des annexes portant sur les principales techniques de synthèse. Enfin j'exposerai le travail que j'ai effectué au GMEM de mars à Juin 2003.

II Corrélation technologie-composition

II.1 Préambule

Quel est aujourd'hui le travail d'un compositeur ? Quelle est sa démarche, quelles sont ses lois compositionelles, quels sont ses outils ? On pourrait presque dire que chaque compositeur a maintenant ses propres lois et ses propres outils. Pourquoi et comment la musique en est elle arrivée à ce stade ? Pour répondre à cette question, il faut se pencher sur l'histoire musicale particulièrement intéressante du siècle dernier.
En effet, le vingtième siècle, riche en innovations en tous genres, dénombre une multitude d'expériences musicales soit en conservant l'instrumentarium classique (élargir la palette des sons par les modes de jeu ou trouver de nouvelles techniques compositionnelles), soit en utilisant et surtout en cherchant de nouvelles manières de générer des sons, de les organiser,de les contrôler. Disposant de tous ces apports, le compositeur d'aujourd'hui doit trouver ses marques, faire des choix, se forger un langage cohérent, bref trouver les moyens les plus adéquats dans cette multitude de possibilités qui s'offrent à lui pour arriver à ses fins.

II.2 Nouveaux processus de composition et nouveaux matériaux

Les concepts de composition ainsi que les matériaux utilisés dans la musique ont profondément évolué durant le siècle dernier, abolissant la suprématie de l'harmonie tonale, du tempérament, et l'utilisation privilégiée des instruments de musique traditionnels. Les débuts de cette révolution se font sentir avec Shoenberg et ses disciples. Avec le Dodécaphonisme, ils introduisent de nouvelles formes de relation entre les évènements musicaux et créent leur propre formalisme compositionnel, leurs propres contraintes, différentes de celles de Bach et de ses règles de contrepoint jusqu'ici plus ou moins conservées par les compositeurs. A partir de ce moment s'ensuivent une multitude d'innovations liées à la Composition, bien trop nombreuses et diverses pour essayer d'en dresser une liste. Voici néanmoins quelques évènements marquants piochés dans cette très riche histoire :

1939 : Cage : *Imaginary Landscape numéro 1* (première œuvre n'existant que sous forme d'enregistrement).

1944-45 : Percy Grainer et Burnett Cross travaillent à la *Free Music Machine*, fonctionnant à partir d'une notation graphique. Les notations graphiques seront utilisées par Cage, Browne, Busotti, Boucourechliev.

1948 : Au studio d'essais de la Radio à Paris, Pierre Schaeffer invente la musique concrète. L'incident du « sillon fermé » sur le disque isole un fragment sonore de son contexte (*Etude aux chemins de fer*).

1950 : A Cologne, naît la *musique électronique*, avec H. Eimert, K. Stockhausen, …

1951 : Cage présente *Imaginary Landscape numéro 4*, musique aléatoire pour 12 radios, 24 exécutants et un chef.

1952 : Dans *December 1952*, Brown utilise une notation graphique comme support à une musique très libre (aux débuts des années soixante, Bussotti dessinera des partitions graphiques très élaborées).

1954 : *Metastasis* de Xenakis : composition pour cordes à partir d'une représentation graphique de surfaces réglées (voir couverture).

1958 : *Poème électronique* de Varèse et Le Corbusier, où les sons de Varèse voyagent à travers des dizaines de haut-parleurs au sein du pavillon de Phillips dans l'Exposition de Bruxelles, avec une architecture et des projections visuelles conçues par Le Corbusier (avec la collaboration de Xenakis).

1960 : *Kontakte* de Stockausen (piano, percussions et bande), pièce mixte instruments-bande qui aura une vaste descendance avec Berio, Amy, Babbit, Risset, Harvey …

1960 : La Monte Young : *Compositions 1960* (« Draw a straight line and follow it »).

1963 : *Poème pour 100 métronomes* de G. Ligeti.

1964 : *In C*, de Riley. Le courant de la musique « répétitive » sera illustré par riley, Reich, Glass, Adams.

1964 : *Mixtur* de Stockhausen : transformation en temps réel des sons instrumentaux (suivi de *Mikrophonie, Hymen* …).

1966 : Mathews et Rosler présentent leur *Graphic System*, qui permet de spécifier des enveloppes et des courbes de fréquence par le dessin.

1967 : La partition partiellement graphique d'*Archipel I* de Boucourechliev ressemble à la carte d'un archipel suggérant divers parcours d'une île à l'autre.

1977 : Première version de l'UPIC (machine à dessiner la musique conçue par Xenakis) : *Mycenae Alpha* de Xenakis (1978), et oeuvres de Mâche, Eloy, Bernard, Yuasa.

1984 : Usage des échantillonneurs dans les musiques mixtes : diverses oeuvres de Mâche ; Reich : *Different trains* ; Bœuf : *Nocturne.*

II.3. OpenMusic et la Composition Assistée par Ordinateur

Les premières expériences d'utilisation de l'ordinateur dans la composition musicale date des années 50 : en 1955, Caplin implémente le *Jeu de dés* de Mozart et en 1956, Pinkerton suggère un système de composition stochastique qu'il appelle le *banal tunemaker*. Mais c'est avec Hiller qu'on peut commencer à parler de composition automatique ; il crée en 1957 avec Isaacson la première véritable pièce composée avec ordinateur : *Illiac suite for string quartet* en utilisant la théorie de l'information et la théorie générale des systèmes (le matériau de base de cette œuvre est essentiellement engendré par l'algorithme de Monte-Carlo).
Baker écrit vers 1963 le logiciel *Musicomp*, premier logiciel d'aide à la composition, par opposition à la composition automatique. Il peut être vu comme un ensemble d'outils destinés

à résoudre des problèmes spécifiques. On peut mentionner les travaux de Barbaud et Blanchard en France qui reprennent des théories aléatoires pour construire leurs pièces, considérant que la musique oscille entre l'ordre et le désordre et se servant de la théorie des ensembles pour la formalisation.

Les travaux exposés jusqu'ici relèvent de la composition automatique ou algorithmique. Koenig fait un pas de plus dans la problématique du partage des processus compositionnels entre la machine et l'homme : ses programmes *Project1* et *Project2* calculent des structures musicales à partir d'une spécification stricte de la forme et donne des réservoirs de paramètres. Des compositeurs comme D. Cope ou C. Ames, entre autres, mènent des recherches basées sur l'idée d'établir des domaines de paramètres musicaux sur lesquels on impose des structures de contrôle automatique dont l'exécution engendre la pièce musicale.
Il faut bien sur également parler de Xenakis, qui a basé sa musique sur le principe d'indéterminisme et dont je décrirai certaines méthodes dans la section IV.1.4.
Dans les années 70 et 80, Truax et Pope ont essayé d'exploiter l'apport de l'intelligence artificielle en proposant des représentations inspirées notamment de la théorie des graphes, et commencé à utiliser des environnements de programmation modernes tels que *Smalltalk.*

La CAO moderne émerge depuis le milieu des années 80 grâce à plusieurs facteurs dont le plus important est sans doute l'évolution des langages de programmation. L'utilisation d'un langage de programmation oblige le musicien à réfléchir sur le processus même de formalisation et lui évite de considérer l'ordinateur comme une boîte noire qui impose ses choix. Cette CAO contemporaine a été progressivement définie par les contributions de chercheurs comme Pope (ouvrant la possibilité aux compositeurs de définir ses propres objets et méthodes), Orlarey (langage original pour la composition basé sur la lambda-calcul). L'IRCAM s'implique avec constance depuis plus d'une quinzaine d'années dans cette discipline : *Formes* de Rodet et Cointe (entre 1982 et 1985) (26), *PreForm* de Boyton et Duthen (en 1987), *Esquisse* par Baisnée et Duthen avec l'aide d'un groupe de musiciens dont Murail, qui fut une des premières librairies CAO disposant de l'expertise de vrais acteurs de la musique contemporaine, *Crime* par Assayag avec le compositeur Malherbe (premier essai d'environnement général avec résultats visualisés sous forme de partition musicale et intégration d'algorithmes de psychoacoustique) utilisé par bon nombres de compositeurs dont Stroppa, Lindberg, Saariaho.
Le langage *OpenMusic*, créé par l'équipe « représentation musicale » de l'IRCAM fondée en 1992, a bénéficié de tous les travaux exposés précédemment dans sa tentative de construire une démarche cohérente clairement identifiée sous le nom de CAO. La CAO est fort utilisée en Europe et commence à se diffuser dans le monde anglo-saxon. « Des compositeurs aussi divers que Murail qui s'en sert pour l'engendrement de l'harmonie et du rythme, Ferneyhough qui y a découvert un défi à la mesure de son goût pour la complexité extrême, Malherbe qui y voit l'outil lui permettant la construction de nouveaux modèles pour chaque pièce, Lindberg qui vise au contrôle vertical complet depuis la forme jusqu'aux éléments syntaxiques du discours » (2). *OpenMusic* est né de la confrontation avec tous ces créateurs qui ont chacun une vision différente du rôle de la technologie. La CAO a profondément bouleversé les habitudes et les méthodes en composition musicale, elle opère une remise en question qualitative des modalités de la création. Son aspect révolutionnaire et le plus polémique est qu'à la notion d'unicité de l'œuvre et de son créateur, elle tend à substituer l'idée du modèle d'œuvre et de la coopération entre créateurs et scientifiques.

II.4. Les compositeurs

Boulez dit: « toute théorie musicale se base sur un réseau plus ou moins complexe de compromis, auxquels elles donnent des répondants de nature scientifique, ou des garants d'ordre philosophique » (6). Il en toujours été ainsi, les évolutions philosophiques et scientifiques balisent et élargissent toujours un peu plus le champ d'investigation de la création musicale et c'est pourquoi : « progressivement, à l'exemple de leurs consœurs scientifiques, les théories musicales se sont vues déloger de l'état de dogme à celui plus précaire d'hypothèse de travail » (6).
On peut considérer qu'aujourd'hui, les outils créés en matière de manipulation sonore, de CAO, sans parler de l'apport des divers domaines scientifiques permettent réellement de tester, d'explorer de manière très concrète les théories compositionnelles les plus ardues : « un domaine d'expérience s'offre à la norme et l'organisation. Ce qui n'était jusqu'alors saisi que d'une façon très labile et incertaine se convertit désormais en un ordre de problèmes entièrement formulés. Les conflits spéculatifs cèdent le pas aux débats argumentés, fondés sur une articulation maîtrisée et une définition opératoire de la nature de l'objet. L'acoustique théorique, la phonétique, la psychoacoustique, la théorie des modèles de perception, l'intelligence artificielle sont autant de disciplines récentes visant à conceptualiser l'empiricité diffuse de l'échange et à déterminer les régions indécises du phénomène » (14). Selon Dufourt, c'est sa génération qui a pour mission de « tirer de ces mutations conceptuelles une nouvelle esthétique sans laquelle le travail de composition, inconscient de ses normes et de ses enjeux, ne peut que se fourvoyer dans un inventaire de gestes métaphoriques de leur propre impuissance » (14).
Le mouvement « spectral » (très inspiré par Scelsi) auquel appartient Dufourt comprend des gens comme Gérard Grisey et Tristan Murail qui sont d'ailleurs des acteurs à part entière de l'histoire de la CAO. Ils reprennent ou sont à l'origine d'idées très représentatives de ce qu'est la composition aujourd'hui. Je perçois leur démarche comme un espèce de « ras-le bol » de l'utilisation de la combinatoire ou du stochastique à outrance dans la musique et propose de nouvelles manières très intéressantes de considérer le matériau musical. Selon Tristan Murail : « il faut se débarrasser des classifications sclérosantes ; il y a erreur dés l'origine ; le compositeur ne travaille pas avec 12 notes, x figures rythmiques, x symboles d'intensité permutables et corvéables à l'infini, il travaille avec des sons et du temps » (17).
Mais comment organiser l'espace des fréquences et celui du temps si on refuse les unités habituelles ? Il n'y a plus de repère absolu, il faut utiliser des repères relatifs et travailler sur des différences, des relations entre les éléments eux-mêmes. On s'intéresse plus aux relations entre les objets qu'aux objets eux-mêmes, le temps est organisé par flux et non par secteurs.
Pour eux, le principal outil est le spectre. Quelle que soit leur nature, harmonique, inharmonique, linéaire, non-linéaire, les spectres peuvent évoluer dans le temps : s'enrichir ou s'appauvrir, dériver de l'harmonicité à l'inharmonicité, de la linéarité à la non-linéarité. C'est ainsi que vont naître des formes, micro-formes ou macro-formes, où tout sera lié et interdépendant ; fréquences, durées, combinaisons de fréquences, donc harmonie et même orchestration.
Le but à atteindre est selon Gérard Grisey que l'événement musical doit dans son ultime réalisation surgir simplement, en dépit des nombreuses stratifications qui le composent. Ecriture et perception seraient ainsi dépendantes l'une de l'autre (12).

III Nouveaux instruments et contrôles

III.1 Timbre et Causalité

Deux évolutions essentielles se sont produites dans la musique au cours du vingtième siècle : D'une part l'émergence du timbre comme référent fondamental (8) :

1913 : Luigi Russolo publie *Arte dei rumori (l'art des bruits)*, manifeste des bruitistes et organise en 1914 des concerts d'un orchestre de bruiteurs.

1917 : Edgar Varèse écrit : « La musique qui veut vivre et vibrer a besoin de nouveaux moyens d'expression, et la science seule peut lui insuffler une sève adolescente. Je rêve les instruments obéissant à la pensée, et qui avec l'apport d'une floraison de timbres insoupçonnés se prêtent aux combinaisons qu'il me plaira de leur imposer et se plient à l'exigence de mon rythme intérieur. »

1931 : *Ionisation* de Varèse, ne comporte que des instruments à percussions.

1938 : Cage réalise plusieurs œuvres pour « piano préparé » : à partir d'une tablature bien connue, celle du clavier chromatique, les gestes du pianiste commande un ensemble de sons non tempérés.

1948 : au studio d'essais de la Radio à Paris, Pierre Schaeffer invente la musique concrète. L'incident du « sillon fermé » sur le disque isole un fragment sonore de son contexte (*Etude aux chemins de fer*).

…

Et d'autre part le développement d'outils matériels nouveaux en passant de la mécanique (technologie primitive des moyens de production du phénomène sonore) à l'électricité, l'électronique, l'enregistrement, la transmission, l'ordinateur, l'informatique (8) :

1906 : Naissance de l'électronique : Lee de Forest invente la lampe triode, appelée *audion*, car il visait la production d'onde sonore par des moyens électriques. La triode permet d'amplifier un signal électrique. L'amplification électrique sera au point dans les années vingt, favorisant le développement de la radio.

1929 : le cinéma parlant inscrit le son sur une piste optique. Par la suite, Norman MacLaren écrira directement sur cette piste pour produire le son, préfigurant la synthèse.

1934 : lors de la première transmission de musique en stéréophonie, Leopold Stokowski préfère tenir la console à Washington.

1950 : à Cologne, naît la *musique électronique*, avec H. Eimert, K. Stockhausen, …

1957 : Max Mathews réalise aux Bell Laboratories le premier enregistrement numérique et la première synthèse numérique des sons.

1961 : Mathews écrit Music3, programme de synthèse qui préfigurent les programmes Music4, 5, 10, Cmusic, Csound (voir annexe 1), les synthétiseurs modulaires et la programmation objet.
…

Ces différentes évolutions technologiques et philosophiques intimement liées vont littéralement briser la relation ancestrale entre timbre et cause qui était que « la vibration sonore, produite par une cause physique naturelle, un corps matériel, est dirigée par une pensée qui s'extériorise par un autre intermédiaire naturel : le geste s'appliquant à cet objet » (8).
La relation causale directe est désormais dissoute, on a dissocié le timbre et la causalité, pourquoi, comment :
- L'apparition de l'amplification amène une notion de relais énergétique, où, contrairement à la chaîne instrumentale, les ordres de grandeurs peuvent être totalement différents.
- Le rapport entre quantité d'information et la quantité d'énergie pour la communiquer est de plus en plus grand.
Ceci conduit à des changements d'échelle considérables entre causes et effets. Normalement la portée de la voix est de quelques centaines de mètres, maintenant elle ne connaît aucune limite, on peut envoyer n'importe où un message sonore ou visuel grâce aux ondes hertziennes à la vitesse de la lumière.
On peut résumer le nouveau contexte musical comme suit :
Aujourd'hui, la possibilité de confectionner le matériau de base de la musique dans ses moindres détails en travaillant sur les modèles physiques de l'évènement sonore permet de se dégager des contraintes mécaniques fixées par les instruments traditionnels. Le compositeur peut construire ses objets sonores en manipulant des paramètres acoustiques : fréquence, intensité, spectre, enveloppe… et peut, de façon analogue, organiser ses objets en les composant arbitrairement et en les exécutant par le biais d'une machine. Ces possibilités ne sont pas limitées à la création de sons originaux, mais préludent à l'apparition de nouvelles règles formelles : d'une syntaxe en accord avec les exigences actuelles de nombreux compositeurs. Le développement de la technologie numérique a rendu les ordinateurs plus puissants et plus rapides, ce qui a permis l'apparition de processeurs numériques fonctionnant en temps réel, c'est à dire calculant à une vitesse telle que notre oreille n'est pas capable de déceler le retard entre le temps de calcul et le résultat sonore produit par les calculs eux-mêmes. Les machines en temps réel ont permis de réintroduire le geste, et par conséquent une certaine forme de pratique instrumentale qui autorise des interprètes à exécuter des « partitions » de musique informatique.

III.2 Contrôle

On rencontre dans l'utilisation des instruments de musique traditionnels des gestes de diverses natures et fonctions. La catégorisation de ces gestes (9) est très utile aux recherches sur le contrôle de nouveaux instruments. On peut en effet penser que l'ancestralité de la musique aura vu quasiment toutes les formes possibles et imaginables de gestes exécutables. L'étude de ces gestes représente donc un capital important dont on peut s'inspirer pour le contrôle de différents générateurs sonores.

Une fois répertoriés, les gestes du musicien à prendre en compte pour le contrôle de la synthèse sonore doivent être captés, c'est à dire traduits en signaux électriques puis numériques. L'acquisition peut être *directe* ou *indirecte.*

- acquisition directe : différents types de capteurs mesurent l'ensemble des mouvements impliqués dans un même geste. C'est la cas, par exemple, de l'acquisition des mouvements complexes de manipulation d'un archet utilisé pour le jeu d'un instrument à cordes. En acquisition directe, la tendance est de capter des grandeurs physiques de base comme des forces, des déplacements, des accélérations… Le plus souvent chaque type de grandeur nécessite un type de capteur particulier.
- Acquisition indirecte : le geste est déduit de l'évolution des propriétés structurelles du son produit par l'instrument. Le seul type de capteur utilisé dans ce cas est un microphone, c'est à dire essentiellement un capteur de pression. Ce type d'acquisition fait ensuite appel à des techniques de traitement de signal temps-réel pour extraire l'information utile, ce qui nécessite l'utilisation de dispositifs de calcul relativement puissants.

L'acquisition directe fournit des informations plus ou moins précises sur chacun des mouvements impliqués par le geste, tandis que l'acquisition indirecte mesure l'effet global du geste sur le signal sonore produit. Nous nous intéresserons ici uniquement à l'acquisition directe.

III.2.1 Les capteurs

Le choix des capteurs représente une étape importante dans la conception d'un nouvel instrument. De nombreuses technologies de transducteurs sont disponibles sur le marché. La plupart sont issus de recherches menées pour de tout autres objectifs que la production musicale. Néanmoins, plusieurs de ces technologies s'adaptent très bien à la nouvelle facture instrumentale. C'est le cas des capteurs utilisant des résistances sensibles à la pression, à une force *(force sensing resistor)*. Par ailleurs les capteurs à effet Hall ou utilisant l'effet piézo-électrique sont respectivement adaptés à la captation des mouvements d'amplitude ou de force. Quant au captage de mouvements de plus grande amplitude, les principales techniques font appel à des systèmes vidéo, à ultra-sons, à rayons infrarouges ou bien électromagnétiques.

Il est important de signaler que le choix des capteurs n'est pas l'élément essentiel dans l'élaboration d'un nouvel instrument, lorsqu'elle constitue la priorité, le résultat s'avère souvent décevant.

Une fois les capteurs choisis, il faut convertir les signaux électriques qu'ils produisent en signaux numériques. On peut par exemple employer la standard MIDI très répandu. Le protocole MIDI, *Musical Instrument Digital Interface* est apparu au début des années 1980 à l'initiative des constructeurs d'instruments numériques. Il a beaucoup facilité l'essor du temps réel en imposant un standard de description numérique d'évènements musicaux simples. Il faut mentionner que MIDI prévoit la transmission d'information « system exclusive » correspondant à des appareillages spécifiques.

III.2.2 Les contrôleurs gestuels

La plupart des contrôleurs gestuels réunissent plusieurs capteurs au sein d'une même unité. Ils constitue le premier bloc fonctionnel de l'instrument virtuel. On peut distinguer trois types de contrôleurs gestuels selon qu'ils imitent, s'inspirent ou se détournent des interfaces de contrôle gestuel des instruments traditionnels :

- Dispositifs *imitatifs* : les claviers à touches piano et les guitares MIDI en sont des exemples type. Leur utilisation bénéficie pleinement de l'expertise gestuelle développée au fil du temps par nombre d'instrumentistes. Leur fabrication se fait dans un souci d'exacte ressemblance avec les dispositifs de contrôle des instruments originaux.
- Dispositifs *analogues* : ils rappellent la forme des interfaces rencontrées dans les instruments traditionnels. Mais dans ce cas, le but est surtout de conserver une analogie avec une interface traditionnelle pour les utiliser dans des situations musicales autres, tout en exploitant éventuellement des propriétés différentes : claviers légers ou le *SuperPolm.* Ce dernier est un contrôleur qui rappelle la forme du violon mais qui peut être utilisé pour déclencher ou contrôler des événements de synthèse sonore n'ayant rien à voir avec la simulation des sonorités d'instruments à cordes. Il est par exemple utilisé pour contrôler de la synthèse granulaire.
- Dispositifs *alternatifs* : conçus pour utiliser des gestes qui ne font pas partie de l'expérience instrumentale musicale traditionnelle. Ils peuvent par exemple capter les déplacements du corps de l'instrumentiste. De nombreuses interfaces ad hoc développées ces dernières années entrent dans cette catégorie, comme les « gants de données » ainsi que les interfaces couramment utilisées dans d'autres domaines que la musique comme les tablettes graphiques.

La comparaison des différents types de contrôleurs gestuels s'avère difficile en raison de la diversité du mode de fonctionnement des capteurs tout autant que des signaux captés. On peut néanmoins profiter des travaux menés dans les domaines de l'interaction homme-machine sur la comparaison entre dispositifs d'entrée fondée sur leur décomposition en unités de captations primaires. Les unités de captation primaire sont classées en fonction de leur gestion de l'espace, de la nature de la grandeur physique captée et de la résolution à laquelle cette grandeur est captée. A ceci s'ajoutent les possibilités combinatoires entre les différentes unités de captation. Soit, de manière explicite :

- Les six degrés de liberté possibles : translation et rotation selon les trois directions X, Y et Z.
- Les grandeurs physiques captées : position P et sa dérivée dP, force F et sa dérivée dF, angle A et couple T ainsi que leurs dérivées dA et dT pour les rotations.
- La résolution de chaque variable.
- Le type de combinaison entre les unités de captation : « merge », « layout » ou « connect »

III.3 Réalisations

Les tentatives dans le domaine du contrôle gestuel sont très nombreuses. En voici quelques unes parmi les plus essentielles :

1969 : J.Chowning simule par programme d'ordinateur des mouvements rapides des sources sonores. Il réalise aussi un dispositif analogique où cette spatialisation est commandée par un Joystick.

1969 : Système hybride de synthèse GROOVE, avec contrôleur gestuel en temps réel (Mathews et Moore, 1970). Mathews développe sur ce système le concept du « conductor program », dans lequel le musicien contrôle au moment du temps réel non pas tous les paramètres du son, mais, à la façon d'un chef d'orchestre, seulement certains aspects très significatifs, comme le tempo et l'équilibre des voix.

1972 : A l'Université d'Illinois, S.Martirano développe le SAL-MAR, synthétiseur qui réagit de façon surprenante aux commandes gestuelles, et qu'il considère comme une œuvre autant que comme un instrument.

1972 : Mathews : violon électrique. Ce violon peut aussi fonctionner comme capteur des gestes d'un violoniste pour commander d'autres sons.

1974 : J.Appleton et S.Alonso réalisent le *Synclavier*, premier synthétiseur numérique, dans lequel la commande fait appel à un clavier mais aussi une roue et à des touches de fonction, qui permettent de déclencher et modifier des séquences. Les méthodes de contrôle tirent parti des expériences de Mathews et Moore sur GROOVE.

1976 : Capteurs gestuels rétroactifs commandant des synthèses par simulation de modèles physiques : C.Cadoz, A.Luciani, J.L.Florens, ACROE, Grenoble. Ces recherches préfigurent les réalités virtuelles, ont abouti à des utilisations musicales.

1980 : Mathews introduit le contrôle d'instruments « intelligents », qui ont une connaissance de la musique à jouer par une « baguette radio ».

1983 : Apparition de MIDI.

1984 : M.Waisvisz présente *The hands*, un capteur de gestes pour la musique développé à l'institut STEIM d'Amsterdam.

1989-90 : M.Puckette publie MAX, environnement graphique modulaire pour la programmation d'interactions en temps réel.

1989 : J.C.Risset et S. VanDuyne mettent en œuvre au MIT l'interaction ordinateur-piano acoustique sur la Yamaha Disklavier.

1989 : les *hyperinstruments* de T.Machover, extensions numériques de gestes ou d'instruments comme le violoncelle. S. de Laubier développera quant à lui des *métainstruments*, puis des *Midi Formers* qui détournent Midi pour des applications proches de la musique concrète.

1990-91 : D.Buchla produit le *Thunder*, puis le *Lightning*, deux capteurs de gestes utilisés dans différentes œuvres.

1992 : Le VL-1 de Yamaha est le premier synthétiseur commercial dont les commandes peuvent s'exercer sur des paramètres ayant une signification physique.

Quelques systèmes gestuels spéciaux :

Le gant de données (*dataglove*) : créé par J.Lanier et J.J.Grimaud, apparu aux alentours de 1985, exploitant des travaux menés à la NASA auparavant. Il utilise des fibres optiques et un capteur de position et rotation montés sur un gant ordinaire. Ces capteurs donnent à l'ordinateur des informations sur la flexion des doigts, l'orientation et les déplacements de la main. Ce système est uniquement capteur.

Le gant à retour tactile : réalisé par R.Stone, utilise 20 ballonnets au contact de toutes les phalanges des paumes de la main. Chaque ballonnet est relié à un compresseur commandé par l'ordinateur et produit localement une pression engendrant la sensation d'un contact. Ce système est à la fois capteur et effecteur. Son retour est uniquement tactile, il permet de connaître le contact avec un objet mais pas sa solidité, déformabilité, sa masse… L'énergie en jeu est très faible.

Le clavier rétroactif modulaire de l'ACROE : premier dispositif à retour d'effort. Il est le résultat de travaux portant sur la simulation par modèle physique des instruments de musique et des objets physiques en général, l'interaction gestuelle instrumentale avec ceux-ci et les systèmes à retour d'effort. Ce système n'est pas seulement haptique. Les dispositifs mis au point à l'ACROE permettent la modularité et la versatilité de l'espace gestuel tout en respectant la catégorisation forte du geste instrumental. Sur la base d'un module permanent, ils permettent par l'adjonction « d'habillages morphologiques » spécifiques, de développer toutes les catégories du geste d'excitation. Il est possible de jouer d'un archet, d'un manche de commande, d'un clavier de piano… Un habillage morphologique est un dispositif qui transforme l'espace capteur-effecteur en l'espace gestuel instrumental voulu.

III.4 Lien instrumental

Dans un instrument virtuel, il faut qu'il y ait un élément traduisant la correspondance entre les variables gestuelles, paramètres issus du contrôleur et les variables de synthèse que sont les paramètres d'entrée de l'algorithme de calcul des échantillons sonores. Cet élément est le *mapping* (lien instrumental), c'est lui qui donne un sens aux paramètres de sortie du contrôleur, il doit permettre au musicien de contrôler l'instrument selon l'approche voulue tout en assurant précision et expressivité. Il y a deux cas selon que le synthétiseur contrôlé repose sur un modèle signal ou un modèle physique :

Dans ce dernier cas, une partie du lien instrumental est déjà intégrée à l'algorithme de synthèse sous forme de contraintes mécaniques ou acoustiques. Un meilleur exemple en est le « Système de modélisation et simulation d'instruments et d'objets physiques pour la création musicale et l'animation d'images » CORDIS-ANIMA de l'ACROE, langage qui repose sur une représentation du monde physique en objets élémentaires (élément matériel et élément de liaison) et qui est capable de simuler des interactions entre un objet physique et un système numérique. L'interaction découle directement des algorithmes de synthèse.

Quant à l'accés de type jeu instrumental de synthétiseurs fondés sur des modèles de signaux, le plus simple est de représenter le mapping par une « couche logicielle » implantant des règles de correspondance entre le vecteur de paramètres de sortie du contrôleur et le vecteur de paramètres d'entrée de l'algorithme de synthèse.
Un avantage de la représentation du mapping sous forme logicielle réside dans la possibilité de l'organiser en deux (voire plusieurs) niveaux indépendants : choix du contrôleur, choix du type d'algorithme de synthèse, et la possibilité de modifier la correspondance laisse entrevoir le choix entre plusieurs niveaux offrant un accés de complexité croissante. La relation entre homme et machine est dans ce cas moins « réelle » que pour les synthétiseurs par modèle physique qui permettent le « retour d'effort », de communiquer avec la machine par l'intermédiaire de transducteurs selon trois canaux sensoriels : acoustique, visuel et gestuel (10).

Tout est « permis » en matière de contrôle gestuel. Comme on l'a vu, on peut aborder le problème de différentes façons, partir d'une base de contrôle de type traditionnel pour un contrôle de type traditionnel ou justement pas, ou bien partir sur de nouvelles interfaces en considérant et en cherchant des aspects essentiels du geste, par exemple l'interaction bimanuelle (3), qui cherche à comprendre fondamentalement la complémentarité entre les deux mains, c'est un aspect évidemment très important des nouveaux contrôles gestuels. La main gauche (pour les droitiers) n'est pas « une mauvaise main droite », mais elle a ses fonctions propres de même que la main droite, elle agit avec elle de manière coopérative pour bon nombre de tâches. De manière générale, il semblerait que la main droite opère dans un référentiel fixé par la main gauche, que la « granularité » des actions de la main gauche est supérieure à celle de la main droite, que la main gauche agit avant la main droite, et que la main droite est préférée car elle occupe les degrés distaux de la chaîne cinématique.

III.5 Conclusion

Dans le foisonnement de considérations que nous avons exposé dans cette partie, on voit que les possibilités en termes d'interaction homme-machine sont immenses, voire illimitées. Ce domaine suscite l'intérêt d'une communauté importante qui travaille dans des directions variées. Il suffit de voir l'éclectisme des programme du congrés NIME (New Interfaces for Musical Expressions) :
Audiopad : a tag-based interface for Musical Performance.
Multi-instrument virtual instrument keyboard.
Development of a virtual violin bow controller haptic human-computer interface.
Cutaneous grooves : composing for the sense of touch..
Noise gate for meta-saxophone.
The electronic tabla contrôler.
Sonar interface for unobtrusive man-machine interfaçage.
… (session 2002)

Cette immensité dans les possibilités est un bien mais aussi un inconvénient, car son exploration pour la découverte de systèmes pertinents au point de vue de la sensibilité, de l'expressivité, du musical est très ardue.

IV La synthèse granulaire

IV.1 Historique

IV.1.1 Norbert Wiener

La notion de quantum sonore apparaît avec N. Wiener en 1925 qui lors d'une conférence sur la physique quantique utilise comme référence la musique. Il donne des détails sur la relation entre temps et fréquence, il énonce que la précision dans le domaine temporel entraîne des imprécisions dans le domaine fréquentiel et vice-versa. Il suggère ainsi que si une note de 20 Hz est jouée pendant moins de 1/20 de seconde elle ne produit pas de son. Apparaît alors une notion de quantum sonore dès lors qu'on connaît la taille minimum d'une particule sonore (en fonction de sa fréquence) d'où l'idée, similaire à celle de la mécanique quantique qui décrit la matière, de pouvoir décrire les sons en termes de taille et nombre de grains.

IV.1.2 Denis Gabor

En 1946 le physicien prix Nobel D. Gabor fabrique une machine utilisant un système de grains pour reproduire des sons. Il argumente ses découvertes par un article appelé « Theory of communication » suivi un an après par « Acoustical Quanta and the Theory of Hearing ». Il y parle notamment des problèmes de l'analyse de Fourier. Gabor dit que alors que les mathématiques y sont absolument correctes, cette théorie ne convient pas à la description des sons notamment ceux à fréquences variables. Un autre problème est qu'utiliser des ondes sinusoïdales implique que la durée du signal est infinie. Gabor présente l'idée d'utiliser les théories de Physique quantique pour l'étude des signaux sonores.

Gabor effectue des études sur les seuils de discrimination. Pour cela il utilise les résultats des expériences menées par Buerck, Kotowski, Lichte puis plus tard par Shower et Biddulph. Il examine la durée nécessaire d'un son pour reconnaître son pitch à différentes fréquences. Ses études montrent un certain nombre de choses :

- Les fréquences entre 500Hz et 1000Hz doivent être jouées au minimum pendant 10 ms avant d'être perçues comme des hauteurs.
- Si une fréquence change, il faut 2 fois le temps qu'il a fallu pour entendre la première fréquence pour discerner le changement. Cette durée de reconnaissance de changement varie avec la fréquence. A basses fréquences, l'homme discerne plus vite.
- Le discernement d'un changement d'amplitude prend minimum 21 ms.

Gabor émet alors une théorie mathématique sur les surfaces de seuil de perception. Il utilise ces idées pour développer un certain nombre de machines. Le *Tempophon* fabriqué par la compagnie allemande *Springer* est une machine à changer le pitch ou la durée de documents sonores et est directement issue d'une machine de Gabor. En 1963 Herbert Eimer utilise le Tempophon pur une composition électronique appelée « Epitaph für Aikichi Kuboyama ».

IV.1.3 Abraham Moles

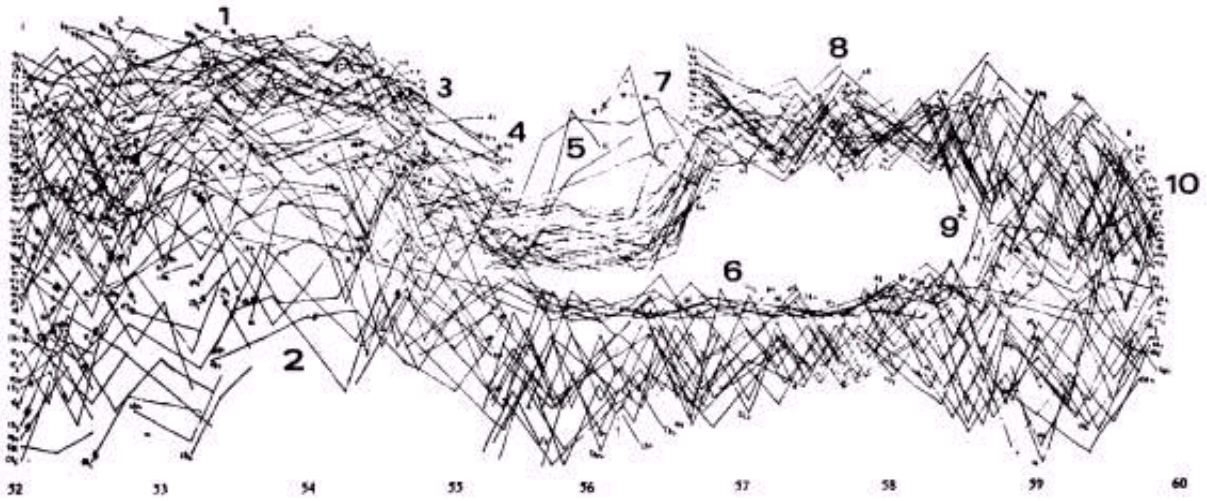
Les recherches de moles sur la théorie de l'information l'ont amené à la conclusion suivante : pour pouvoir définir et déchiffrer un message, qu'il soit visuel ou sonore, on doit prendre en considération les caractéristiques psychophysiologiques du récepteur, car elles seront circonstancielles. « Un message est un groupe fini, ordonné, d'éléments de perception puisés dans un répertoire et assemblés en une structure. Les éléments de ce répertoire sont définis par les propriétés du récepteur. »
Moles est le premier à dire que le nombre de sensations que reçoivent nos organes psychophysiologiques est quantifiable. Moles a étudié le pouvoir de l' ouïe pour résoudre des petites différences de fréquence et d'amplitude et parle d'un répertoire d'éléments audibles qui est plus ou moins limité à 340000. Moles représente « l'atome sonore » comme une cellule tridimensionnelle ayant pour côté le seuil différentiel de fréquence $\Delta F/F$, le seuil différentiel d'intensité $\Delta L/L$, le seuil différentiel de durée $\Delta T/T$. Cette formulation du quantum sonore est à peu près la même que celle de Gabor mais est plus précise car grâce à elle on peut répertorier le nombre existant de « quanta de sensation ».

IV.1.4 Iannis Xenakis

Xenakis a été le premier musicien à créer une théorie structurelle de la composition où les grains sonores sont les éléments constitutifs de base.
Avant d'utiliser ces idées dans la musique électronique, il réalise des pièces instrumentales basées sur le concept de cellule sonore dont les paramètres sont régis par des lois statistiques.
Dans Pithoprakta pour orchestre à cordes écrit en 1955-56, il utilise le glissando comme cellule élémentaire sonore et décrit la gestion des paramètres (30) :

- la durée est constante
- la masse de pitch varie librement
- la densité de sons est à chaque moment constante
- la dynamique est *ff* sans variation
- le timbre est constant
- les vitesses qui déterminent la « température » sont réparties suivant une loi gaussienne

Le terme de température est du à l'analogie avec un gaz où la température dépend de la vitesse des molécules qui le composent. La vitesse d'un glissando est sa variation de pitch en fonction de sa durée. Il calcule un grand nombre de vitesses distribuées sur 58 valeurs distinctes qu'il traduit ensuite sous forme de partition traditionnelle. « La distribution étant gaussienne, la configuration macroscopique est une modulation plastique du matériau sonore ».



Graphe de glissandi pour Pithoprakta

Il déduit de ses expériences instrumentales certains postulats quant à l'utilisation de quanta sonores:
On peut contrôler des transformations continues d'une grande quantité de sons continus ou granulaires. Les paramètres de densité, durée, registre, vitesse… peuvent tous être soumis à la loi des grands nombres. On peut donc par le biais de moyennes et de déviations façonner ces ensembles et les faire évoluer dans différentes directions. La transformation la plus connue est celle qui va de l'ordre au désordre, ou vice-versa, ce qui introduit le concept d'entropie.
Une transformation peut être explosive quand les déviations à la moyenne deviennent subitement énormes, des atmosphères sonores très raréfiées peuvent être façonnées et contrôlées à l'aide de formules comme celle de Poisson.
Il formalise ses idées en décrivant les sons comme une succession temporelle « d'écrans » contenant une certaine densité de vecteurs à deux composantes, intensité et fréquence qu'il contrôle d'une manière statistique : « un livre d'écrans équivaut à la vie d'un son complexe ».
Inspiré des idées de Maxwell-Boltzmann sur la description des états gazeux en terme de mécanique statistique, il établit des méthodes macroscopiques pour contrôler ses micro-évènements. Il crée les bases théoriques pour pouvoir composer des sons complexes en partant d'une distribution aléatoire constituée par des milliers de grains sonores dans le temps. Il appellera cette distribution « nuage de points qui évolue », pensant qu'avec l'utilisation des principes quantiques du son on peut créer des sons entièrement nouveaux.
Xenakis propose une unité compositionnelle de haut niveau, dans laquelle les grains seraient organisés en entités nommés trames, qui sont constituées à leur tour par des rectangles occupés ou inoccupés. Ces trames représentent des plans de temps-fréquence dans lesquels se trouvent disséminés des centaines de grains (le nombre de grains est déterminé par un facteur de densité d). La première composition granulaire de Xenakis est *Analogique A-B* en 1958-59. *Analogique A* est écrit pour orchestre, la partition est la transcription musicale de processus issus de méthodes stochastiques. *Analogique B* consiste en quatre pièces électroacoustiques diffusées à travers 8 haut-parleurs. Les deux parties sont faîtes pour être exécutées simultanément.

IV.1.5 Curtis Roads

En 1975, C. Roads essaie de vérifier et d'expérimenter les théories du quantum sonore de Gabor. Il construit un système informatique automatisé pour la génération de grains sonores synthétiques à l'Université de Californie à San Diego, en utilisant le système informatique Borroughs B6700, avec le programme Music V. Dans ce système, chaque grain a une durée et une fonction d'amplitude fixes pendant que les fréquences, formes d'onde et amplitudes sont variables. Un paramètre très important est la forme de l'enveloppe du grain, il utilise une enveloppe qui combine la courbe de Gauss suggérée par Gabor et le rectangle suggéré par Xenakis obtenant un début et une chute légère et en même temps un sommet soutenu. Ce système pouvait générer 32 grains simultanés et des densités de 1600 grains par seconde. Roads est l'auteur de nombreux ouvrages dédiés à la synthèse granulaire.

IV.1.6 Barry Truax

B. Truax dans les années 80 construit son propre système de synthèse granulaire, installé à l'Université de SFU au Canada et au GMEB de Bourges. La diffusion de ses travaux musicaux et de recherche a contribué à l'intérêt de la communauté scientifique et musicale pour cette forme de synthèse dans les années 90 qui ont vu l'enrichissement des techniques granulaires.

IV.2 Paramètres en synthèse granulaire

On peut regrouper les techniques de synthèse granulaire en deux grands groupes :

- les techniques qui font la modélisation des sons traditionnels instrumentaux et vocaux
- les techniques qui n'ont pas besoin d'analyse à priori : QSGS (Quasi-Synchronous Granular Synthesis), AGS (Asynchronous Granular Synthesis) et la granulation temporelle d'un ou plusieurs échantillons.

Je parlerai essentiellement du deuxième groupe, car il est représentatif des techniques sur lesquelles j'ai travaillé.

Dans la QSGS, il s'agit de générer un ou plusieurs flots de grains, chaque grain suivant un autre grain avec un délai variable. Cette période de délai peut être plus ou moins régulière. Quand elle est régulière on obtient une fonction périodique et on peut analyser cette technique comme un cas particulier d'amplitude modulée (Annexe 3).
Lorsqu'elle est irrégulière, les possibilités sonores sont décuplées mais on perd le contrôle sur la phase et donc sur le contenu spectral. On peut obtenir des effets intéressants mais uniquement de manière intuitive.

Avec l'AGS, le concept linéaire des flots de grains est abandonné. Les grains sont distribués d'une façon statistique sur une durée spécifique dans des régions inscrites sur le plan temps-fréquence (principe de Xenakis). Ces régions sont appelées nuages, unités avec lesquelles un compositeur travaille.
Les paramètres des nuages sont :

- temps du début et durée du nuage
- enveloppe d'amplitude du nuage

- durée du grain
- enveloppe des grains
- forme d'onde des grains
- bande de fréquence du nuage, spécifiée en général par deux pentes qui forment deux bornes de fréquence (inférieure et supérieure), entre lesquelles les grains sont éparpillés
- densité de grains par seconde (variable au cours du temps)
- dispersion spatiale du nuage

Je vais maintenant parler du rôle de ces différents paramètres.

IV.2.1 Durée du grain

Les limites psychoacoustiques proposées pour la perception du quantum sonore varient entre 10 et 60 ms. En réalité, on peut aller au delà de ces valeurs. Il faut prendre en compte les effets de la périodicité ou apériodicité des grains et la densité granulaire car ces deux facteurs vont affecter nettement les effets de la durée du grain. Pour des grains périodiques, si le grain contient moins d'une période, on aura des produits de modulation. Cet effet peut être intéressant, utilisé dans un nuage de grains où on aurait la sensation d'une hauteur présente mais altérée par les produits de modulation.
Roads propose 3 classes de durées pour les grains :
- durées constantes
- durées aléatoires (comprises entre deux limites, typiquement 10 et 100 ms)
- durées dépendant de la fréquence

Pour des durées constantes, on a un effet de modulation d'amplitude. Pour des grains d'une durée Dg, la fréquence centrale de modulation d'amplitude est de 1/Dg. Par exemple, pour des grains de 100 ms, on a un effet de tremolo autour de 10 Hz. La fréquence à l'intérieur des grains est perçue comme une région formantique dans le spectre.
La durée du grain influe également au niveau de la largeur de bande du spectre. L'effet que l'on obtient en diminuant la durée est un élargissement du spectre.
C'est autour de 50 ms que se rencontrent les effets psychoacoustiques possibles ; autour de ce seuil et au dessus de lui, les qualités de timbre du grain dominent, et au dessous, la fusion de la proportion d'audio des fragments de l'onde du grain dominent (24).
Roads propose une table qui décrit les effets de la durée du grain mais sans tenir compte de la densité et de la largeur de bande du nuage granulaire :

Durée du grain	Taux de fréquence de l'enveloppe de modulation	Effet
200 µs	5000 Hz	Bruiteux, désintégrations des particules
500 µs	2000 Hz	
1 ms	1000 Hz	
10 ms	100 Hz	Battement, gazouillement, glouglou
50 ms	20 Hz	
100 ms	10 Hz	Trémolo apériodique, froissement de la position spatiale
200 ms	5 Hz	

IV.2.2 L'enveloppe

Dans la conception originale de Gabor, l'enveloppe des grains est gaussienne, mais elle peut prendre différentes formes suivant ce que l'on veut faire :

- Roads propose une enveloppe quasi-gaussienne
- pour l'analyse synthèse une fenêtre de Hanning est adéquate car elle est aussi utilisée en conjonction avec la transformée de Fourier rapide
- pour la synthèse en temps réel, il peut être nécessaire d'utiliser une simple enveloppe à segments pour économiser l'espace mémoire et le temps de calcul
- l'enveloppe du grain peut également être fonction de la fréquence, une telle corrélation est la caractéristique de la transformation en ondelettes qui est un genre de synthèse granulaire.

IV.2.3 Effets de la forme d'onde

Roads utilise des formes d'ondes synthétiques allant de la sinusoïde pure à l'addition de n sinusoïdes. Il décrit en termes de couleurs les différentes manières d'utiliser la forme d'onde des grains dans les nuages qu'il crée:

- monochrome : utilisation d'une unique forme d'onde
- polychrome : utilisation d'au moins deux formes d'onde différentes
- transchrome : les formes d'onde des grains évoluent pendant le déroulement du nuage

Dans le cas polychrome, plusieurs tables d'onde sont définies et la sélection de la forme d'onde des grains se fait de manière aléatoire. Pour le cas transchrome, l'évolution de le forme d'onde se fait de manière progressive, par exemple l'évolution d'une sinusoïde pure à une forme d'onde additive à bande limitée en élargissant peu à peu le spectre. La durée très courte des grains fait que cette technique donne souvent des sons très bruités, des textures distordues.

IV.2.4 Effets de la bande de fréquence

Roads propose deux classes de spécifications de fréquence :

- *cumulus*, où les fréquences des grains sont définies aléatoirement entre une borne inférieure et une borne supérieure qui peuvent évoluer au cours du temps
- *stratus*, où les fréquences sont tirées d'un ensemble de pitches spécifiques

La première classe crée des effets de timbres et d'harmonies complexes, quand la bande de fréquence est suffisamment large (plusieurs demi-tons) et la densité élevée. Tandis que l'autre classe sert à avoir un meilleur contrôle sur le pitch.

IV.2.5 Effets de la densité

Définir la densité en termes de nombre de grains par seconde perd de la signification dès lors que l'on fait varier la taille des grains au cours du temps, c'est pourquoi Roads propose trois différents niveaux de densité :

- clairsemé : plus de 50% de la durée du nuage est constituée de silence
- plein : le nuage est rempli par des grains soniques
- dense : le nuage est caractérisé par un haut niveau de recouvrement de grains

Les relations entre densité et durées ne sont pas catégoriques. En AGS les dates de déclenchement des grains sont aléatoires, on ne peut donc garantir pour une durée de grain et une densité données si il y aura toujours recouvrement ou des périodes de silence. Les périodes de silence ne sont parfois pas perçues comme telle. Les trous dans un nuage sont plutôt perçus comme des moments de fluctuations d'amplitude.
Roads énonce quelques procédés relatifs à la densité. Par exemple, pour avoir un nuage plein une bonne règle est de régler une densité par seconde d'au moins 2/Dg (Dg = durée d'un grain). Pour obtenir un effet pointilliste, où chaque grain est entendu comme un événement distinct, garder une densité inférieure à 0.5/Dg.
La relation de la densité avec la largeur de bande (échelle de fréquences autorisée pour les grains) est très importante. En augmentant la densité, on intensifie le son, créant des effets dépendant de la largeur de bande :

- des bandes étroites avec de grandes densités créent des flux de pitch avec des spectres à formants, comme ceux de QSGS (Quasi-Synchronous Granular Synthesis) (Annexe 7)
- des bandes moyennes (de l'ordre de plusieurs demi-tons) et des grandes densités génèrent des bruits gonflés et colorés
- des bandes larges (de l'ordre d'une octave) et de grandes densités forment des nuages massifs

IV.2.6 Effets d'espace

On peut distribuer les différents grains sur plusieurs localisations spatiales. D'un point de vue psychoacoustique, la perception de la position spatiale d'un grain ou d'une série de grains dépend à la fois des propriétés physiques du signal et le fait qu'une source ponctuelle crée une image auditive étalée sur une région de l'espace. Roads propose deux manières de contrôler la spatialisation des grains :

- une enveloppe d'amplitude pour chaque canaux
- une distribution aléatoire des grains à travers n canaux.

IV.2.7 La synthèse granulaire à partir de fichiers audio

Un nouveau paramètre intervient qui est l'endroit dans l'échantillon où débute le grain. Cette technique permet des modifications de pitch ou de durée d'échantillons reprenant le principe de la machine de Gabor dont nous avons parlé avec en plus un bon contrôle sur la phase et des optimisations relatives aux fenêtres temporelles utilisées. Elle peut permettre de créer des sons hybrides en utilisant plusieurs sources sonores. Cette idée nécessite un contrôle des paramètres de grains précis pour avoir un résultat convaincant. Selon les sources sonores (largeur de bandes, articulations), la durée des grains est limitée si on veut garder une bonne représentation de la source originale. Par exemple pour de la voix, une relativement longue durée de grain est appropriée pour travailler (plus de 50 ms).

Roads précise bien que ses « postulats » relèvent plus de la curiosité scientifique que de la création sonore à proprement parler. La tâche consistant à générer des sons musicalement intéressant se fait plus à l'aide d'intuition et de goût. Néanmoins ses indications sont très utiles pour quiconque veut contrôler la génération d'un flux important de grains sonores.

IV.3 Différents programme de synthèse granulaire

- ***Csound*** : Macintosh, PC, Linux …

Plusieurs instruments Csound pour la synthèse granulaire ont été réalisés (31).

- ***Audio Mulch*** : PC

Ce programme est très facile à utiliser, il a une interface Macintosh. Il permet à l'utilisateur d'insérer des fichiers wave ou de créer des sons de la sinusoïde au bruit. Démonstration gratuite.

- ***Granulab*** : PC

Ce programme crée des grains dont on laisse définir par l'utilisateur la longueur, le pitch, l'enveloppe, l'amplitude ainsi que d'autres paramètres. Il est facile d'utilisation mais pas très stable. Il tourne en temps-réel. Démonstration gratuite.

- ***Grain Maker*** : Macintosh

Ce programme utilise une interface MAX. Il aide à créer des fichiers orchestre et des fichiers score (Annexe 1) pour Csound, il n'est pas en temps-réel.

- ***Cloud Generator*** : Macintosh

Ce programme ne laisse pas le choix de l'enveloppe à l'utilisateur mais permet de manipuler plusieurs autres paramètres. Il n'est pas en temps-réel. Il a été écrit par C. Roads, il est gratuit.

- ***Oversyte*** : Macintosh

Ecrit par R. Bencina. Programme de synthèse granulaire en temps-réel.

V Ma participation au projet GMU

V.1 Contrôle direct dans MAX

Le but de mon travail est de fabriquer des outils de contrôle pour un objet MAX *LKpgran~* implémenté en C++ qui génère des grains à enveloppe linéaire à partir de fichiers audios.
Les paramètres à contrôler sont :

- *begin* (en millisecondes) : début du grain, c'est à dire l'emplacement temporel dans l'échantillon où débute le grain.
- *transpose* : variation du pitch du grain
- *amplitude* (de 0 à 1) : amplitude du grain.
- *panoramic* (de 0 à 1) : répartition stéréo, 0 équivaut à envoyer le grain sur le canal gauche, 1 sur le canal droit.
- *attack* (en millisecondes) : temps d'attaque de l'enveloppe d'amplitude.
- *steady state* (en millisecondes) : temps de maintien.
- *release* (en millisecondes) : temps de décroissance.

L'idée principale de contrôle est de générer des valeurs aléatoires de paramètres comprises entre des bornes qui évoluent au cours du temps (IV.1.4 et IV.2).

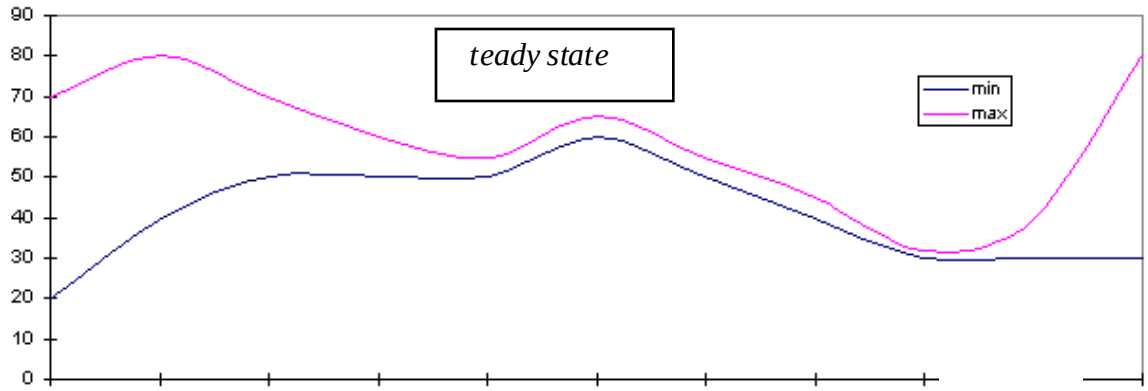


fig.1

L'objet MAX *drivegran* que j'ai développé suit ce principe. Il permet, pour chacun des paramètres de synthèse, de contrôler une borne inférieure, une borne supérieure, ainsi qu'un écart maximal entre deux valeurs successives. Le contrôle se fait par des *sliders* ou par l'intermédiaire des boîtes à nombres, le but étant que les informations communiquent entre ces deux types de contrôle : modifier un slider modifie la boîte à nombre qui lui correspond et vice-versa. *Drivegran* permet également un contrôle direct des valeurs de paramètres. Le choix du contrôle se fait en activant ou désactivant des interrupteurs.
Voici l'aspect général du synthétiseur, je détaillerai ensuite les divers éléments qui le composent :

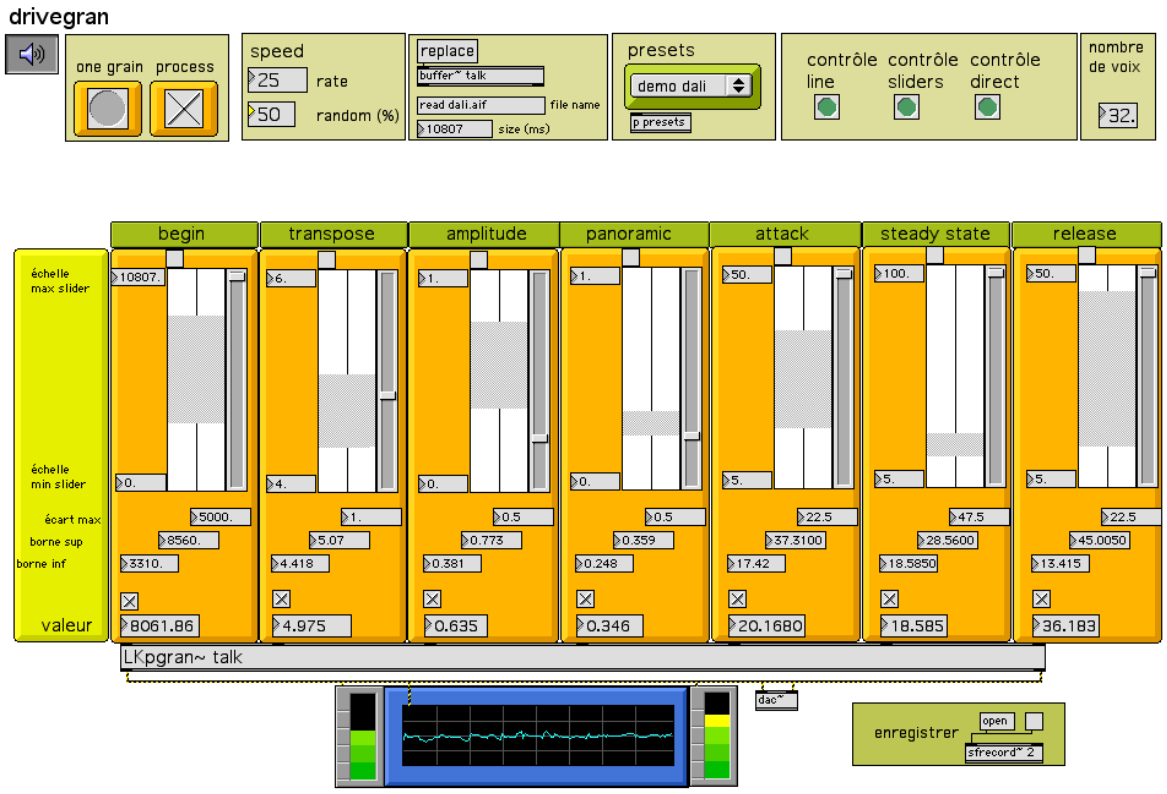


fig.2

Chaque paramètre est contrôlé par un ensemble de modules compactés dans une tranche qui permettent les utilisations que j'ai décrites plus haut. Voici ce que contient chacune de ces tranches (ici celle de *begin*):

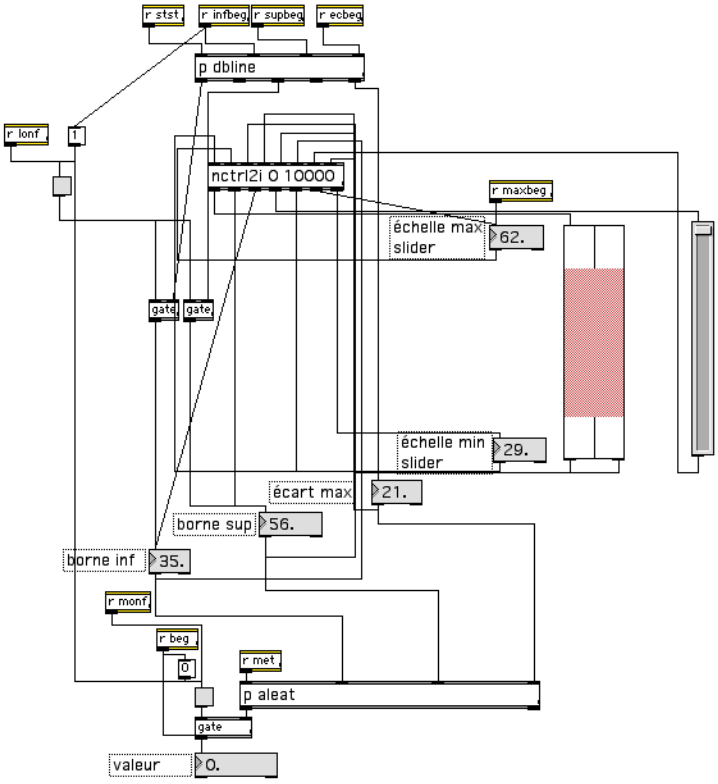


fig.3

On retrouve bien :

- le *slider* indiquant la borne supérieure et la borne inférieure (la zone sélectionnée représente la zone allant de borne inf. à borne sup.)
- le *slider* indiquant l'écart maximal entre deux valeurs successives
- la boîte à nombre pour la borne supérieure
- la boîte à nombre pour la borne inférieure
- la boîte à nombre pour l'écart maximal
- la boîte à nombre pour la valeur maximum du premier *slider*
- la boîte à nombre pour la valeur minimum du premier *slider*
- la boîte à nombre pour la valeur du paramètre

V.1.1 L'objet *nctrl2i*

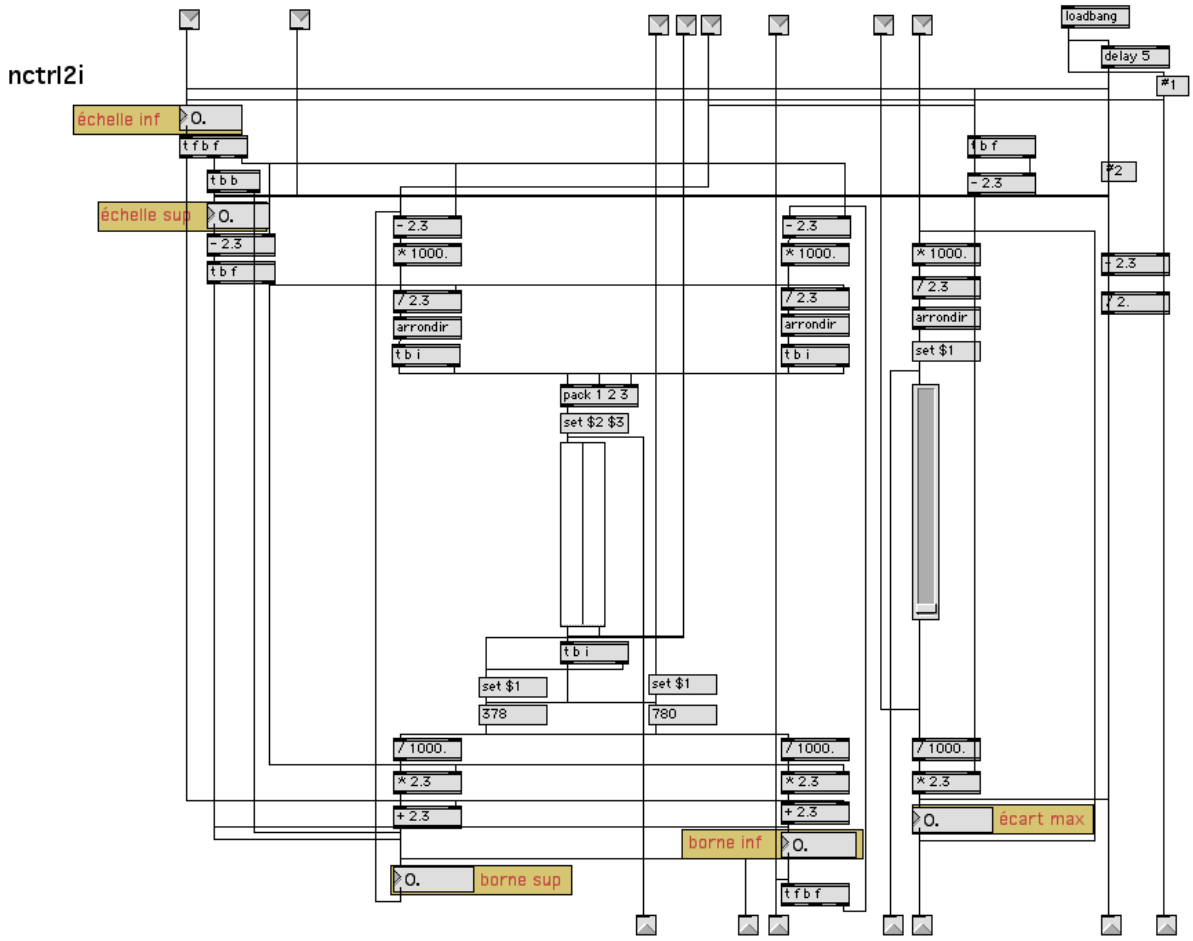


fig.4

C'est cet objet qui permet de faire dialoguer les informations entre les *sliders* et les boîtes à nombre. L'utilisation du message *set* y est précieuse pour éviter que les données partent dans des boucles infinies. L'échelle du *slider* représentant l'écart maximal va de 0 à (borne sup. – borne inf.) et s'ajuste dès que l'une des deux bornes varie. Dans la figure 4, la borne supérieure est à 56, la borne inférieure à 35 et on voit que la valeur maximale de « écart max » est à 21.

Dans MAX on ne peut utiliser que des valeurs entières avec les *sliders*, d'où l'utilisation de divisions par 100 entre *sliders* et boîtes à nombres et de multiplications par 1000 et d'un objet *arrondir* (figure 5) entre boîtes à nombres et *sliders*. Cela permet de manipuler des valeurs flottantes, utiles pour les paramètres *panoramic, amplitude et transpose.*

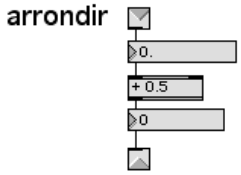


fig.5

nctrl2i prend deux arguments optionnels servant à initialiser l'échelle du *slider* des bornes supérieure et inférieure.

V.1.2 Synchronicité

Comme nous l'avons vu dans IV.2, la principale différentiation entre les techniques de synthèse granulaire se fait par le caractère synchrone ou asynchrone du déclenchement des grains.
Dans *drivegran* , c'est un objet *jfmetro2* qui gère le déclenchement des grains :

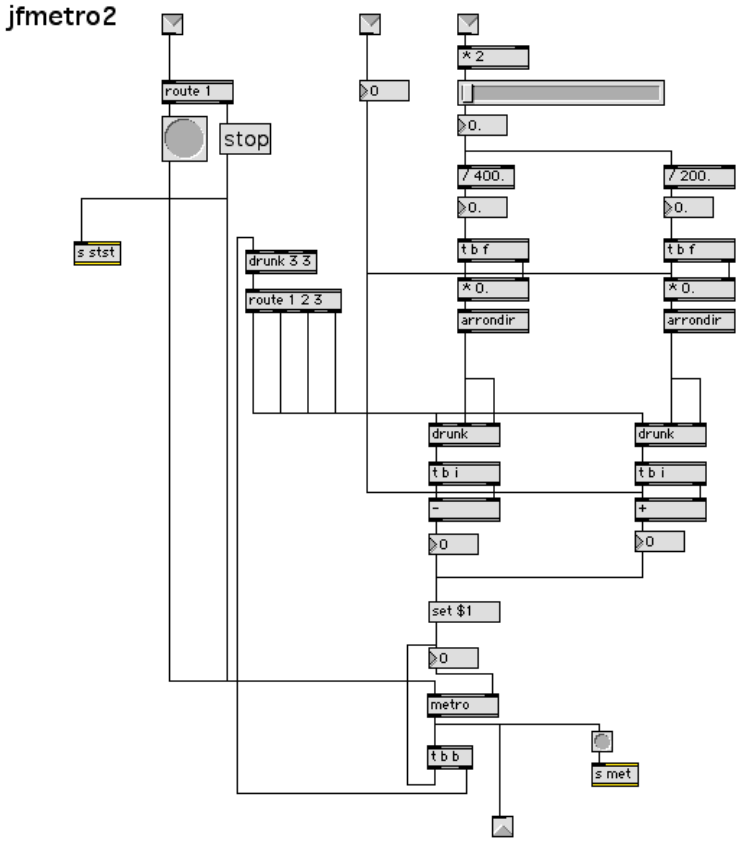


fig.6

Cet objet est construit sur la base d'un *metro* dont on fait varier la vitesse de déclenchement. Dans *drivegran*, on règle le pourcentage de variation aléatoire pour la vitesse de déclenchement, à 100 % le délai entre deux déclenchements varie entre la moitié et le double de la valeur centrale (*rate*). Par exemple pour un *rate* de 100 ms, les délais entre deux bangs successifs varie entre 50 ms et 200 ms pour un *random* de 100. On peut ainsi passer progressivement d'un déclenchement synchrone de grains à un déclenchement asynchrone.
Le principe de *jfmetro2* est qu'à chaque bang émis par le *metro*, on déclenche un processus alèatoire (*drunk*) autour de la valeur centrale qui va nous fournir une valeur que l'on réinjecte comme argument du *metro*.

J'ai également proposé un objet *jfseq* qui permet des déclenchements de grains suivant un pattern rythmique précis que l'utilisateur programme lui même.

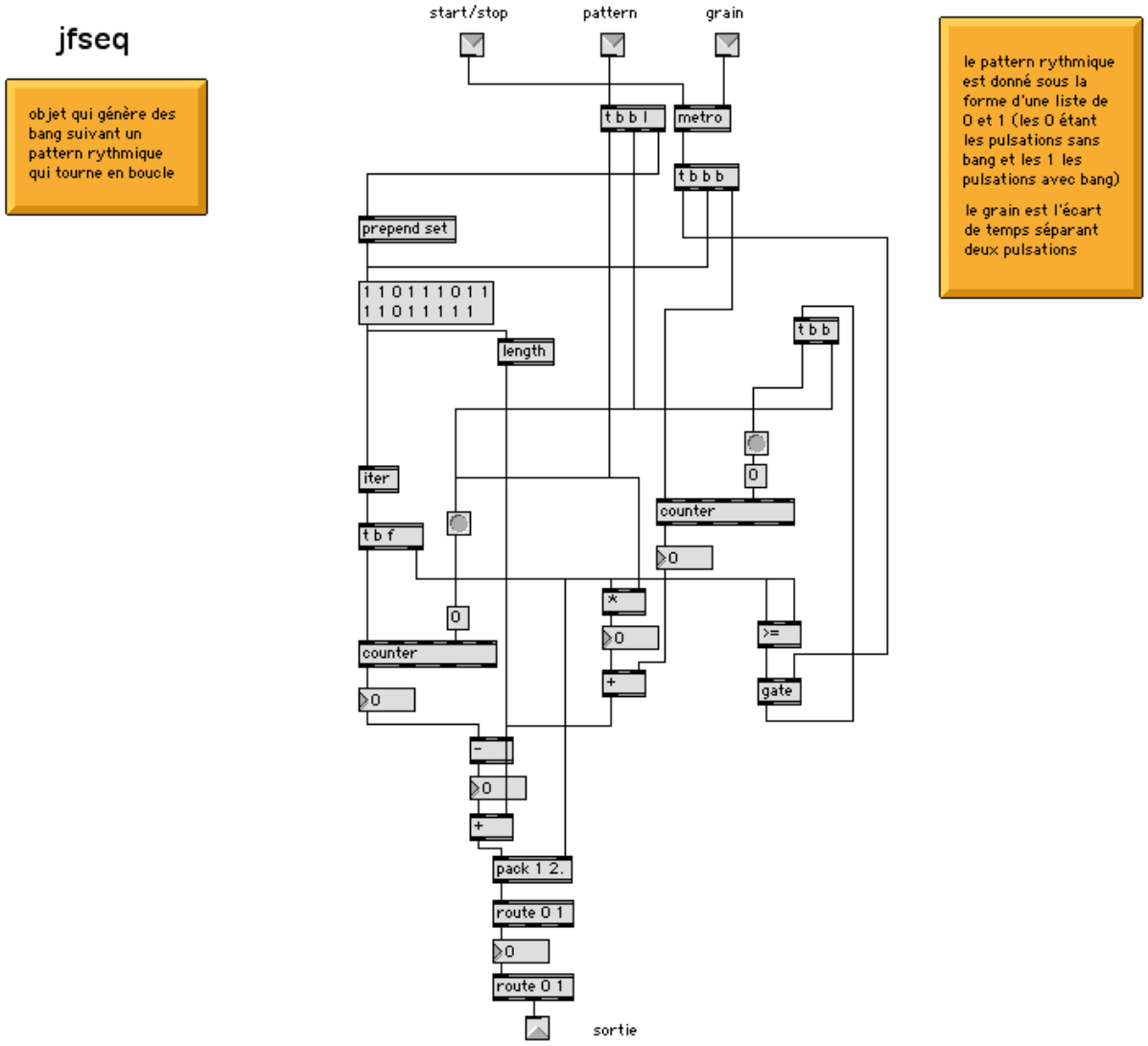


fig.7

Cet objet est un « séquenceur de *bang* », il reçoit en entrée un pattern rythmique sous forme d'une liste de 0 et 1, les 1 représentant les bang émis et les 0 les bang non-émis, ainsi qu'une période qui est donnée en nombre de millisecondes séparant deux bangs successifs (norme de l'objet *metro*). Le principe de l'objet est d'envoyer à chaque bang fourni par le *metro* la liste de 0 et 1 dans un *iter* puis dans l'entrée de droite d'un *pack* ; l'entrée de gauche du *pack* est régie par un système de compteurs. Le *pack* numérote donc les éléments de la liste, il suffit ensuite de lire le bon indice pour avoir l'élément voulu, c'est à ça que serve les compteurs. Grâce à un *route* juste avant la sortie, on filtre les éléments en ne faisant sortir que les 1. A la fin du pattern, on initialise les compteurs et le processus recommence. En sortie, on a une séquence rythmique composée de bang, qui tourne en boucle. Si on donne comme entrée le pattern (1 1), on aura une pulsation régulière. Le but de cet objet est d'introduire, si on le désire, une irrégularité contrôlée dans la génération des grains (cet objet ne se trouve pas dans *drivegran* mais peut facilement être rajouté).

V.1.3 Calcul des valeurs stochastiques : le sous patch *p aleat*

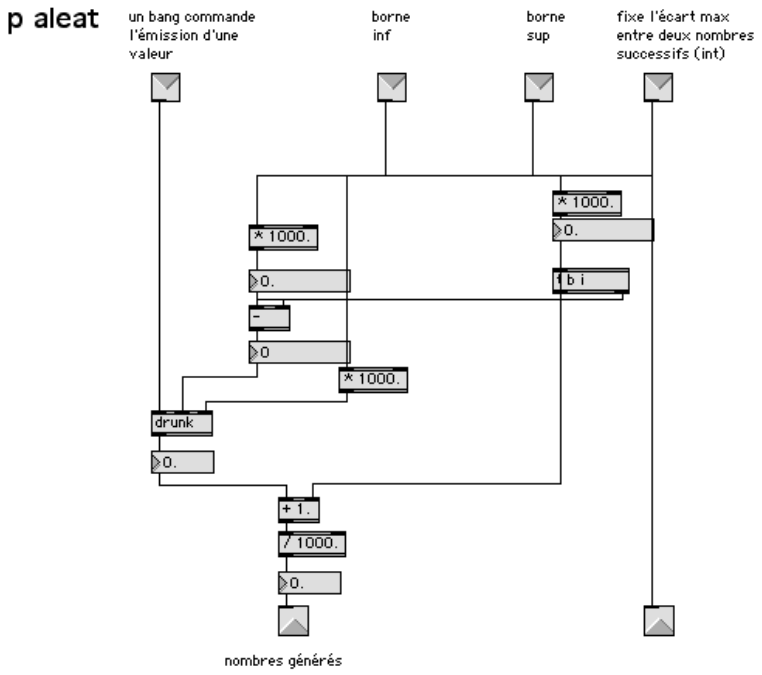


fig.8

Ce sous-patch sert à générer des valeurs aléatoires comprises entre la borne inférieure et la borne supérieure. Pour cela j'ai utilisé l'objet *drunk* qui fournit des valeurs aléatoires entre 0 et une valeur d'entrée. La valeur d'entrée est la différence entre les deux bornes. A la sortie du

drunk, il faut ajouter la valeur de la borne inférieure à la valeur aléatoire pour que cette dernière soit effectivement comprise entre les deux bornes. Les multiplications et divisions par 1000 qu'on voit dans ce sous-patch sont là pour permettre d'avoir des valeurs flottantes car *drunk* a le même défaut que les *sliders*, c'est à dire qu'il ne prend et ne génère que des valeurs entières.
Le sous-patch *p aleat* possède quatre entrées :

- Une entrée qui commande l'émission des valeurs aléatoires.
- Une entrée pour la borne supérieure
- Une entrée pour la borne inférieure
- Une entrée qui fixe l'écart maximum entre deux valeurs aléatoires successives (fonction de l'objet *drunk* que l'on a gardée).

V.2 Contrôle par messages dans MAX

Dans la figure 3 qui détaille le contrôle des paramètres, on voit un sous-patch nommé *p dbline* en amont de l'objet *nctrl2i*

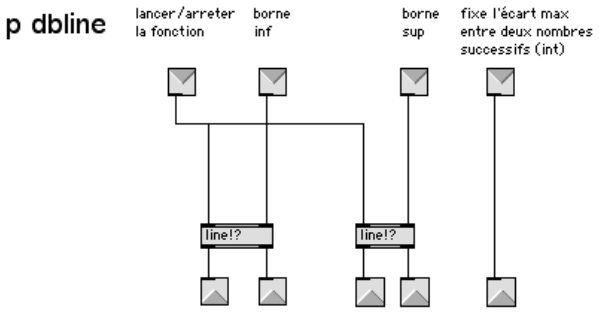


fig.9

Ce sous-patch sert à générer des valeurs évoluant au cours du temps pour les bornes supérieure et inférieure, il utilise pour cela l'objet *line !?*.

L'objet *line!?* génère des enveloppes à segments, chose qui n'existe pas actuellement dans MAX : *line* génère un seul segment et *line~* génère des segments de signaux.
Nous avons donc construit *line!?* à partir de *line*.
La fonction est donnée de la même manière que pour l'utilisation de *line~*, c'est à dire :
(valeur initiale, valeur1 temps1 valeur2 temps2 …) . Cette forme de donnée est interprétée comme 2 listes ; une liste de 1 élément (valeur initiale) et une liste d'au moins 2 éléments (valeur1 temps1 valeur2 temps2 …) :

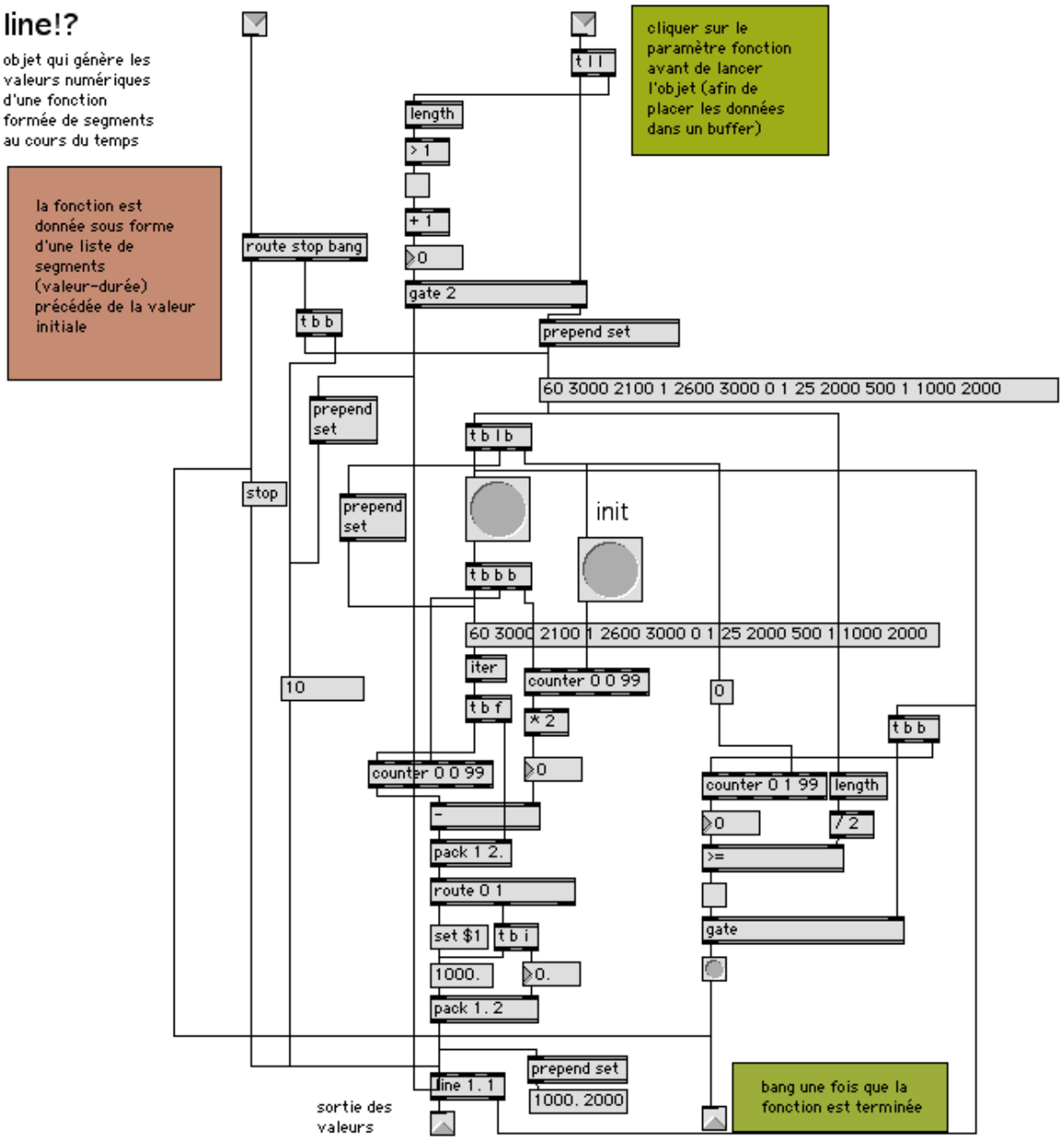


fig.10

Selon leur longueur évaluée par *length* , < 1 ou > 1, les listes sont stockées dans deux buffers différents, on peut ainsi séparer la valeur initiale qui, lors du déclenchement du patch, sera envoyée directement à *line* alors que le reste des données subira un traitement spécial : grâce à un *trigger*, le *bang* de déclenchement effectue plusieurs opérations dans un ordre imposé :

- initialise un premier compteur.

- stocke la liste (valeur1 temps1 valeur2 temps2 …) dans un buffer.
- incrémente ce premier compteur.
- initialise un deuxième compteur.
- envoie la liste dans *iter*.

iter va envoyer les éléments de la liste un par un dans *pack* qui reçoit également la sortie du deuxième compteur incrémenté par chaque bang généré par les éléments de la liste si bien qu'on obtient à la sortie de pack une liste de la forme :
0 valeur1
1 temps1
2 valeur2
3 temps2
…

Ensuite un *route* va sélectionner uniquement les deux éléments correspondant à 0 et 1 puis les envoyer à *line*. Ainsi *line* reçoit des données qu'il comprend, à savoir une liste d'un élément qui est la valeur initiale suivie d'une liste de deux éléments qui sont la valeur à atteindre et la durée que cela prendra.

Une fois que le segment est terminé, *line* envoie un *bang* que l'on récupère pour recommencer le même processus à ceci près qu'on incrémente le premier compteur et qu'on l'utilise pour transformer les données issues du deuxième compteur ; on retranche 2 à toutes ses valeurs. Donc la numérotation des éléments à la sortie du *pack* est décalée et *line* lira alors les valeurs 3 et 4 de la liste initiale c'est à dire valeur2 et temps2. Le processus continue ainsi pour chaque nouveau segment.
Processus d'arrêt : à chaque bang de fin de segment, on incrémente un compteur qui une fois qu'il atteint la longueur de la liste divisée par 2 (c'est à dire le nombre total de segments) déclenche un stop relié à *line*.

line!? possède deux entrées :
- Une entrée qui reçoit les messages de déclenchement (*bang*) et d'arrêt (*stop*) car il faut bien sur pouvoir arrêter la fonction en cours d'exécution.
- Une entrée qui reçoit les données de la courbe à effectuer.

Et deux sorties :
- Une sortie qui fournit les valeurs générées.
- Une sortie qui envoie un bang quand le dernier segment se termine.

Par l'intermédiaire des objets *receive* (voir figure 3, *r supbeg* et *r infbeg* pour le paramètre *begin*), les sous patch *dbline* peuvent recevoir des listes d'arguments préprogrammées. On peut ainsi définir des préréglages dont l'utilisateur pourra se servir avec le sous-patch *p presets* (voir figure 2).

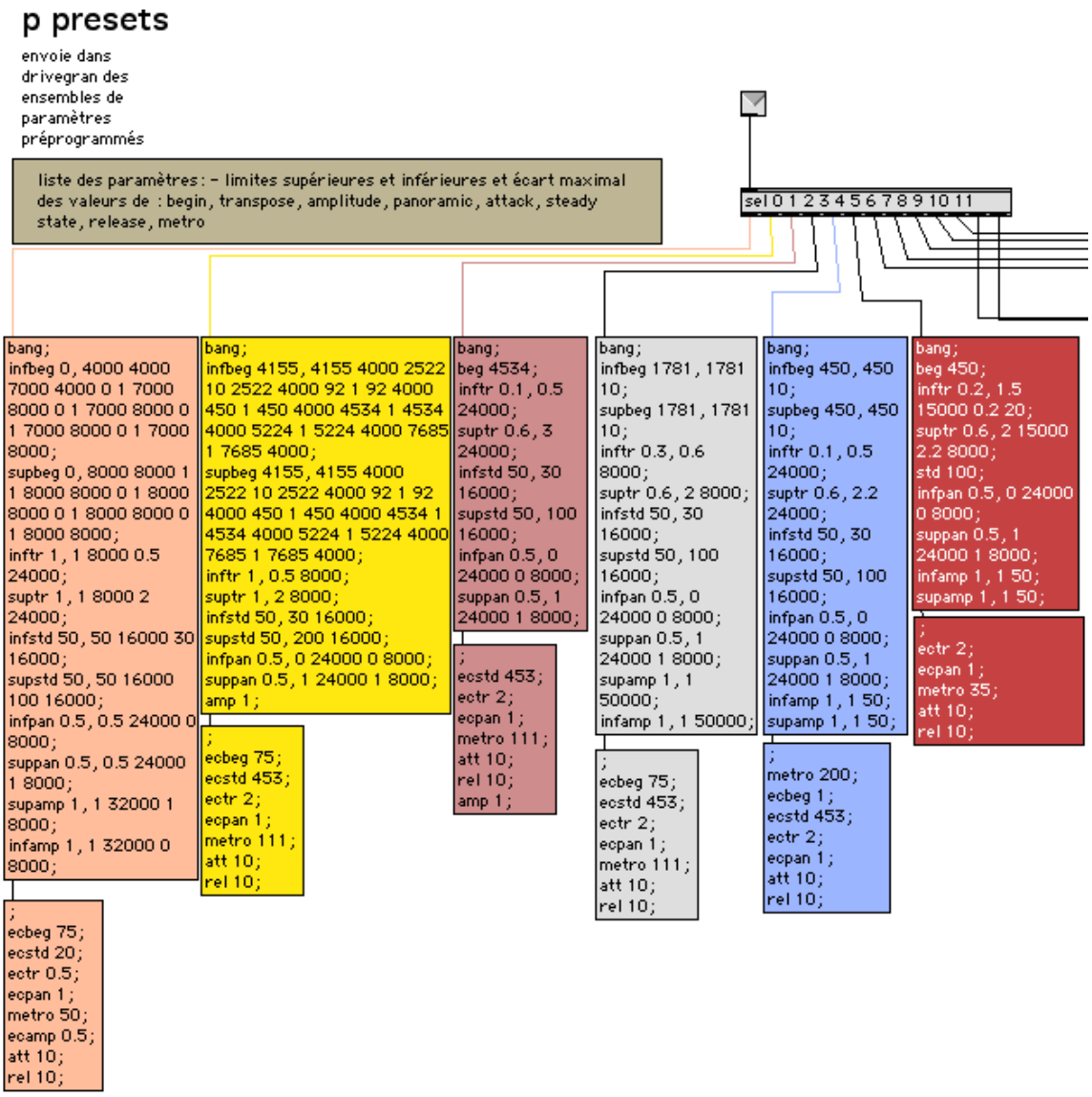


fig.11

V.3 Choix

On a vu que *drivegran* propose trois types de contrôle différents :

- un contrôle direct des valeurs de paramètres par l'intermédiaire de boîtes à nombres
- une génération stochastique de valeurs comprises entre des bornes que l'on règle soit par des sliders, soit par des boîtes à nombre
- une génération stochastique à partir de valeurs des bornes préprogrammées

Ces possibilités sont encore plus intéressantes si on peut passer avec « souplesse » de l'une à l'autre. C'est pourquoi j'ai inséré un système d'interrupteurs pour chacun des paramètres

permettant de régler individuellement le type de contrôle selon le paramètre. Il a bien sur fallu adapter ce système aux possibilités existant déjà :

- un bouton permet l'utilisation de tous les paramètres en contrôle direct, c'est à dire envoi d'un 0 aux sept interrupteurs du bas par *r monf* (voir figure 3)
- un bouton permet l'utilisation de tous les paramètres en contrôle stochastique manuel, donc envoi d'un 1 aux sept interrupteurs du bas et d'un 0 aux sept interrupteurs du haut par *r lonf.*
- Lorsqu'un utilise les *presets*, ils envoient des informations aux interrupteurs afin que la configuration logique du patch soit prête à recevoir leurs informations. Pour un paramètre donné, par exemple *begin* (figure 3), l'utilisation du contrôle par *dbline* envoie un 1 à l'interrupteur du haut ainsi qu'à celui du bas par l'intermédiaire de *r infbeg*. Si le preset contient une valeur fixe pour ce paramètre, elle sera transmise à la boîte à nombre de la valeur par l'intermèdiaire de *r beg* qui enverra également un 0 à l'interrupteur du bas, coupant ainsi tout contrôle stochastique.

V.4 Création de messages MAX dans OpenMusic

V.4.1 Adaptation des données

J'ai créé dans OM un ensemble d'objets qui permettent la génération de fichiers textes qui, lus dans MAX, créent des fenêtres contenant des messages du type preset (figure 11). L'utilité de cet ensemble d'objets est de permettre la génération de paramètres stochastiques essentiellement à partir d'entrées graphiques.
Les fichiers textes MAX que nous voulons élaborer ont la forme :

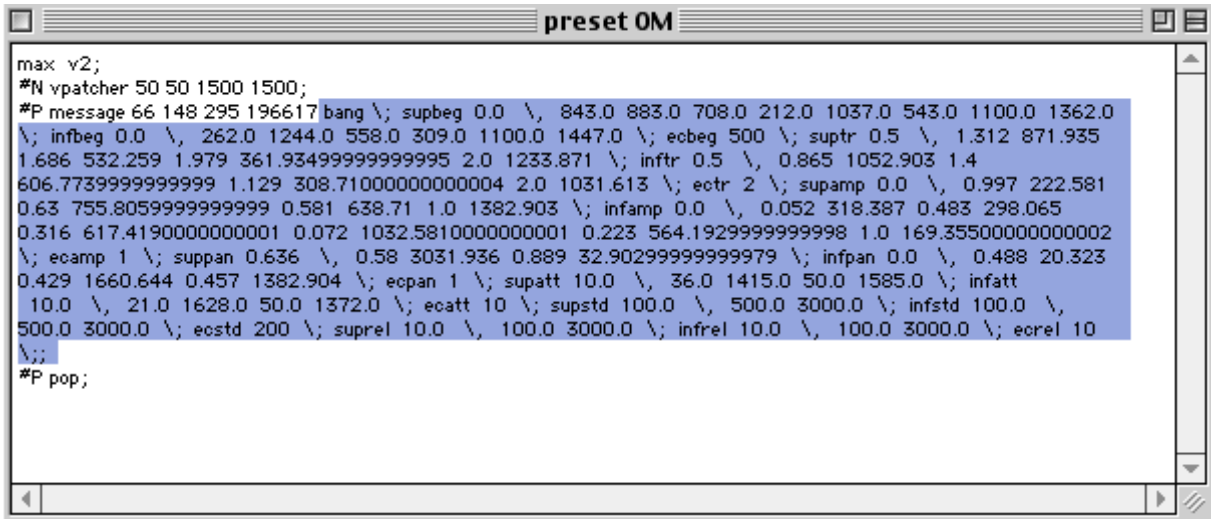


fig.12

Les caractères qui ne sont pas surlignés représentent des informations relatives à la création d'une fenêtre MAX contenant une boîte « message » et aux dimensions des éléments. Ce sont des caractères que l'on gardera identiques pour tous les fichiers que nous créerons dans OM. La partie surlignée représente les informations contenues dans le message.
Voici le résultat de ce fichier texte lu dans MAX :

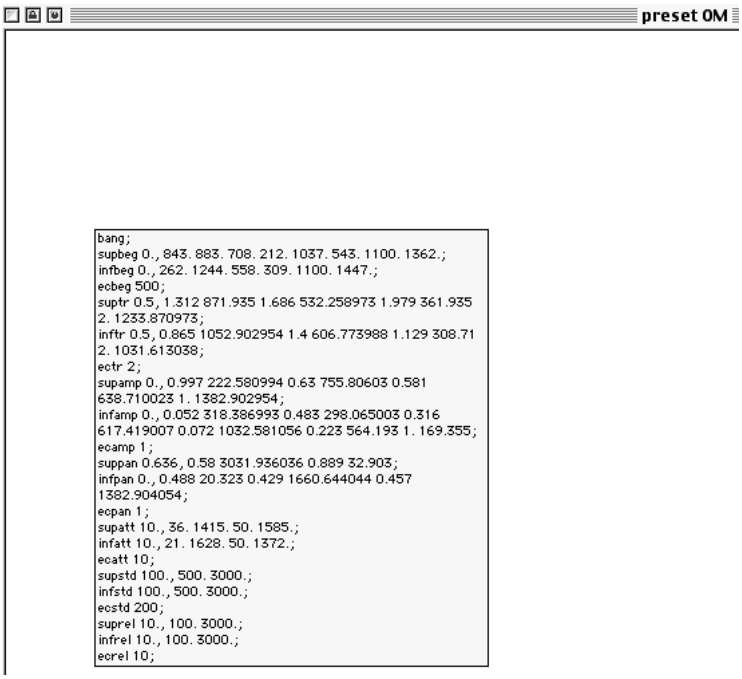


fig.13

Nous avons donc construit un objet *bpf->linseg* (figure 14) dans OM qui transforme les données d'une BPF-LIB (éditeur graphique multi-courbe) en données du type *(valeur initiale, valeur1 temps1 valeur2 temps2 …)* qui sont interprétables par *line !?:*

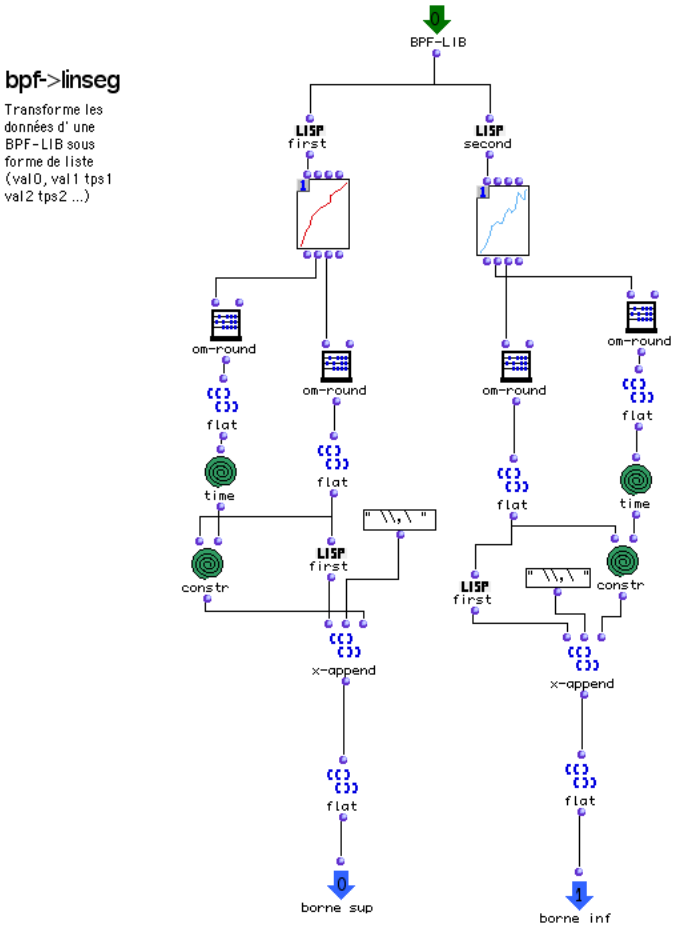


fig.14

Cet objet sépare les deux courbes (borne supérieure et borne inférieure), prend les données d'abscisses et d'ordonnées de chaque point, puis les manipule de manière à obtenir la forme souhaitée.

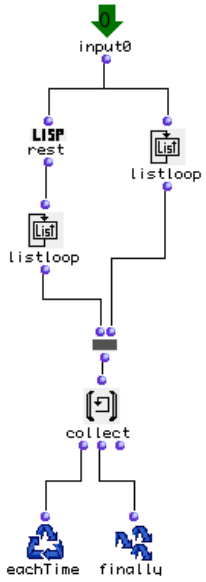


fig.15

Les données d'abscisse, représentant le temps, doivent être réexprimées en valeurs temporelles séparant chaque date et non pas comme une liste de dates comme c'est le cas à la sortie d'une BPF.
On envoie donc ces données dans une boucle appelée *time* (figure 15) qui va soustraire à la liste de dates cette même liste privée de son « car » :

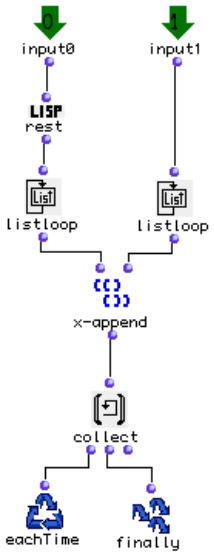


fig.16

Il faut ensuite alterner les valeurs d'ordonnée et les valeurs de durée obtenues avec la boucle *time*. Pour ce faire, on envoie les deux listes dans une boucle *construct* (figure 16) qui va créer une liste du type (*valeur1 temps1 valeur2 temps2 …)*.

La valeur initiale, suivie d'une virgule, est ensuite insérée au début cette liste et on obtient la formulation désirée pour les données.

V.4.2 Choix des paramètres

Le but est de fabriquer la partie surlignée de la figure 12. Dans cette figure, il y des données pour les sept paramètres de synthèse. Ce fichier est créé avec le patch *maxmessage* (figure 17) :

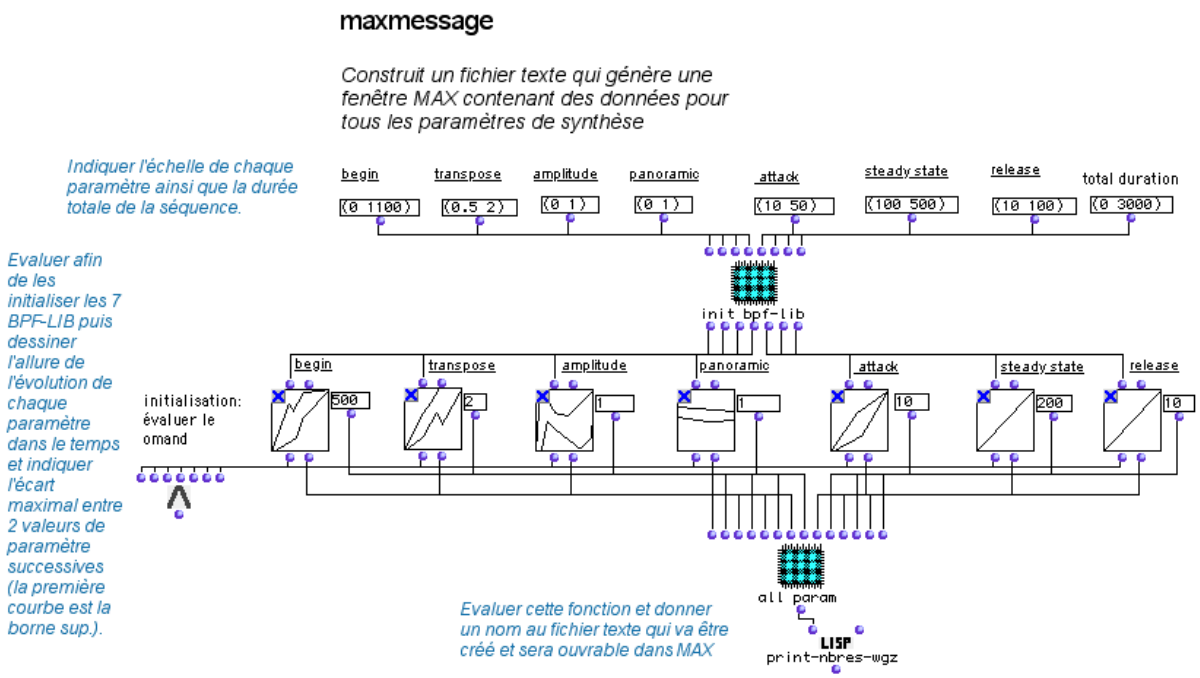


fig.17

Ce patch contient deux sous-patch. Le premier, *init bpf-lib* (figure 18), sert à initialiser les sept BPF-LIB dans lesquelles l'utilisateur dessinera ses courbes.

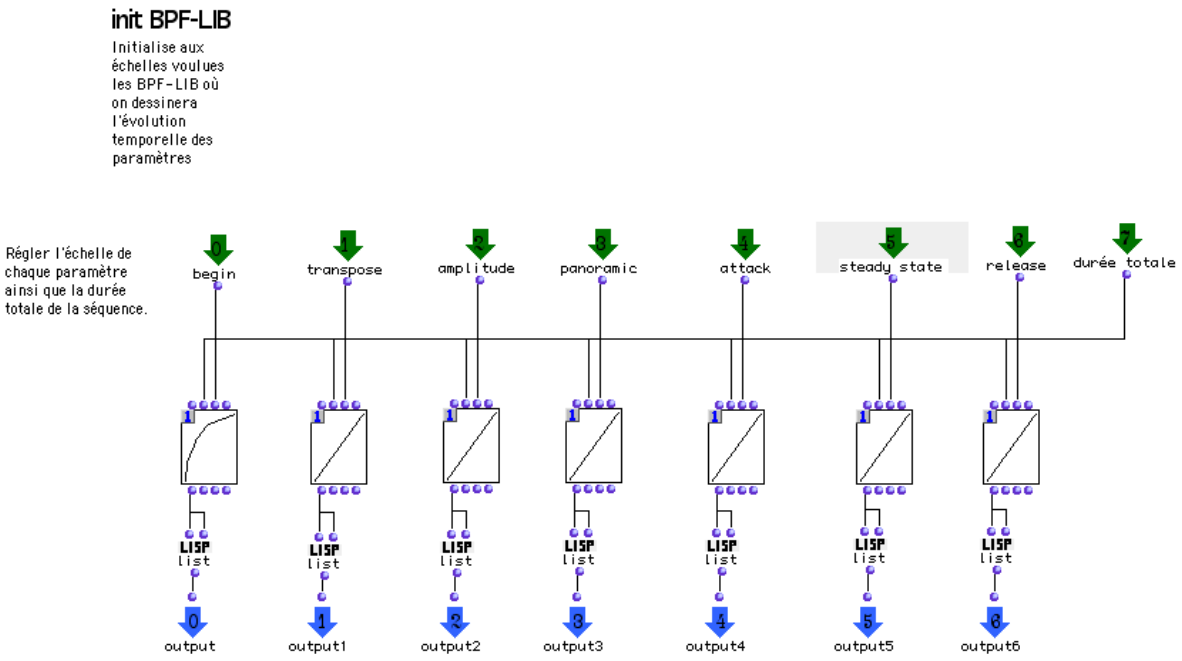


fig.18

On rentre les échelles souhaitées pour chaque paramètre ainsi que la durée totale des courbes puis on évalue les sept BPF-LIB. Ensuite on bloque (position croix des boutons des BPF-LIB) de manière à ne pas réinitialiser lors de l'évaluation finale, puis on dessine l'évolution temporelle et on indique l'écart maximum entre deux valeurs successives pour chaque paramètre. Toutes ces données vont ensuite dans le deuxième sous-patch *all parameters* (figure 19), qui sert à arranger la syntaxe des données de manière à obtenir la forme de la partie surlignée dans la figure 12 :

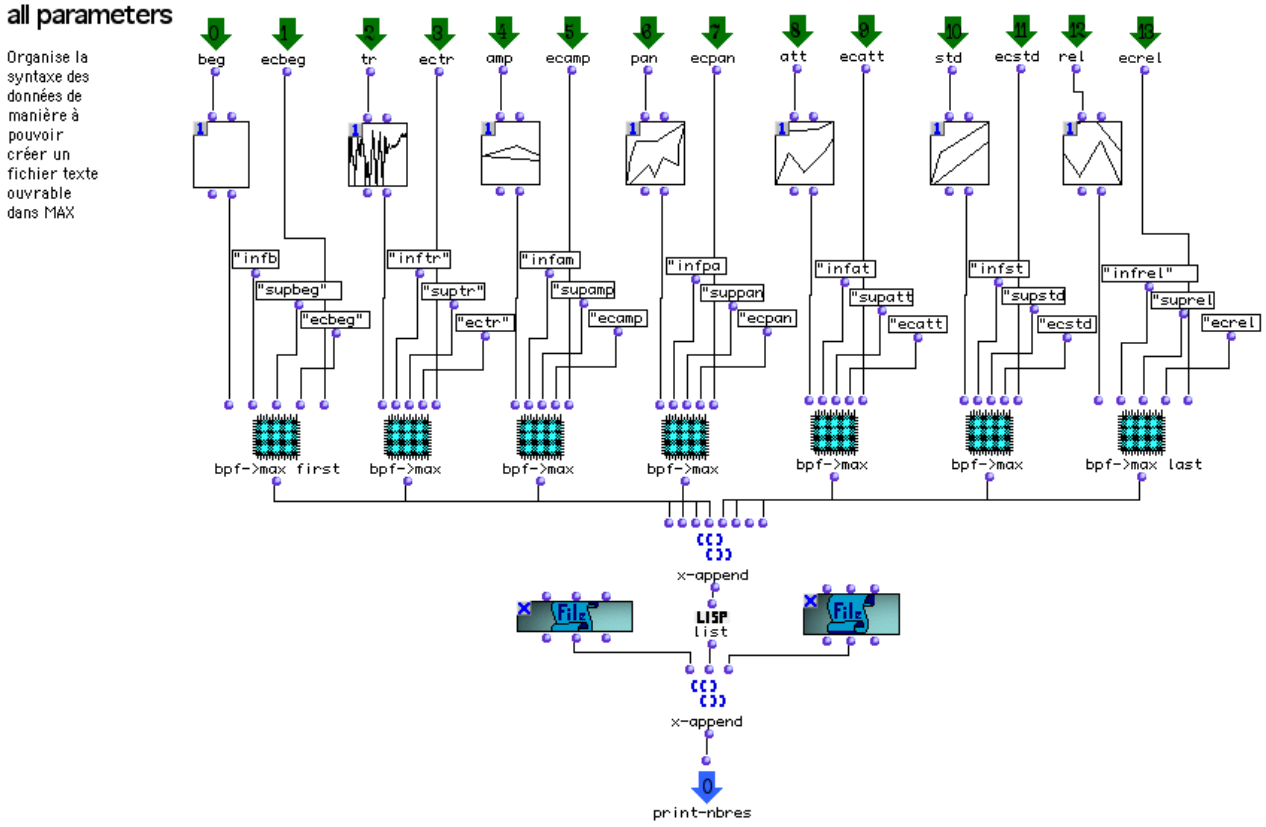


fig.19

Les sous patch *bpf->max first, bpf->max* et *bpf->max last* sont également destinés à la syntaxe. Ils prennent en entrée le nom des paramètres, les sorties BPF et les valeurs d'écart maximum :

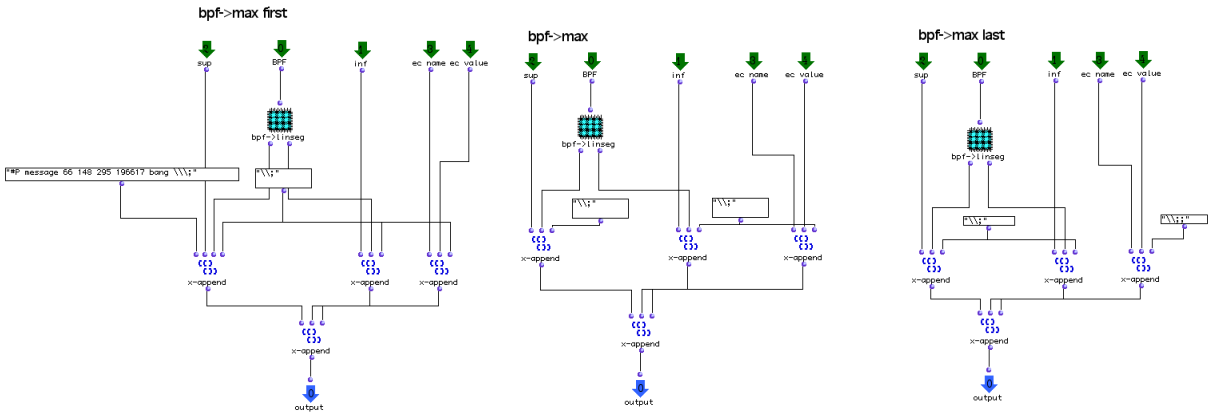


fig.20

Ce sont eux qui contiennent les *bpf->linseg* (figure 14).
Enfin, on évalue la fonction *print-nbres* et on obtient un fichier texte du type de celui de la figure 12. La fonction *print-nbres* appartient à la librairie OM LP-Max créée par L.Pottier et sert à créer de tels fichiers. *Maxmessage* permet de générer tous les paramètres à la fois mais pour un contrôle plus fin des paramètres, j'ai créé des objets permettant de générer séparément les données de chaque paramètre du synthétiseur granulaire. Ceci conserve l'esprit de modularité de *drivegran* où on peut gérer indépendamment chaque paramètre. Ainsi les objets OM *beg* (figure 21), *tr*, *pan*, *amp*, *att*, *std*, *rel*, contenant l'objet *single parameter* (figure 22) génère des fenêtres MAX pour un seul paramètre de synthèse.

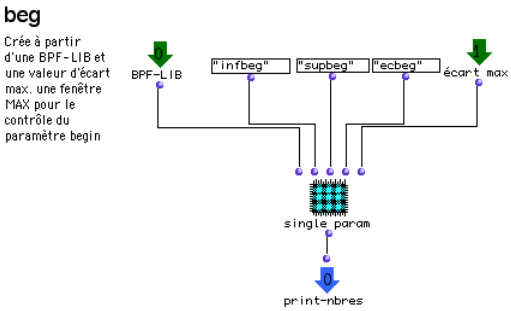


Fig.21

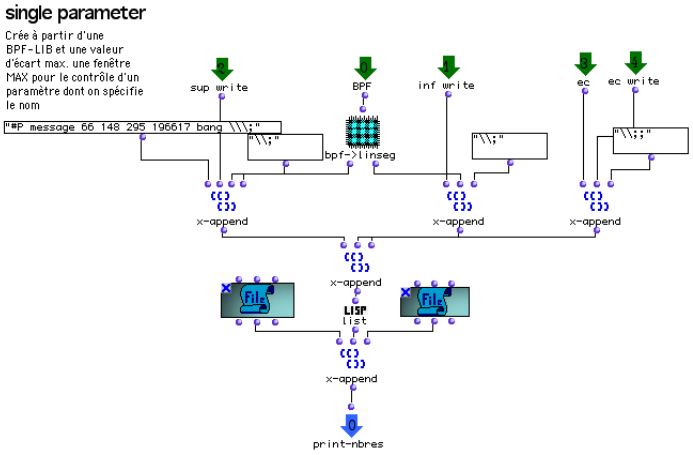


Fig22

V.4.3 Graphisme

Dans *maxmessage*, l'utilisateur dessine ses courbes dans des BPF-LIB qu'il a préalablement initialisées. Par contre on peut se servir de n'importe quelles données pour générer les BPF-LIB que l'on connectera aux objets *beg*, *tr*, …
Voici un exemple de patch fait à partir de l'objet *beg* (figure 23):

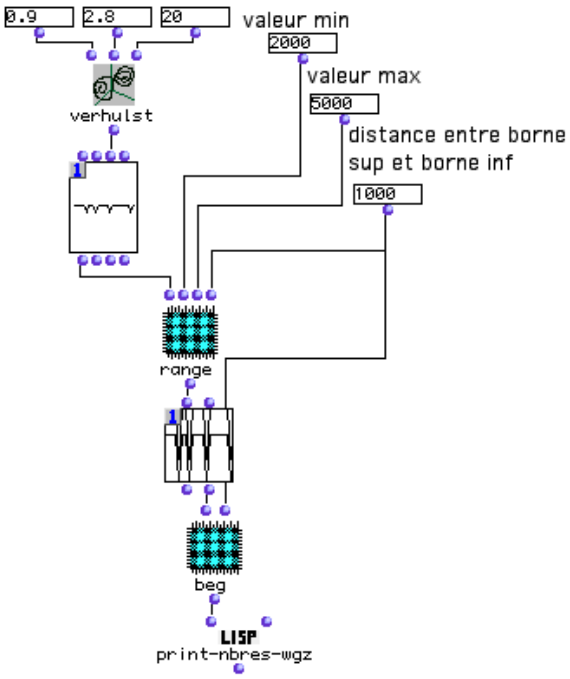


fig23

L'équation de *Verhulst* est inspirée des idées de *Quetelet* qui pensait que des « forces » intervenaient dans l'évolution d'une population biologique et que ces forces dépendaient du taux de croissance de la population. En approfondissant cette théorie *P.F. Verhulst* a déduit une équation non-linéaire décrivant l'évolution d'une population biologique. (il prédit d'ailleurs vers 1850 une limite de la population belge à 9.400.000). La fonction *verhulst* de la librairie OMChaos est basée sur cette équation et prend comme arguments une valeur initiale représentant le taux par rapport à la population maximum, un coefficient de perturbation et un nombre d'itérations.
Les données de sortie sont transférées dans une BPF, le temps en abscisse et les valeurs de la fonction en ordonnée.
Ces valeurs sont un taux, donc comprises entre 0 et 1. C'est pourquoi on utilise un objet appelé *range* qui va fournir une BPF-LIB contenant deux courbes proportionnelles à l'originale mais dans une échelle que l'utilisateur aura choisie. On fixe également une distance constante entre les deux courbes. On envoie ensuite cette BPF-LIB dans l'objet *beg* puis on crée le fichier texte en évaluant la fonction *print-nbres*.

Voici le fichier texte OM que l'on a créé (figure 24) :

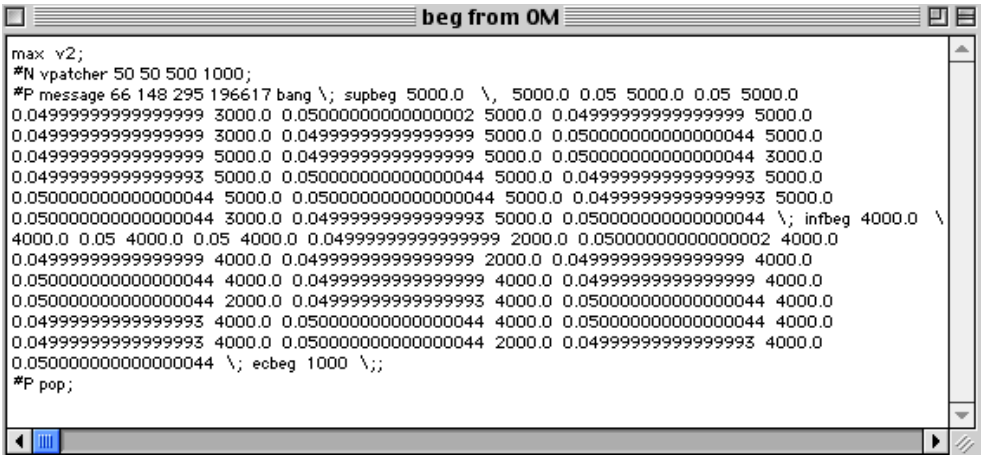


fig24

Une fois ouvert dans MAX, il devient le message suivant (figure 25), c'est à dire un message pour le contrôle du paramètre *begin* du synthétiseur granulaire. On clique sur ce message et les informations sont envoyées dans *drivegran.*

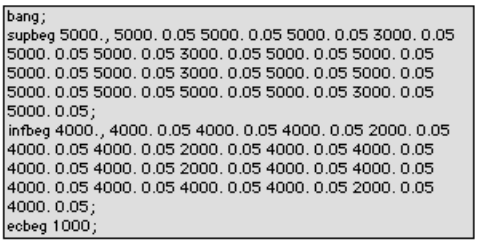


fig25

On se sert ici de *range* pour adapter les données provenant de la fonction *verhulst* à une échelle qui nous intéresse ; ici, c'est celle du paramètre *begin* qui est de l'ordre du millier de millisecondes. Cet objet sert à adapter les données de n'importe quelle fonction qui peut inspirer l'utilisateur pour le contrôle de la synthèse. La librairie OMchaos possède plusieurs fonctions relatives à des phénomènes non-linéaires ou chaotiques (*lorentz, navier-stokes, henon, stein* …) qui peuvent être utilisées pour le contrôle des paramètres grâce à *range* (figure 26).

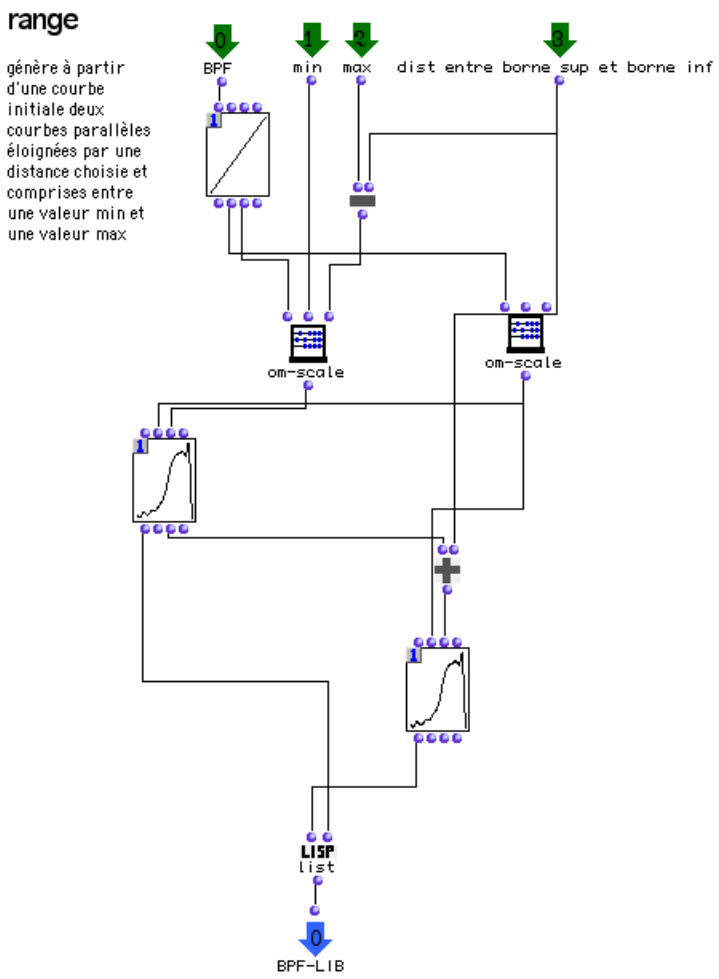


fig26

On vient de voir comment on peur créer des messages MAX à partir de données graphiques ou stochastiques de OM. Ces messages enregistrés sous forme de fichiers textes peuvent être sauvés sous forme de fichiers MAX. Ainsi, l'utilisateur peut se confectionner une bibliothèque de presets qu'il pourra utiliser dans *drivegran*. Le fait de pouvoir générer des données pour un paramètre isolé permet par la suite d'assembler des messages relatifs à des paramètres différents et d'avoir des possibilités combinatoires pour la production de presets regroupant tous les paramètres ; il suffit de coller dans un patch MAX les différents messages que l'on désire combiner et on peut créer un nouveau preset.

V.5 Transfert de données sous forme de fichiers MIDI

V.5.1 Sauvegarde dans OM sous forme de fichier MIDI

Nous avons également fabriqué des outils permettant de stocker des informations sous forme de fichiers MIDI.
L'inconvénient des messages MIDI est qu'ils sont faits pour communiquer des valeurs entières allant de 0 à 127, cela pose des problèmes de compatibilité avec les ordres de grandeurs de quasiment tous les paramètres de synthèse. C'est pourquoi on code les valeurs des paramètres sur deux ou plus « transporteurs » de données MIDI.
Par exemple, pour le paramètre *begin*, on utilise 4 messages contrôleurs différents : 2 pour la borne supérieure et 2 pour la borne inférieure. La fonction qui permet de créer ces fichiers MIDI est *write-mf* de la bibliothèque LP-Midifile créée par L. Pottier.

Grâce à ce patch (figure 27), on crée une liste de messages MIDI pour le contrôle des paramètres *begin* et *transpose*. Les messages que nous créons relatifs à des évènements MIDI sont sous forme de listes de 5 éléments :
Exemple : (0 "ctrl" 1 0 1)
Le premier élément représente la date de l'événement MIDI (en centièmes de seconde), le deuxième le type de message (ici contrôleur), le troisième le type de contrôleur, le quatrième la valeur MIDI (comprise entre 0 et 127) puis enfin le canal MIDI. Ce message signifie qu'à la date 0, on transporte la valeur 0 avec le contrôleur 1 par le canal 1.

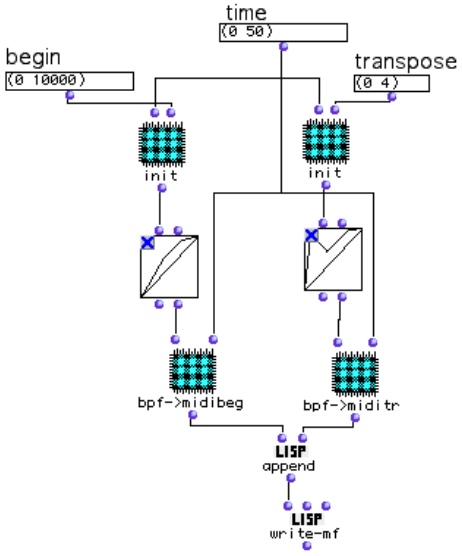


fig27

Comme pour la création des messages MAX, j'ai créé des objets permettant de générer des valeurs pour les paramètres indépendamment les uns des autres. On initialise d'abord les BPF-LIB dans laquelle on va dessiner les courbes inférieures et supérieures, puis on passe par des objet *bpf->midibeg* et *bpf->miditr* qui vont effectuer la conversion des données (figure 28).

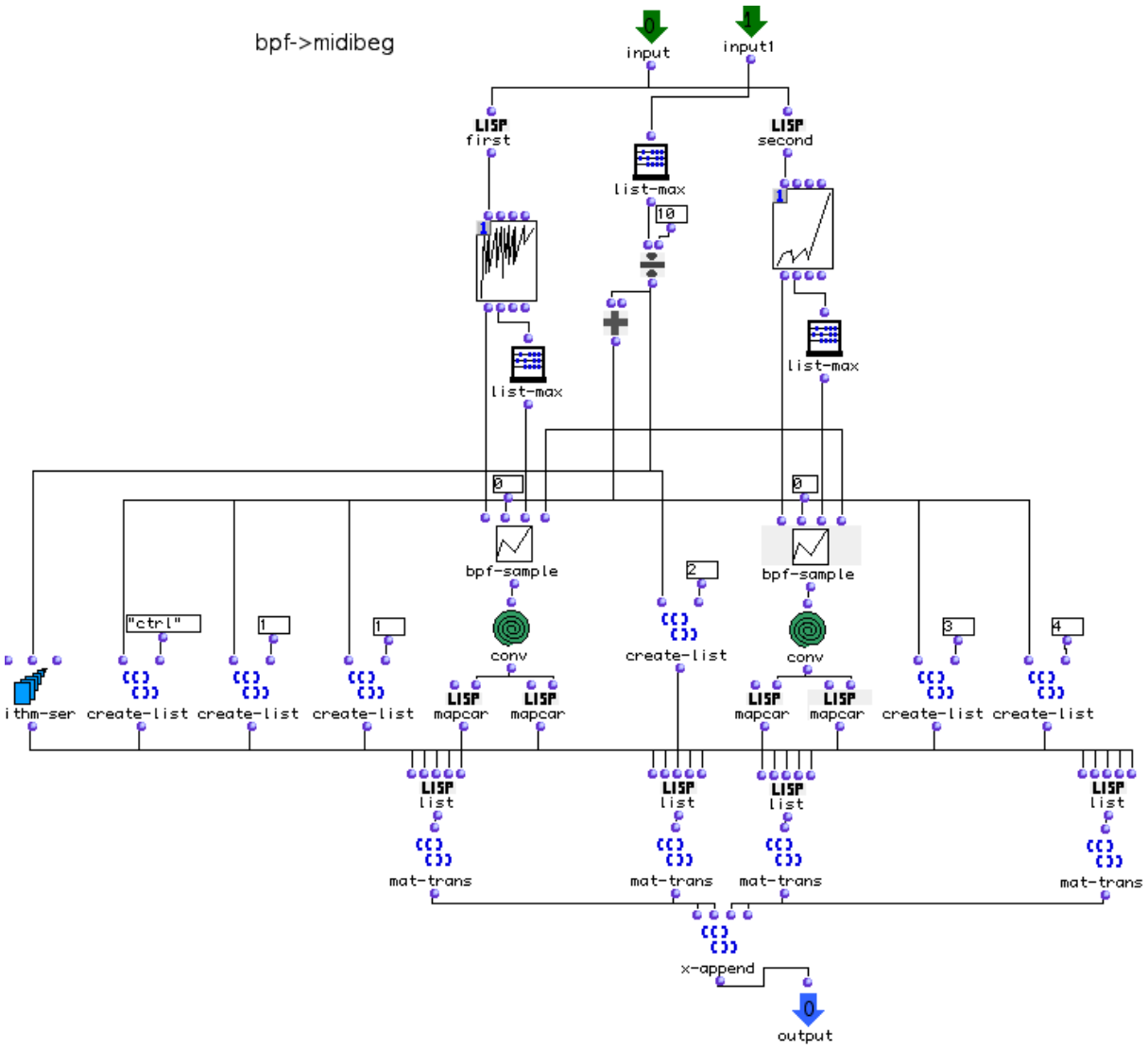


fig28

C'est cet objet qui crée les listes de 5 éléments que j'ai décrites plus haut. Dans les BPF, j'ai gardé la même graduation temporelle que pour les messages MAX, c'est à dire la milliseconde. Les dates MIDI, elles, sont exprimées en 1/100^{e} de seconde. Pour optimiser le transfert de données, on créé donc des messages MIDI séparés d'1/100^{e} de seconde. Pour cela, on prend la valeur maximale de l'axe temporel, on la divise par 10 et le résultat nous donne une valeur dont on se sert pour sampler les BPF.

C'est dans les boucles *conv* qu'on code les valeurs issues des BPF. On peut coder des valeurs entières allant jusqu'à 16383 en utilisant deux « transporteurs » MIDI (deux fois sept bits): on divise la valeur par 128 puis on transporte le quotient avec un message MIDI et le reste de la division avec un autre message MIDI.

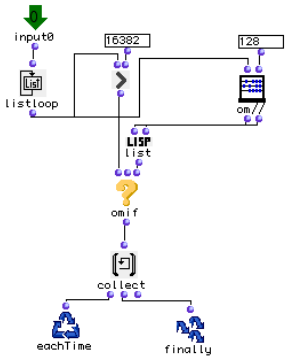


fig29

Pour les paramètres à valeurs flottantes comme par exemple transpose, on code de la même manière avec en plus une multiplication par 100 donc une précision de 10^{-2} (qui est suffisante).

On peut également transporter les données par le système exclusif qui présente l'avantage de transporter autant de valeurs qu'on le souhaite.

V.5.2 Lecture des fichiers MIDI dans MAX

On vient de voir comment coder les informations provenant de OM sous forme de fichiers MIDI . Voyons maintenant comment récupérer les données dans MAX. MAX possède un séquenceur MIDI, l'objet *seq.* Le sous-patch *p seqbegtr* est construit autour de cet objet (figure 30) :

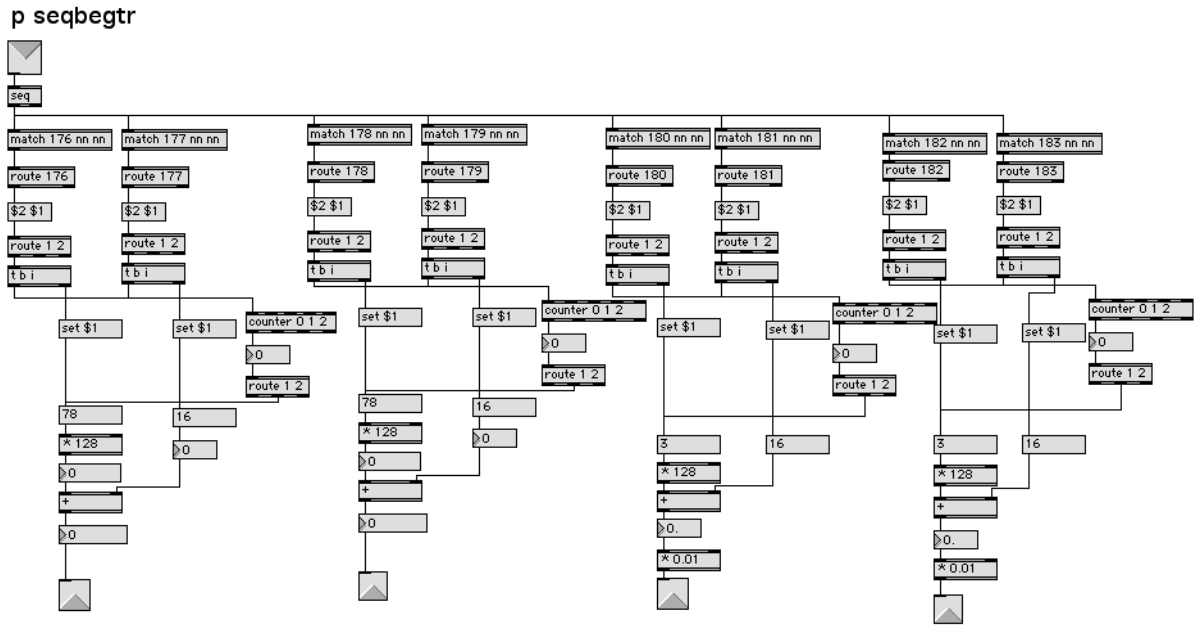


fig30

Ce sous-patch récolte les informations que l'on a codées à la figure 27. On a utilisé huit contrôleurs différents ; quatre pour le paramètre *begin* (deux pour la borne inférieure et deux pour la borne supérieure) et quatre pour le paramètre *transpose*.
P seqbegtr ré assemble les informations venant des huit contrôleurs. Pour chaque évènement MIDI, *seq* fournit trois informations à chaque date:

- le numéro du contrôleur
- la valeur MIDI transportée
- le numéro du canal MIDI utilisé

Il faut donc récupérer à chaque événement la valeur :

- avec *match* on crée une liste de la forme (numéro de contrôleur, valeur, canal)
- avec *route « numéro de contrôleur »* (176 pour le contrôleur 1, 177 pour le contrôleur 2 etc…), on crée la liste (valeur, canal)

- avec *$2 $1* on crée la liste (canal, valeur)
- avec *route « numéro canal »* (numéro de canal est toujours égal à 1 dans nos fichiers MIDI), on récupère la valeur

Ensuite, on lui fait subir le traitement inverse de celui qu'elle a subi lors de son codage dans OM. Par exemple pour reconstituer les valeurs de la borne supérieure de *begin*, on multiplie la valeur du contrôleur 1 par 128 avant de l'ajouter à celle du contrôleur 2. Pour *transpose*, c'est la même chose avec en plus une multiplication par 0.01 (on avait multiplié par 100 dans OM pour pouvoir transporter les valeurs sous forme d'entier).
Si on branche les sorties de *p seqbegtr* aux entrées des boîtes à nombre de *drivegran*, on peut lire l'évolution temporelle des bornes sous forme de fichiers MIDI.
Ce mode de contrôle ne se trouve pas encore dans *drivegran*.

VI Perspectives

VI.1 Optimisation

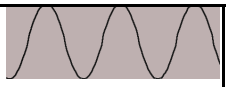
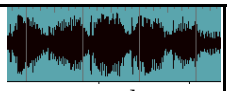
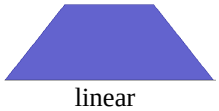
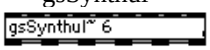
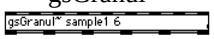
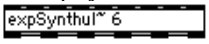
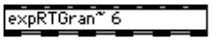
Les objets MAX que j'ai présentés sont fonctionnels et sont le résultat de multiples versions que j'ai élaborées tout en apprenant le langage MAX et en réfléchissant sur le contrôle. Ce travail a été relativement empirique. On se rend compte des avantages et défauts des objets une fois qu'ils sont terminés et qu'on peut les tester. *Drivegran* réunit les possibilités de contrôle qui nous paraissent les plus essentielles mais cet objet peut encore être optimisé, notamment en travaillant sur l'objet *nctrl2i* qui oblige à un câblage assez lourd (voir figure 3).
D'autre part, on ne peut prédire les désirs des utilisateurs vu les possibilités énormes de la synthèse granulaire. Je suis d'ailleurs en train de développer un objet permettant de passer continûment d'un son à l'autre (sorte de morphing sonore) en utilisant la granulation selon les volontés d'un compositeur qui va venir travailler prochainement au GMEM.

VI.2 Exploration d'autres moyens de transférer les données

J'ai étudié le problème du transfert de données entre OM et MAX à l'aide de fichiers textes et de fichiers MIDI en utilisant les bibliothèques conçues par L. Pottier, ces bibliothèques avaient d'ailleurs été créées pour permettre le transfert d'informations de OM vers Csound (travail de L. Pottier en collaboration avec M. Rocha).
Un projet consiste à également explorer ce type de transfert sous forme de fichiers SDIF.

VI.3 Autres générateurs de grains

Comme je l'ai déjà expliqué, mon travail a porté sur le contrôle d'un objet précis *LKpgran~* (équivalent à l'objet *lnGranul~* du tableau ci-après), implémenté par L. Kessous, qui est un générateur de grains à enveloppes linéaires à partir de fichiers audios. Mais le projet du GMEM est de réaliser neuf générateurs de grains différents :

envelopes \ waveforms	 sine waveform	 sample	 audio input
 linear	lnSynthul~ 	lnGranul~ 	lnRTGran~ 
 gaussian	gsSynthul~ 	gsGranul~ 	gsRTGran~ 
 cosine	expSynthul~ 	expGranul~ 	expRTGran~ 

Les possibilités de ces générateurs ne sont pas identiques, leur contrôle ne le sera sûrement donc pas non plus.

VI.4 Applications musicales

Tous ces outils sont bien entendu destinés à être utilisés par des créateurs sonores. Cette étape constitue la validation de leurs performances. J'ai pu tester une version de *drivegran* avec le compositeur Marseillais J.L. Gergonne avec qui j'ai constitué une banque de sons à partir de laquelle il est en train de composer une pièce que j'espère pouvoir diffuser lors de la soutenance de ce rapport.

Jean-Luc Gergonne

VI.5 Apports personnels

Le GMEM a été pour moi une structure idéale pour réaliser mon stage de DEA ATIAM. Elle fait se rencontrer « recherche en informatique musicale » et « création musicale ». C'est un lieu où réside un échange permanent entre musiciens de passage, équipe de recherche (L. Kessous, L. Pottier), équipe technique (J. Decque, H. Barroero), équipe administrative (C. Mayaudon, A. Arciéro, I. Matteo, E. Arrigo) et direction (R. de Vivo). Dans ce cadre, j'ai pu utiliser de nombreuses notions qui m'ont été enseignées à la formation ATIAM et en aborder d'autres très précieuses : essentiellement l'utilisation des applications incontournables que sont MAX/MSP, Open Music et Csound.

Bibliographie

(1).ARFIB D., « Digital synthesis of complex spectra by means of multiplication of nonlinear distorded sine waves », *Journal of the Audio Engineering Society* 27, 1979, p.757-768.

(2).ASSAYAG G., « Du calcul secret au calcul visuel », *Interfaces homme-machine*, Edition Hermès, 1999, p.37-65.

(3).BEAUDOUIN-LAFON M., « Moins d'interfaces pour plus d'interaction », *Interfaces homme-machine*, Edition Hermès, 1999, p.123-141.

(4).BLOCH G., Cours DEA ATIAM 2002-2003.

(5).BOSSEUR D. et J.Y., *Révolutions musicales*, Minerve, 1986.

(6).BOULEZ P., « Quoi ? Quand ? Comment ? », *Quoi, quand, comment la recherche musicale*, Christian Bourgois Editeur, 1985, p.271-285.

(7).CADOZ C., « Continuum énergétique du geste au son, simulation multisensorielle interactive d'objets physiques », *Interfaces homme-machine*, Edition Hermès, 1999, p.165-181.

(8).CADOZ C., « Timbre et causalité », *Le timbre, métaphore pour la composition*, Ed. Christian Bourgois, 1991.

(9).CADOZ C., « Musique, geste, technologie », *Les nouveaux gestes de la musique*, sous la direction de H. Genevois et R. de Vivo, Marseille, Ed. Parenthèses, 1999, p.47-92.

(10).CADOZ C., LUCIANI A., FLORENS J.L., « CORDIS-ANIMA Système de Modélisation et de Simulation d'Instruments et d'Objets Physiques pour la Création Musicale et l'Animation d'Images », *Actes du colloque International sur les modèles physiques*, Grenoble, 1990.

(11).CHOWNING J., « The synthesis of complex audio spectra by means of frequency modulation », *Journal of the audio Engineering Society* 21, 1973, p. 526-534.

(12).COHEN-LEVINAS, « Gérard Grisey : Du spectralisme formalisé au spectralisme historisé », *Vingt-cinq ans de création musicale contemporaine*, L'Harmattan, deuxième édition, 1998, p.51-67.

(13).DEPALLE P., TASSART S., WANDERLEY M., « les instruments virtuels », revue *Résonnance*, IRCAM, septembre 1997, P.5-8.

(14).DUFOURT H., « L'ordre du sensible », *Quoi, quand, comment la recherche musicale*, Christian Bourgois Editeur, 1985, p.223-240.

(15).GRISEY G., « Autoportrait avec l'Itinéraire », *Vingt-cinq ans de création musicale contemporaine*, L'Harmattan, deuxième édition, 1998, p.41-51.

(16).MALHERBE C., Cours DEA ATIAM 2002-2003.

(17).MURAIL T., « Spectres et lutins », *Vingt-cinq ans de création musicale contemporaine*, L'Harmattan, deuxième édition, 1998, p.309-323.

(18). QUINET J.J., *Le système MIDI, techniques de base*, Publication ACME, 1993.

(19).RISSET J.C., « Evolution des outils de création sonore », *Interfaces homme-machine* , Edition Hermès, 1999, p.17-33.

(20).RISSET J.C., « Nouveaux gestes musicaux : quelques points de repère historiques », *Les nouveaux gestes de la musique*, sous la direction de H. Genevois et R. de Vivo, Marseille, Ed. Parenthèses, 1999, p.19-33.

(21).RISSET J.C., Cours DEA ATIAM 2002-2003.

(22).ROADS C., « Asynchromous Granular Synthesis », *Representations of Musical Signals*, MIT Press, 1991, p.143-186.

(23).ROADS C., *Microsound*, MIT Press, 2001.

(24).ROCHA M., *Les techniques granulaires dans la synthèse sonore*, thèse de doctorat en Musique, Université Paris VIII, 1999.

(25).RODET X., « Time-domain Formant-wave-function Synthesis », J.C. Simon Ed., *Spoken Language Generation and Processing*, D. Reidel Publishing Company, 1980.

(26).RODET X., POTARD Y., BARRIERE J.B., « The CHANT Project : from the synthesis of the singing voice to synthesis in general », *Computer Music Journal* 8, 1984, p.15-31.

(27).SAPIR S., « Informatique musicale en temps réel : geste-structure-matériau », colloque International *Structures Musicales et Assistance Informatique*, Marseille, Juin 1988, p.5-19.

(28).WANDERLEY M., « Non-obvious Performer Gestures in Instrumental Music », *Gesture based communication in human-computer interaction*, Lecture notes in Artificial Intelligence 1739, Mars 1999, p.37-48.

(29).WANDERLEY M., DEPALLE P., « Contrôle gestuel de la synthèse sonore », *Interfaces homme-machine*, Edition Hermès, 1999, p.145-163.

(30).XENAKIS I., *Formalized Music*, Bloomington : Indiana University Press, 1971.

Webographie

(31).http://www.csounds.com/

(32).http://www.granularsynthesis.live.com.au/hthesis/contents.html

(33).http://www.ircam.fr/equipes/repmus/OpenMusic/Documentation/OMDevDocumentation/index.html

(34).http://www.mle.ie/nime/processions/navig/toc.html

(35).http://www.musique.umontreal.ca/electro/

Annexe 1 : Les langages MusicN

En 1957, Max Mathews fut le premier chercheur à fabriquer de la matière sonore avec un ordinateur, dans les Laboratoires de Bell Telephone au New Jersey. La grande majorité des langages de traitement sonore contemporains sont des descendants directs de ces premiers efforts (MusicI à MusicV). Le logiciel Csound, qui est de ceux-ci, a été créé par Barry Vercoe, du MIT en 1991 et est aujourd'hui un des langages de synthèse et de traitement sonore numérique les plus couramment utilisés. Les langages de la famille MusicN fonctionnent presque tous sur le modèle "orchestre/partition". Le calcul de synthèse est ainsi effectué à partir de deux documents texte qui sont :

- le fichier **orchestre** (avec extension .orc de *orchestra*), le programme de synthèse
- le fichier **partition** (avec extension .sco de *score*) contenant la déclaration des tables de formes d'onde et la liste des évènements à exécuter avec le programme orchestre

La synthèse sonore logicielle repose sur deux principes:

- une boucle d'exécution qui calcule la valeur de l'échantillon sonore courant selon les directives du programme de l'usager,
- une conversion de ces échantillons en un signal analogique continu pour audition.

Dans les systèmes actuels, ces deux étapes sont souvent exécutées en temps réel. Les échantillons seront, selon le cas, écrits dans un fichier audionumérique, ou envoyés directement vers un convertisseur audionumérique (DAC). Pour obtenir le résultat sonore, le logiciel Csound est lancé avec les deux fichiers en argument.

1 Le fichier orchestre

Le fichier orchestre contient les instructions (programme) pour le compilateur audio de Csound. Ce fichier texte débute habituellement avec une **entête**, où sont donnés :

- le taux d'échantillonnage **sr** (de *sampling rate*) pour les signaux audio
- le taux de contrôle (**kr**) pour les signaux de contrôle (pour les enveloppes par exemple)
- le nombre de canaux audio du signal de sortie.

Exemple :

---

sr = 44100
kr = 4410
ksmps = 10
nchnls = 2

Avec l'entête donnée dans cet exemple, les signaux audio seront échantillonnés à 44100 Hz, les signaux de contrôle seront, quant à eux, mis à jour à un taux 10 × plus bas (c'est-à-dire 4410 Hz), et le signal de sortie sera stéréo vu que le nombre de canaux est ici fixé à 2. La variable **ksmps** donne le rapport entre les deux taux **sr/kr**.

2 Les instruments

Tous les langages MusicN fonctionnent selon le modèle de **générateur unitaire**. Ces générateurs (qu'on nommera **opcode**) sont des modules accomplissant une tâche particulière et sont utilisés pour constituer les **instruments** de l'orchestre. Le langage de programmation permet de connecter plusieurs opcodes les uns aux autres par le biais de "variables" et ainsi donne la possibilité de construire des réseaux de synthèse et de traitement très complexes. Typiquement, un opcode aura une ou plusieurs variables de sortie, et un ensemble de paramètres.

3 Le fichier partition

Le fichier partition contient la liste des événements à exécuter par l'orchestre lorsque le calcul est lancé. Comme le fichier orchestre, le fichier partition comprend deux parties :

- les déclarations de **tables** où sont définies les fonctions génératrices de formes d'onde
- les déclarations de **notes** qui "jouent" les instruments avec les paramètres spécifiés.

4 Les déclarations

C'est dans cette partie que sont appelées les sous-routines génératrices de fonctions de Csound, appelées **GENS**. Chacune d'elles (il y en a une trentaine) est optimisée pour calculer une classe spécifique de fonctions ou de formes d'onde. Par exemple, les sous-routines **GEN5** et **GEN7** construisent des fonctions à partir de segments de courbes exponentielles ou de droites; **GEN9** et **GEN10** génèrent des formes d'onde complexes à partir d'une somme de sinusoïdes; **GEN1** permet de transférer dans une table des échantillons en provenance d'un fichier-son préenregistré (en vue d'utiliser une fonction de mise en boucle par exemple).

5 Les déclarations de notes

Dans cette seconde partie du fichier partition, on établit la liste des notes qui "jouent" des instruments spécifiés dans le fichier orchestre. Toutes les lignes de cette partie commencent par la lettre **i**. Les trois premiers paramètres qui suivent cette lettre **i** ont toujours la même signification : le numéro d'instrument qui jouera cet événement, le moment ou cet événement doit survenir et sa durée.
Quand aux autres champs-P à la suite des trois premiers, il revient au programmeur de décider de leur fonction. Typiquement, **P4** est réservé pour l'amplitude et **P5** est réservé pour la fréquence.
Il faut donc bien comprendre que les paramètres des "notes" sont passés à l'orchestre par le biais des "champs-P" qui donnent une valeur d'initialisation pour chaque événement.

Annexe 2 : La synthèse additive

La synthèse additive est l'une des toutes premières méthodes utilisées pour l'obtention de spectres sonores riches se rapprochant du comportement naturel des sons. L'analyse acoustique des signaux audio nous apprend que les sons naturels sont en effet composés d'une multitude de composantes simples, celles-ci étant, dans les cas de sons périodiques à hauteur déterminable, des multiples entiers de la fréquence fondamentale.

Le concept de synthèse additive est apparu dès le Moyen-Âge, appliqué aux orgues qui pouvaient comprendre des ensembles de plusieurs tuyaux pour chaque hauteur (accompagnement par plusieurs quintes et octaves). Plus tard, on la retrouve avec les débuts de la musique électrique et électronique (le Telharmonium en 1906 et les orgues Hammond).

On distingue la **synthèse additive à forme d'onde fixe**, de la **synthèse additive à forme d'onde variable**.

La synthèse additive à forme d'onde fixe n'est pas toujours satisfaisante à l'oreille. La synthèse par addition de partiels fixes est limitée puisqu'elle ne peut pas donner de **spectre dynamique**, c'est-à-dire qui varie dans le temps. La forme d'onde, bien qu'elle contienne plusieurs harmoniques, ne se comporte pas comme dans les signaux naturels dont la forme d'onde change dans le temps.
De plus, les sons obtiennent leur caractère distinctif grâce aux périodes d'**attaque** et de **chute**, périodes durant lesquelles le spectre du son est très différent et varie rapidement. Le problème posé par la synthèse à forme d'onde fixe est que ces caractéristiques essentielles y sont incontrôlables.

Pour obtenir des timbres de synthèse plus convaincants et riches, on voudra donc changer au cours du temps les caractéristiques et le nombre d'harmoniques présents dans le signal. Bien que coûteuse en temps de calcul, la synthèse additive à forme d'onde variable est, en principe, très générale et permettra de créer des familles entières de sons. Il s'agira d'utiliser, pour chaque partiel désiré dans le spectre, un générateur d'onde sinusoïdale dont on pourra contrôler de façon indépendante la fréquence et l'amplitude. Les sorties des oscillateurs sont additionnées en un seul signal. Pour synthétiser avec fidélité des sonorités naturelles, il faut utiliser un grand nombre d'oscillateurs. Pour une synthèse réaliste, on pourra avoir besoin de plus d'une trentaine d'oscillateurs. Cette méthode est donc très onéreuse, tant pour le temps de calcul que pour le nombre de données à spécifier pour le contrôle de cette banque d'oscillateurs.

Annexe 3 : La synthèse par modulation

La synthèse sonore par modulation permet de contourner un des problèmes les plus épineux de la synthèse additive qui est la production d'une grande quantité de données pour le rendu de timbres suffisamment riches.

Ce type de synthèse sonore offre une alternative intéressante parce qu'elle permet de générer des spectres riches de façon économique tant pour la spécification des données de contrôle que pour le temps de calcul nécessaire.

1 La modulation

Pour expliquer le concept de modulation, considérons d'abord un **oscillateur** générant un signal sinusoïdal. Si tous les paramètres d'entrée (amplitude, fréquence et phase) sont constants, l'équation de l'amplitude instantanée d'un signal sinusoïdal s'exprime comme suit :

$y = A \sin (2 \pi f t + F)$

où A est l'**amplitude**, f la **fréquence** et F la **phase**. Rappelons que l'on peut aussi utiliser la notation w pour la **vitesse angulaire**, liée à la fréquence via la relation $w = 2\pi f$ On peut donc écrire :

$y = A \sin (w t + F)$

Pour obtenir un phénomène de modulation, il faut mettre au moins deux oscillateurs en présence. L'idée est d'utiliser un des deux signaux pour faire varier, *moduler*, un des paramètres de l'autre signal : la sortie du premier oscillateur devient une des entrées du second oscillateur. En fonction de la nature du paramètre, on dira que l'on effectue une **modulation d'amplitude**, une **modulation de fréquence** ou une **modulation de phase**.

2 La modulation en anneau (RM)

Ce type de modulation est un « classique » de la musique électronique des premiers jours. La modulation en anneau (*Ring Modulation* - RM, en anglais) tire son nom d'une technologie analogique et correspond à la **multiplication de deux signaux bipolaires** (signal qui s'articule entre les valeurs positives et négatives autour de 0). Un signal porteur y_C est multiplié par un signal modulateur y_M :

$y_{RM} = y_C \times y_M$

Pour connaître le contenu spectral d'un signal résultant de la multiplication de deux signaux sinusoïdaux, nous avons recours aux formules trigonométriques :

$\cos a \times \cos b = [\cos(a+b) + \cos(a-b)] / 2$

Autrement dit, si on multiplie un signal porteur de fréquence C et un signal modulant de fréquence M, le signal résultant ne contiendra aucune de ces deux fréquences, mais par contre, il contiendra deux composantes, l'une à la somme des fréquences de départ (C+M) et l'autre à

la différence de ces fréquences (C-M). Quant à l'amplitude de ces composantes, elle vaudra la moitié de l'amplitude des signaux multipliés.
Si le signal porteur est complexe (constitué d'une somme de sinusoïdes) et le signal modulant sinusoïdal, la modulation en anneau aura pour effet de remplacer chaque partiel du signal porteur par une paire de composantes à la somme et à la différence des fréquences.

Si les signaux multipliés sont tous deux complexes, alors le spectre résultant est encore plus riche, puisque qu'il contient les sommes et différences de toutes les fréquences mises en présence. Les sons résultant de la modulation en anneau sont donc généralement des sons **inharmoniques** et à caractère **bruité**, surtout si les deux sources sont complexes.
Finalement, dans le cas de la multiplication de fonctions cosinus, il faut noter que si la différence entre deux fréquences donne une fréquence négative, celle-ci se repliera positivement dans le spectre.

3 La modulation d'amplitude (AM)

La modulation d'amplitude est à presque tout égard identique à la modulation en anneau sauf que le signal modulateur est un signal **unipolaire** (signal se situant uniquement dans les valeurs positives).
L'application la plus évidente de la modulation d'amplitude est la multiplication du signal porteur par une enveloppe d'amplitude. Musicalement, une modulation lente (fréquence modulante inférieure à 20 Hz) correspond à un **trémolo**.
Dans le cas de signaux sinusoïdaux, on peut exprimer mathématiquement la modulation d'amplitude de la manière suivante :

$y_{AM} = A_C \sin (2 \pi f_C t)$ où $A_C = [1 + A_M \cos (2 \pi f_M t)]$

On obtient donc :

$[1 + A_M \cos (w_M t)] \times \sin (w_c t) = \sin (w_c t) + A_M [\sin (w_c t) \times \cos (w_M t)]$
$= \sin (w_c t) + (A_M / 2) [\sin (w_c + w_M) t + \sin (w_c - w_M) t]$

Comme l'indique l'équation finale, le spectre résultant contient la fréquence porteuse f_C ainsi que deux autres composantes à ($f_c + f_M$) et ($f_c - f_M$). Exemple : Si la fréquence porteuse vaut 1000 Hz et la fréquence modulante vaut 400 Hz, le spectre résultant contient la porteuse à 1000 Hz ainsi que des composantes latérales respectivement à 600 et 1400 Hz.
Dans l'équation donnée plus haut, le facteur A_M (l'amplitude du signal modulant) est le **facteur de modulation** qui permet d'ajuster l'amplitude des composantes latérales par rapport à la composante centrale. Si le facteur vaut 1, l'amplitude des composantes latérales vaut la moitié de l'amplitude de la composante centrale.

4 La modulation de fréquence (FM)

La modulation de fréquence fut une étape importante dans le développement des méthodes modernes de synthèse sonore.
Ce type de synthèse sonore, exploitant dans la plage des signaux audio les mêmes principes qui sont utilisés dans la transmission radiophonique FM, fut inventée en 1967 dans le département de musique de Stanford par **John Chowning**. Une variante de ce type de

synthèse (par modulation de phase) a été implémentée dans les fameux synthétiseurs DX7 de Yamaha.
En transmission radio, le processus implique la modulation de la fréquence d'une onde porteuse par le signal contenant l'information à transmettre. La fréquence porteuse d'un signal de radio FM est de l'ordre de la centaine de MégaHertz (10^8 Hz). Quand cette méthode est appliquée à la synthèse sonore, la fréquence porteuse est ramenée dans la gamme des fréquences audio (de 20 à 20 000 Hz) et la modulante, dans les fréquences audio ou, plus souvent, sub-audio.
Si la fréquence modulante est inférieure à 8 Hz, le résultat de la modulation est un **vibrato**. Pensons au violoniste qui génère un vibrato sur son instrument en faisant varier rapidement la longueur de la corde (et ainsi la hauteur résultante), par un mouvement du poignet et du doigt pressé sur la corde. Mais si cette fréquence est autour de 20 Hz, on obtient une **modification de timbre** du son modulé. Entre ces deux fréquences, il y a une transition progressive d'un effet à l'autre. On peut aussi expérimenter avec des fréquences modulantes plus grandes, voire même supérieures à la fréquence porteuse.
Le signal modulé peut être exprimé mathématiquement comme suit :

$$y_{FM} = A_C \sin [w_C + A_M \sin (w_M t)] t$$

Le spectre résultant contient la fréquence de la porteuse ainsi que toute une série de fréquences distribuées de façon symétrique autour de la fréquence porteuse, à des distances valant tous les multiples entiers de la fréquence modulante. Ces groupes de composantes de part et d'autre de la porteuse sont appelées les **bandes latérales**.
Exemple : si la fréquence porteuse C est 800 Hz et la fréquence modulante M est 200 Hz, les bandes latérales contiennent des composantes aux fréquences suivantes :

C - M = 600 Hz	C + M = 1000 Hz
C - (2xM) = 400 Hz	C + (2xM) = 1200 Hz
C - (3xM) = 200 Hz	C + (3xM) = 1400 Hz

etc..

Dans cet exemple le rapport des fréquences C/M est un entier. Il en résulte un spectre de nature **harmonique**. La hauteur perçue correspond à la fréquence modulante elle-même (ici 200 Hz). Si le rapport C/M n'est pas un rapport d'entiers, le spectre résultant est **inharmonique**. Si le signal subissant la modulation contient plusieurs partiels, des bandes latérales apparaîtront autour de chacun de ces partiels. Il en résulte un enrichissement considérable du spectre.

5 Modulation de fréquence ou modulation de phase ?

Dans son article (11), John Chowning donne la formule suivante pour décrire mathématiquement la synthèse FM :

$y = A \sin [w_C t + I \sin (w_M t)]$ où

- A est l'amplitude de la porteuse
- w_C est la fréquence angulaire de la porteuse (en radians par secondes)
- w_M est la fréquence angulaire modulante
- I est l'index ou taux de modulation

Il faut remarquer que cette formule est en réalité l'expression mathématique d'une **modulation de phase**. Mais il faut savoir que modulation de fréquence et modulation de phase sont équivalentes. En effet, moduler la phase avec la fonction m(t) revient à moduler la fréquence avec m'(t), la dérivée de m(t). Donc, si une fonction sinus module la phase, une fonction cosinus (sa dérivée) module la fréquence. La modulation de phase est précisément la méthode qui est implantée sur les synthétiseurs Yamaha DX7. Cette méthode offre certains avantages par rapport à la modulation de fréquence, tout en permettant de générer le même type de sons. Le premier avantage de la modulation de phase est qu'elle permet l'**automodulation** d'un oscillateur sans changer la fréquence fondamentale de la forme d'onde résultante. Le second avantage est que la fondamentale ne dérive pas si la forme d'onde modulante contient une composante continue (valeur moyenne non nulle). Par contre, il faut noter que la modulation de phase ne fonctionne correctement que sur des synthétiseurs numériques car elle requiert des oscillateurs très stables. L'acronyme FM est tellement courant qu'on le rencontre dans la plupart des documents traitant du sujet, même si la modulation de phase est la technique utilisée en pratique.

6 Bandes latérales et fonctions de Bessel

Mathématiquement, on peut décomposer la formule donnée par Chowning

$$y = A \sin [w_C t + I \sin (w_M t)]$$

en une somme de termes correspondant à des sinusoïdes aux fréquences C, C+M, C-M, C+2M, C-2M, C+3M, C-3M, etc...
L'amplitude des composantes aux fréquences C+nM et C-nM est donnée par $J_n(I)$, où J_n est une fonction de Bessel de première espèce et d'ordre n, fonction de l'indice de modulation I.
On comprend dès lors qu'une variation de l'indice de modulation I va induire une variation importante du contenu spectral. Quand I = 0, l'amplitude de la porteuse est maximale et il n'y a pas de bandes latérales (l'effet de modulation est annulé). Si on augmente progressivement la valeur de I, l'amplitude de la porteuse diminue tandis que les bandes latérales apparaissent.

Annexe 4 : La synthèse soustractive

1 Principe

La synthèse soustractive est l'opération inverse de la synthèse additive. Cette méthode implique l'utilisation de filtres pour modifier le spectre d'une source sonore. Le filtre amplifie ou atténue des régions désignées du spectre. Si la source possède un spectre riche et si le filtre est flexible, la synthèse soustractive peut permettre de sculpter des sons à sonorité naturelle (tels que la voix et les instruments traditionnels) ainsi que tout une variété de timbres originaux. Le résultat d'une synthèse soustractive dépend du type de traitement mais aussi, bien évidemment, de la source de départ. C'est le contenu spectral d'une source qui conditionne son choix. Si on a l'intention de traiter des composantes dans une certaine région du spectre, il est important de s'assurer que de l'énergie à ces fréquences existe dans la source, sinon il n'y aura rien à filtrer. C'est pour cela que les **bruits** et les **trains d'impulsions** sont des sources couramment utilisées comme matériau de base de la synthèse soustractive, car elles offrent une étendue uniforme de composantes dans toute la gamme des fréquences audibles.
Les filtres les plus courants sont les filtres passe-bas, passe-haut, passe-bande et coupe-bande (ou réjecteur de bande), ils sont caractérisés par leur **réponse en fréquence** ou **spectre d'amplitude**. Les filtres peuvent être combinés en série et/ou en parallèle afin d'obtenir des réponses en fréquence plus complexes.

2 Les filtres numériques

Le fonctionnement de base d'un filtre numérique est relativement simple. On distingue en fait deux types de fonctionnement :

- on **retarde** légèrement une copie du **signal d'entrée** (d'une ou plusieurs périodes d'échantillonnage) et on combine le signal d'entrée retardé avec le nouveau signal d'entrée. Les filtres numériques basés sur ce fonctionnement sont dit à "réponse impulsionnelle finie" ou **FIR** (pour Finite-Impulse-Response).
- on **retarde** une copie du **signal de sortie** que l'on combine au nouveau signal d'entrée. Les filtres numériques basés sur ce fonctionnement sont dit à "réponse impulsionnelle infinie" ou **IIR** (pour Infinite-Impulse-Response). On les qualifie également de filtres **récursifs** ou à "feedback".

Les filtres FIR offrent en général une réponse de phase plus linéaire et ils n'entrent jamais en oscillation (c'est-à-dire deviennent instable) puisqu'ils sont dépourvus de récursion. Mais ils requièrent un grand nombre de termes dans leurs équations et sont ainsi plus coûteux en temps de calcul. Un filtre FIR avec coupure très nette (bande de transition très courte) peut requérir jusqu'à des centaines de délais. Les filtres IIR, quant à eux, sont très efficaces et peuvent donner des pentes de coupure très raides. Toutefois, vu les caractéristiques de feedback, ils ont tendance à entrer en oscillation et à résonner.

Annexe 5 : La distorsion non-linéaire (waveshaping)

L'illustration la plus populaire des effets de la distorsion non-linéaire est celle de l'amplificateur de guitare poussé au-delà de ses capacités d'amplification. Un amplificateur peut amplifier un signal linéairement (sans déformer la forme d'onde) jusqu'à une certaine limite. Au-delà de cette limite, l'amplificateur **sature**, ce qui se traduit en un enrichissement du spectre dans les hautes fréquences.
Un autre moyen de produire de la distorsion est d'utiliser une boîte de distorsion (*fuzz-box*). Chaque boîte de distorsion a sa propre couleur et celle-ci est déterminée par la **fonction de transfert** de son circuit électronique.
Quand la distorsion non-linéaire est appliquée à un signal audio, on parle d'**effets de distorsion**. Mais la distorsion non-linéaire peut être aussi utilisée comme outil de **synthèse sonore**. Il s'agit de la ***waveshaping synthesis***, technique de synthèse développée par **D. Arfib** (1). Ce type de synthèse permet de simuler le comportement typique des instruments acoustiques: plus on joue "fort", plus le spectre du son émis est riche. On simule cet effet en **variant l'amplitude du signal d'entrée** qui détermine le contenu spectral du signal de sortie vu que des zones différentes de la fonctions de transfert seront utilisées. On obtient alors des spectres dynamiques.

Il est possible de prédire exactement le contenu harmonique du signal de sortie à condition que le signal d'entrée soit un signal cosinusoïdal et que la fonction de transfert soit générée à l'aide d'une famille de polynômes dit de **Chebychev**. Ces polynômes T_k possèdent la propriété suivante :

$$T_k [\cos (q)] = \cos (k q)$$

Les huit premiers polynômes de Chebychev sont :

$T_0 = 1$
$T_1 = x$
$T_2 = 2 x^2 - 1$
$T_3 = 4 x^3 - 3 x$
$T_4 = 8 x^4 - 8 x^2 + 1$
$T_5 = 16 x^5 - 20 x^3 + 5 x$
$T_6 = 32 x^6 - 48 x^4 + 18 x^2 - 1$
$T_7 = 64 x^7 - 112 x^5 + 56 x^3 - 7 x$
$T_8 = 128 x^8 - 256 x^6 + 160 x^4 - 32 x^2 + 1$

où $x = \cos q$.

L'avantage d'utiliser les fonctions de Chebychev est que l'on peut garantir que la sortie du "déformeur d'onde" (*waveshaper*) est **limitée en fréquence** (c'est à dire que le spectre ne contient pas de fréquences dépassant la fréquence de Nyquist).

Annexe 6 : La synthèse par modélisation physique

La modélisation physique est une classe de méthodes qui visent à recréer les caractéristiques acoustiques des instruments de musique ou de la voix en mettant en fonction les principes physiques mis en jeu pour émettre des sons. La synthèse par modélisation physique est donc une technique de synthèse sonore qui, au lieu de s'attacher à reproduire le son lui-même (par l'analyse des fréquences qui le composent par exemple), part du dispositif physique producteur de ce son. Ainsi, cette technique se démarque des méthodes de synthèse numérique additive, soustractive, ... L'objectif de cette synthèse est autant scientifique qu'artistique car celle-ci permet aux scientifiques d'approfondir leur connaissance des phénomènes acoustiques et offre aux musiciens un nouveau matériau pour la composition.
De l'analyse du comportement des instruments, on cherche à déduire des algorithmes de **simulation**. Les modèles physique impliquent en général des calculs mathématiques intensifs et, à quelques exceptions près, ne sont utilisables pour la synthèse que depuis l'avènement des ordinateurs très rapides. Voici quelques techniques de modélisation physique parmi les plus essentielles :

1 La synthèse MSW

Parmi les grandes techniques de modélisation, certaines sont destinées à un usage principalement scientifique, comme celle de **MacIntyre, Schumacher et Woodhouse**.
Selon les principes de la synthèse **MSW** (acronyme tiré des initiales de ses inventeurs), la production du son est divisée en deux étapes : une **excitation non-linéaire** (c'est-à-dire qui, si elle dépasse certaines valeurs, provoque un changement de réaction) et une **résonance linéaire** (c'est-à-dire qui réagit en proportion de la quantité d'énergie qui lui est appliquée). L'interaction entre résonateur (ex: le corps d'une clarinette) et excitateur (ex: l'anche d'une clarinette) constitue une **rétroaction** ou **réaction** (*feedback*). Dans un modèle MSW, chaque instrument est représenté par un ensemble compact d'équations. L'excitation, non linéaire, nécessite des équations plus complexes et plus spécifiques. Les principales variables sont la source d'énergie, l'énergie fluctuante de l'élément non-linéaire et la fonction de réflexion décrivant le filtrage effectué par la partie linéaire.
Le modèle Karplus-Strong, que nous décrivons plus loin, a été reconnu comme un cas particulier des modèles physiques pour cordes de MacIntyre, Schumacher et Woodhouse.

2 Les techniques de Hiller et Ruiz

Les travaux de Hiller et Ruiz (1971) sont représentatifs de l'approche dite classique de la modélisation physique. Dans leur méthodologie, il faut indiquer les dimensions et les constantes physiques de l'objet vibrant, les conditions aux limites (risque de rupture d'une corde par exemple), l'état initial de l'objet, l'excitation et le comportement transitoire. Dans leurs modèles, les résonateurs sont modélisés par des réseaux de masses-ressorts (à une, deux ou trois dimensions). L'inconvénient majeur de cette technique est son coût de calcul.

3 La synthèse modale

La synthèse modale est une solution de rechange au paradigme des masses et ressorts. Elle a l'avantage d'être plus souple que la première. Elle est fondée sur deux contraintes primordiales, qui sont, d'une part de pouvoir appliquer aux modèles de simulation tous les

gestes instrumentaux présents dans les modes de jeu d'instruments traditionnels et d'autre part, d'obtenir un son le plus proche possible de son modèle acoustique.
Le corps sonore y est représenté comme un ensemble de sous-structures vibrantes dont le nombre est bien moins important que celui des masses et des ressorts. Ces sous-structures sont des cordes, des embouchures, des tubes, des archets, etc. qui réagissent à des excitations extérieures (souffle d'air, frappe, ...). Ce type de synthèse fut à la base du logiciel MOSAÏC développé par l'IRCAM. On connaît la version plus récente de le logiciel sous le nom MODALYS. La synthèse modale se retrouve également dans CORDIS-ANIMA (10).

4 La synthèse par guides d'ondes

La synthèse par guides d'ondes, fondée uniquement sur des techniques de traitement du signal, trouve son origine dans la découverte dans les années 80, par Kevin Karplus et Alex Strong, d'un algorithme très économique en calculs permettant d'obtenir un résultat sonore de corde pincée très convaincant.
C'est Julius O. Smith du CCRMA (Stanford) qui a formulé la synthèse par guides d'ondes dans sa forme générale.
Le guide d'ondes permet la simulation d'un corps vibrant le long duquel se propagent les ondes. Ce corps est généralement une corde ou un tuyau, mais il existe également des simulations d'objets à deux (pour les membranes) ou trois dimensions. En utilisant une paires de lignes de délais, on peut simuler deux ondes qui voyagent le long d'une corde dans des directions opposées, qui se réfléchissent aux extrémités et repartent en sens inverse, comme cela se produirait dans le cas d'une corde frappée en son milieu par un marteau.
Les délais simulent le temps pris par l'excitation (l’onde de choc du marteau ou du pincement) pour se propager le long du médium (corde, membrane, colonne d'air, ...).
Les deux fronts d'ondes progressent à travers les lignes de délais et sont renvoyés par les filtres de réflexion qui produisent une inversion de la phase et un léger amortissement dépendant de la fréquence. Le modèle peut contenir des jonctions de dispersion (*scattering junctions*), comme dans le cas des modèles d'instruments à vent où les jonctions de dispersion servent à modéliser les trous ouverts le long du tuyau.

5 La méthode Karplus-Strong

La méthode Karplus-Strong est une implantation très efficace de la simulation d’une corde pincée ou d'une membrane frappée, construite à l’aide d’un mécanisme de stimulation, d’un délai numérique, et d’un traitement.
Le signal excitateur est injecté dans une ligne de délais qui est repliée sur elle-même (rétroaction). Chaque fois que le signal est réinjecté, une modification (un filtrage habituellement) est effectué. Le signal excitateur est généralement un bruit blanc de très courte durée. La sensation de hauteur provient de la longueur variable de la ligne de délais qui implique une périodicité de la recirculation du signal de sortie vers l’entrée.
Plusieurs raffinements et variations sont possibles dans cet algorithme. Entre autres, il est possible d’étirer la chute du son et de stimuler le circuit avec un signal autre qu’une impulsion de bruit.

Annexe 7 : La synthèse vocale par fonction d'onde formantique

On considère que la voix humaine est le résultat d'une modification continuelle par le conduit vocal du signal émis par trois types de sources sonores:

- une source **voisée**, qui correspond à la vibration des cordes vocales et se présente sous la forme d'un signal quasi-périodique
- une source **fricative**, qui correspond aux turbulences engendrées par les rétrécissements en certains points du conduit buccal (lèvres, langue-palais, glotte)
- une source **plosive** qui correspond au bruit d'explosion engendré par la fermeture puis l'ouverture brusque du conduit buccal avec les lèvres ou la langue

On peut modéliser la production de voix chantée par un modèle de synthèse sonore du type excitateur-résonateur:

- l'**excitateur** correspond aux cordes vocales et définit le timbre de la voix; il est modélisé par une impulsion ou un arc
- le **résonateur** correspond au conduit vocal et définit la voyelle chantée; il est modélisé par un jeu de filtres en parallèle pour constituer une enveloppe spectrale comprenant des formants

1 Les formants

Un formant est un « pic » d'amplitude dans le spectre d'un son composé de fréquences harmoniques, inharmoniques et/ou du bruit.
Les pics formantiques sont caractéristiques des sons vocaux voisés (voyelles chantées, parlées ou murmurées) et de plusieurs instruments de musique.
Une voyelle est produite en imposant une position particulière aux différents articulateurs (lèvres, langue, ...). Le conduit vocal présente des fréquences de résonance, ce qui se manifeste dans le spectre en l'apparition de pics formantiques.
La position des formants est indépendante de la hauteur (fréquence fondamentale) et est caractéristique d'une voyelle particulière. Mais il faut aussi préciser que la position des formants pour une même voyelle peut différer d'un locuteur (ou chanteur) à l'autre. Les analyses spectrales rapportent aussi que de quatre à cinq formants importants sont présents dans tous les spectres de voix. Il est donc possible de simuler un timbre de voix chantée si on peut générer ces zones formantiques.

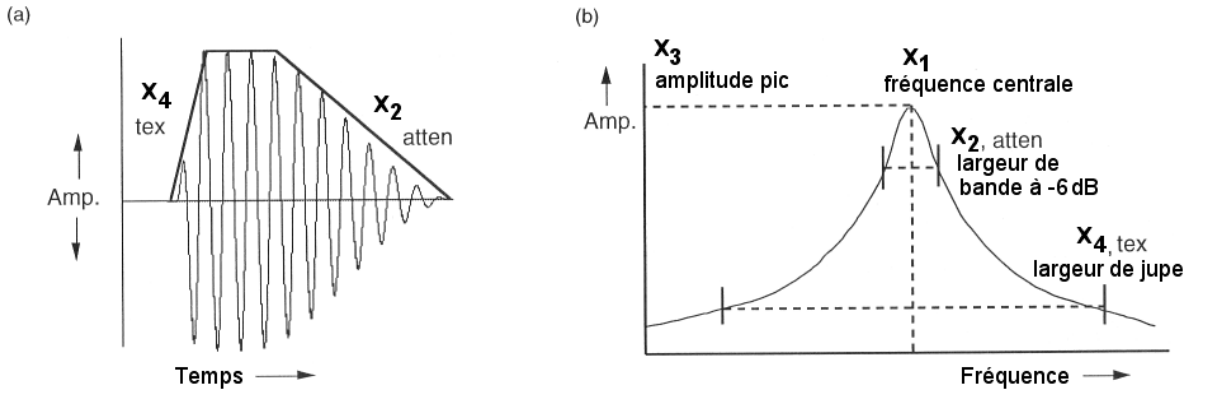
2 La synthèse à forme d'onde formantique (FOF)

La synthèse à forme d'onde formantique est à la base du système de synthèse sonore CHANT développé à l'IRCAM (25), (26). Cette technique a été implantée dans les synthétiseurs 4X (1980) et dans Csound en 1990 par J. M. Clarke. CHANT a été conçu pour modéliser une classe large de mécanismes naturels qui résonnent quand ils sont excités, et qui sont atténués par des forces physiques (de friction par exemple).
La méthode FOF part des méthodes de synthèse à formants basée sur une approche soustractive traditionnelle. En synthèse soustractive, une source à spectre large (comme un train d'impulsions ou un signal de bruit) est envoyé au travers d'un filtre complexe qui en modifie le contenu spectral pour faire apparaître des formants. Xavier Rodet a montré que les

filtres complexes utilisés en synthèse soustractive peuvent être décomposés en un ensemble équivalent de filtres passe-bande en parallèle excités par des impulsions.
Comme les FOFs ont une nature duale, une implantation alternative consiste à remplacer les filtres par une banque de **générateurs d'ondes sinusoïdales amorties**. Le signal temporel et le spectre de ces générateurs sont équivalents à ceux générés par des filtres passe-bande excités par des impulsions.
Le signal produit par un générateur FOF est appelé **grain formantique** ou **grain FOF**.
D'après Rodet, remplacer les filtres par des générateurs offre plusieurs avantages : les générateurs sinusoïdaux sont efficaces et requièrent moins de précision numérique que les filtres correspondants. De plus, les grains formantiques peuvent être progressivement changés en sinusoïdes (contrôlables en amplitude et en fréquence) permettant une transition continue entre de la synthèse FOF et de la synthèse additive.
La synthèse FOF produit des grains de son au rythme de la fréquence fondamentale désirée, ces grains étant "colorés" spectralement, de manière à reproduire les résonances formantiques.

3 Anatomie et paramètres d'un grain FOF

Un générateur FOF produit un grain de son à chaque période fondamentale. À 440 Hz par exemple, 440 grains sont produits par seconde. Chaque note contient donc un grand nombre de grains. Puisque la durée de chacun de ces grains est indépendante de la fréquence fondamentale, plusieurs grains pourront se superposer. Ceci implique que les FOFs à fréquence élevée sont plus onéreux à générer.
Un grain formantique est constitué d'une onde sinusoïdale amortie, d'une durée fixe, avec une attaque plus ou moins raide et une atténuation quasi-exponentielle. L'enveloppe temporelle d'un grain FOF est appelée l'**enveloppe locale**, en opposition à l'enveloppe globale de la note.
Le spectre d'une sinusoïde amortie est équivalent à la réponse en fréquence d'un filtre passe-bande. Le résultat de la somme de plusieurs générateurs FOF est donc un signal dont le spectre comprend plusieurs formants.
Chaque générateur FOF est contrôlé par un certain nombre de paramètres, incluant la fréquence fondamentale et l'amplitude. La figure suivante illustre les quatre paramètres de formant principaux, dans le domaine temporel à gauche (a) et dans le domaine fréquentiel à droite (b).



- **x1** est la fréquence centrale du formant

- **x2** est la largeur de bande du formant, définie comme la largeur de bande entre les points à -6 dB par rapport à la crête du formant
- **x3** est l'amplitude du formant
- **x4** est la largeur de jupe du formant qui se situe à -40 dB par rapport à la crête du formant.

Les liens entre domaine temporel et domaine fréquentiel se manifestent dans la manière dont les paramètres FOF sont spécifiés. En effet, deux paramètres du formant (**x4** et **x2**) sont spécifiés dans le domaine temporel.

- **x4** est spécifié en secondes pour la **durée de l'attaque**. Si l'attaque s'allonge, la jupe du formant se rétrécit
- **x2** est également spécifié en secondes pour la **durée de l'atténuation** du grain FOF. Une longue atténuation se traduit en un pic de résonance étroit.