

Outils pour une approche transformationnelle de l'interprétation des musiques électroniques

Benoît Courribet

CICM (Centre de recherche en Informatique et Création Musicale)
Université de Paris 8
Maison des Sciences de l'Homme Paris-Nord.

bcourribet@mshparisnord.org

Mots-clés

MIDI, séquenceur, interpolation, interprétation des musiques électroniques, contrôle probabiliste

Résumé

Ce texte présente des outils logiciels développés à l'aide de Max/MSP dans le but de proposer au musicien, dans le cadre de la composition et de l'exécution de pièces de musiques électroniques, des possibilités d'interprétation à différentes échelles structurelles. Le premier outil est un séquenceur MIDI avec contrôle probabiliste des événements, permettant des interpolations de motifs prédéfinis. Le second outil est un utilitaire générique d'interpolation de paramètres.

Contexte esthétique et technologique

En premier lieu, il convient de préciser que les outils que nous allons présenter ont été développés en fonction de critères esthétiques et technologiques inhérents à un genre musical appelé *electronica*, que l'on pourrait définir comme l'ensemble des musiques électroniques populaires [Courribet2002].

Depuis l'avènement des musiques électroniques et électroacoustiques se pose le problème de l'interprétation de ces musiques. Il reste d'actualité pour l'*electronica* et intervient lorsque l'on s'intéresse au cas du concert. On peut distinguer deux types de prestations: le *set DJ* et le *live act*.

Le *set DJ* ou *mix*, consiste à enchaîner la lecture de morceaux provenant de plusieurs sources sonores mélangées à l'aide d'une console de mixage. Si on peut penser que la fonction du DJ est juste d'assurer la transition entre les morceaux, on se rend compte assez vite (et surtout pour les DJs techno) que leur rôle s'assimile plus à celui d'un architecte. Ils développent en effet la construction d'un flux musical ininterrompu et cohérent sur des durées pouvant atteindre plusieurs heures. Le matériel de base du DJ lui permet d'opérer des modifications sur les vitesses de lecture et l'égalisation des sources sonores. L'adjonction de matériel supplémentaire (une troisième platine vinyle, des effets, une boîte à rythme) rend possible une intervention de plus en plus importante du DJ.

Le second type de prestation, le *live act*, pose plus de problèmes aux musiciens. Le problème majeur est le suivant: comme reproduire dans une situation de concert ce qui a nécessité plusieurs semaines de travail de studio? Certains musiciens n'hésitent pas à ne monter sur scène que pour lancer la restitution des morceaux (ce qui est équivalent à une simple diffusion). La position opposée consiste à proposer en concert des improvisations, cette posture s'applique aux franges les plus expérimentales de l'*electronica* et les prestations en résultant sont des plus inégales. Entre ces deux extrêmes se situe un continuum de pratiques.

La plus répandue consiste à remixer en temps réel ses propres compositions: le musicien dispose d'une version multipiste de sa composition (que ce soit MIDI ou audio) et a la possibilité de modifier, par la manipulation de la console de mixage et d'effets audionumériques, la structure interne de la composition. Il présente ainsi des variations de ses travaux.

On voit que le *live act* est fortement conditionné par les moyens technologiques mis en oeuvre, et on constate que les outils généralement utilisés pour l'établissement de la structure de la prestation fournissent des instructions de bas niveau (déclenchement de sons de manière déterministe), ce qui ne permet pas de produire ex nihilo une forme musicale. Le musicien en est donc réduit dans ce cas-là à préparer à l'avance ses structures dont il confie l'exécution aux machines.

Cependant, avec l'aide de logiciels du type de Max/MSP, on a la possibilité de contrôler des structures en temps réel à l'aide de paramètres de haut niveau (comme la densité d'évènements), ce qui laisse la possibilité au musicien, moyennant la préparation au préalable d'une ossature générale, d'avoir une grande liberté au niveau de l'interprétation de ses compositions. Sont là pour en attester les prestations des formations Autechre et Phoenecia, qui construisent leurs *live acts* par extrapolation de leurs parutions discographiques (ils en reprennent les sons, les tendances formelles...) et en proposent ainsi des versions transformées inédites.

C'est dans cette optique qu'ont été développés les outils dont nous allons maintenant décrire les fonctionnalités.

Le séquenceur *Flexstep*

Dans l'instrumentarium du musicien utilisant les technologies informatiques, le séquenceur, qu'il soit MIDI ou audio, sert à définir le placement temporel des différents événements sonores, ou variations de paramètres.

On peut classer les différents séquenceurs en deux catégories : les séquenceurs linéaires d'une part, et les séquenceurs à patterns (ou *step sequencers*) d'autre part.

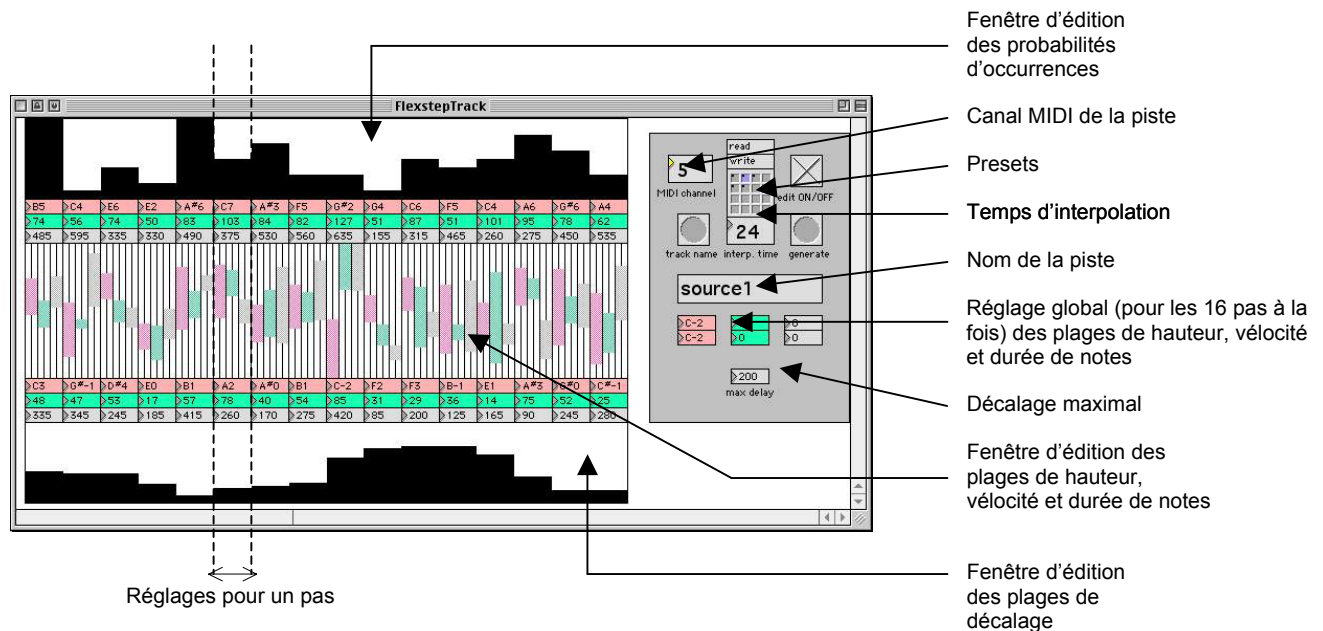
Les séquenceurs linéaires, comportent une ligne de temps destinée à être parcourue de manière linéaire. Ce type de séquenceur, s'il permet à l'utilisateur de contrôler l'activation et la désactivation des différentes pistes, ne lui donne pas la possibilité d'intervenir sur le déroulement temporel des événements.

Le *step sequencer*, en revanche, fonctionne dans une toute autre logique. Si l'architecture multipiste est conservée, la gestion du déroulement temporel est radicalement différente de celle des séquenceurs linéaires. En effet, le *step sequencer*, procède par l'enchaînement de cellules élémentaires, les patterns. Les patterns sont des motifs dont la durée est souvent de quelques mesures et qui sont destinés à être joués en boucle. Chaque mesure propose un nombre de pas (*steps*) en général limité à 16, ce qui correspond à des double-croches dans une mesure à 4/4 (ce qui est très réducteur, convenons-en). Ainsi, l'utilisateur peut stocker en mémoire différents motifs, et les déclencher à loisir durant le fonctionnement du séquenceur. Ce type de séquenceur est particulièrement bien adapté à l'utilisation lors d'un *live act d'electronica*, et c'est sur ce modèle que s'est basé le développement du logiciel *Flexstep*, tout en essayant de proposer une plus grande flexibilité dans le déclenchement des événements, afin de rompre avec la rigidité et le déterminisme des *step sequencers* traditionnels.

Il existe à l'heure actuelle deux versions de *Flexstep* : la première est conçue pour piloter un périphérique MIDI externe, la seconde pilote un instrument VSTi dans Max/MSP. Le logiciel est pour l'instant limité à un maximum de 8 pistes.

Le principe général est le suivant : au lieu de décider pour chaque pas de chaque piste si un événement MIDI a lieu ou non, comme dans la plupart des *step sequencers*, on attribue une probabilité à cet événement. A chaque début de mesure, le logiciel calcule, en fonction des probabilités, une variation du motif initial. Ainsi, on peut écrire un motif, et laisser au logiciel le soin de dériver des variations. Nous allons maintenant voir plus en détail les spécificités du logiciel.

Le logiciel est composé d'une fenêtre générale et d'une fenêtre d'édition graphique pour chaque piste. Commençons par le détail en nous intéressant à la fenêtre d'une piste.



L'écriture du pattern se réalise dans les trois fenêtres d'édition graphique. Chaque fenêtre est constituée des réglages pour chacun des 16 pas. Ainsi, pour chaque pas, on peut :

- Définir la probabilité qu'un événement MIDI ait lieu (entre 0 et 1 par pallier de 0,1)
- Dans le cas où l'événement se produit, on peut définir les plages de hauteur, vélocité et durée de la note produite (ces valeurs seront choisies aléatoirement à l'intérieur de ces plages).
- On peut enfin définir une plage de décalage temporel pour la note produite. Cette plage, entre 0 et 1 est à multiplier par le décalage maximal, défini pour tout le pattern. Ainsi, pour une valeur de 0,5 et un décalage maximal de 200 ms, la note sera décalée aléatoirement de 0 à 100 ms.

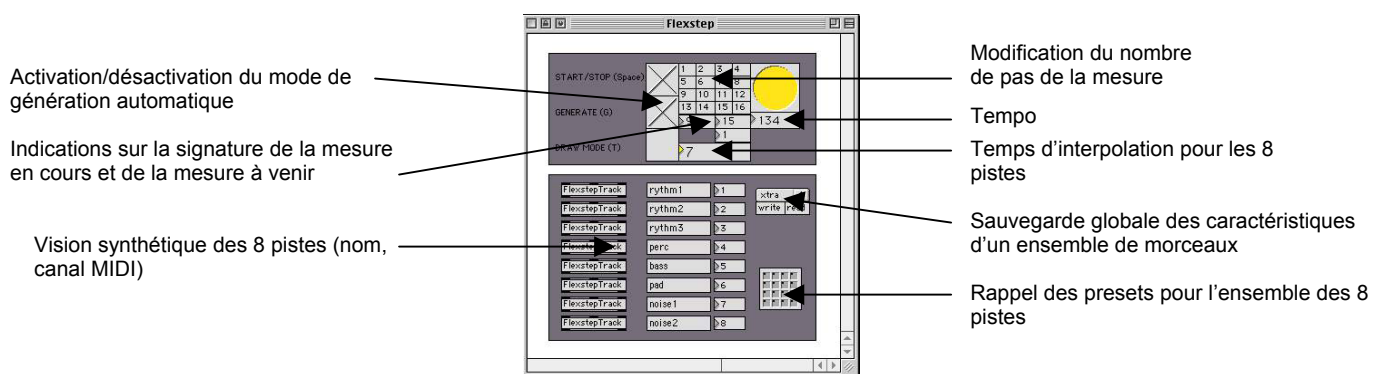
Lorsque le pattern est écrit, on peut stocker l'ensemble de ses paramètres dans un preset. Ainsi, chaque valeur de preset correspond à un pattern, dont une nouvelle variation est calculée à chaque début de mesure en fonction des différentes probabilités.

De plus, lorsque l'on désire, durant l'exécution du logiciel rappeler un pattern stocké dans un preset, on a accès à un paramètre de temps d'interpolation, qui provoque une interpolation de chacun des paramètres du pattern actuel vers les valeurs à atteindre du pattern que l'on vient de rappeler. Ce temps d'interpolation est ici donné en nombre de mesures.

Cette fonction donne accès à un contrôle global sur de véritables transformations structurelles dynamiques (des *morphings*) permettant de réaliser en temps réel le contrepoint macro-structurel de phénomènes décrits par Curtis Roads [Roads2002] à l'échelle micro-temporelle, comme la désintégration ou la coalescence. On retrouve également, pour des temps d'interpolation très longs, la notion évoquée par Steve Reich de *processus graduel* [Reich].

Il est également intéressant de noter que l'on peut décider d'écrire de manière déterministe un motif, en utilisant des probabilités égales à 0 ou 1 et en réduisant les plages de hauteur, vélocité et durée des notes à une seule valeur.

La fenêtre générale fournit quant à elle l'accès à des fonctions globales pour le morceau en cours.



Ainsi, cette fenêtre permet de sauvegarder et rappeler les caractéristiques et paramètres d'un ensemble de morceaux (nom des pistes, canaux MIDI, presets associés, tempo, nom du morceau). On peut également modifier le temps d'interpolation pour l'ensemble des 8 pistes, modifier la signature de la mesure (nombre de pas). On peut également désactiver/activer le mode de génération automatique. Lorsque ce mode est désactivé, le logiciel ne calcule pas un nouveau jeu de paramètres en fonction des probabilités et répète donc les valeurs de la mesure précédente.

Interpolation de paramètres avec l'outil *morphing*.

Toujours dans l'optique d'une approche transformationnelle de l'interprétation des musiques électroniques, un utilitaire générique d'interpolation de paramètre a été développé sous la forme d'une abstraction dans Max/MSP.

Son mode de fonctionnement est le suivant : il est destiné à être utilisé dans un contexte où l'utilisateur stocke des valeurs de différents paramètres dans des objets *preset* ou des objets *coll*. Il peut s'agir par exemple de l'ensemble des paramètres régissant un effet audionumérique. Ainsi, à chaque case de l'objet *preset* ou ligne de l'objet *coll* correspond un état morphologique du son.

L'outil *morphing* permet alors, moyennant d'être relié à chacun de ces paramètres, de réaliser des interpolations entre ces états, et de contrôler le déroulement temporel de ces interpolations. Ce contrôle peut se faire à l'aide d'un potentiomètre MIDI, par exemple. Ainsi, à chaque fois que le potentiomètre arrive en fin de course, l'outil *morphing* envoie automatiquement une information à l'objet *coll* ou *preset*, lui permettant de charger les valeurs suivantes pour l'ensemble des paramètres. Il est à noter que l'interface des plug-ins de traitement audio-numériques *GRM-Tools* propose ce type de fonctionnalité à travers le *Super Slider* [Favreau1998].

Conclusion

Nous avons présenté deux outils logiciels qui, si le premier peut être considéré comme un outil de composition et d'interprétation alors que le second n'est qu'un utilitaire, présentent des caractéristiques communes et ont été conçus dans le même but, celui de faciliter une approche transformationnelle de l'interprétation des musiques électroniques. Ce terme fait écho à la réflexion élaborée par Horacio Vaggione dans le cadre de la composition musicale [Vaggione1996].

En effet, chacun de ces outils propose, moyennant l'écriture au préalable d'états structurels ou sonores de dériver ceux-ci, que l'on peut considérer comme des classes, en autant de nouvelles sous-classes que désiré, étant donné que ces dérivations se font par le biais de processus d'interpolation qui, s'ils sont bien entendu discrets, proposent une idée du continu dans le cadre des variations structurelles.

Références bibliographiques :

[Courribet2002] Benoît Courribet [2002], *Electronica, aspects esthétiques, technologiques et perceptifs*, Mémoire du DEA Arts de la Scène et du Spectacle (Musique) sous la direction d'Horacio Vaggione, Paris, CICM-Université Paris 8

[Reich1974] Steve Reich [1974], *Ecrits et entretiens sur la musique*. Christian Bourgois Editeur (édition française 1981)

[Roads2002] Curtis Roads [2002], *Microsound*, Cambridge, MIT Press

[Favreau1998] Emmanuel Favreau [1998], *Les outils de traitement GRM Tools*. Actes des Journées d'Informatique Musicale 1998, CNRS-LMA Marseille.

[Vaggione1996] Horacio Vaggione [1996], *Vers une approche transformationnelle en C.A.O.* Actes des Journées d'Informatique Musicale 1996