



# Mono-Replay, une application pour animer la musique sur support : conception, développement et évaluation

Thomas Lucas

## ► To cite this version:

Thomas Lucas. Mono-Replay, une application pour animer la musique sur support : conception, développement et évaluation. Interface homme-machine [cs.HC]. Sorbonne Université, 2022. Français. NNT : 2022SORUS149 . tel-03813523

**HAL Id: tel-03813523**

**<https://theses.hal.science/tel-03813523>**

Submitted on 13 Oct 2022

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

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



École doctorale de  
Sciences mécaniques, acoustique, électronique et robotique de Paris

# Mono-Replay, une application pour animer la musique sur support : conception, développement et évaluation

THÈSE  
pour obtenir le grade de  
DOCTEUR DE SORBONNE UNIVERSITÉ  
Présentée par :

**Thomas LUCAS**

**Soutenue le 13 mai 2022 devant le jury composé de :**

<i>Rapportrice :</i>	Myriam DESAINTE-CATHERINE	Professeure, Université de Bordeaux
<i>Rapporteur :</i>	Pierre COUPRIE	Professeur, Université d'Évry
<i>Directeur :</i>	Christophe D'ALESSANDRO	Directeur de recherche, CNRS
<i>Examineur :</i>	Jean-Luc ZARADER	Professeur, Sorbonne Université
<i>Examinatrice :</i>	Anne SÈDES	Professeure, Université de Paris 8
<i>Examineur :</i>	Alain BONARDI	Maître de conférence HDR, Université de Paris 8
<i>Examineur :</i>	Atau TANAKA	Professeur, Goldsmiths University of London
<i>Invité :</i>	Serge DE LAUBIER	Directeur artistique, PUCe MUSE





## Remerciements :

Tout d'abord je souhaite remercier Alain Bonardi qui m'a motivé à reprendre mes études lors d'un échange en 2016. Sans cette rencontre je n'aurais jamais pu aboutir à ce travail de thèse.

Ensuite je remercie mon directeur de thèse Christophe d'Alessandro pour la richesse de son encadrement et la qualité des échanges que nous avons eus durant ces trois années. Je remercie aussi mon co-encadrant Serge de Laubier pour son expertise musicale et technique.

Je remercie les membres du jury qui ont lu avec attention mon travail et y ont apporté des retours très constructifs. Je remercie particulièrement Pierre Couprie et Myriam Desainte Catherine qui ont écrit un rapport synthétique et très éclaircissant sur ce travail.

Je remercie le centre de création visuelle PUCE MUSE qui a financé cette thèse ainsi que l'école doctorale SMAER qui m'a permis de la réaliser dans de bonnes conditions.

Je remercie l'Institut  $\delta$ 'Alembert qui m'a permis de concrétiser de magnifiques projets durant cette thèse. Un très grand merci à Sandrine Bandeira qui tout mis en œuvre pour que nous puissions réaliser des concerts dans des lieux fantastiques. Grâce à elle, je ne perçois plus le campus de Jussieu de la même façon :). Je tiens aussi à la remercier pour son soutien moral et son impressionnant travail administratif pour que la fin de ma thèse se déroule dans les meilleures conditions.

Un grand merci à Simona Otarasanu qui est toujours à l'écoute des étudiants de l'institut  $\delta$ 'Alembert. Grâce à son accueil et sa bienveillance, je me suis senti appartenir à une nouvelle famille.

Je remercie les musiciens Nicolas Rivière et Paul Denis Kingue Majole (Aka Pehoz) qui ont utilisé mon logiciel pour composer des musiques que j'apprécie énormément. Merci à Damien Bestard qui a organisé bénévolement et filmé avec talent les deux concerts qui ont eu lieu dans le cadre de cette thèse. J'en suis très reconnaissant.

Merci à Grégoire Locqueville pour les nombreux échanges scientifiques, les discussions philosophiques, les sorties entre amis et son écoute précieuse lors des moments de doutes. Merci à Priscilla, Daniel, Amine, Sarah, Patricia, Guiseppe, Camilla et tout·e·s les doctorant·e·s de l'Institut  $\delta$ 'Alembert avec qui j'ai passé de très bons moments. Un grand merci à mes ami·e·s et à ma famille qui m'ont donné énormément de soutien pour que je garde confiance en moi jusqu'au bout.

Merci à Jessie d'avoir toujours eu les bons mots quand mes questions existentielles de doctorant prenaient le dessus sur notre vie de couple. Cette thèse, je l'ai terminée

grâce à toi :). Enfin merci à ma fille Billie qui est arrivée quelques jours avant ma soutenance. Sans toi je n'aurais jamais été aussi fier de ce travail aujourd'hui.

**Résumé :** Animer la musique sur support consiste à interpréter, à l'aide d'un dispositif de contrôle gestuel, une œuvre enregistrée sur un média (disque vinyles, bande magnétique, fichier audionumérique, fichier midi). Le musicien peut ainsi proposer un parcours de lecture différent de celui fixé lors de l'enregistrement en modifiant, par le geste, la morphologie temporelle et spectrale de l'œuvre originale. Cette thèse décrit la conception, le développement et l'évaluation de Mono-Replay, une application pour animer la musique sur support audionumérique. Dès les débuts de l'enregistrement du son, le phonographe d'Edison a été envisagé, au-delà de la simple reproduction de l'œuvre fixée, comme un nouvel instrument pour l'expression sonore (la poésie phonographiste d'Apollinaire et Barzun). Depuis, de nombreux dispositifs permettant de transformer et de jouer un son enregistré de manière expressive ont été inventés. Qu'ils soient analogiques, comme la platine vinyle ou numérique comme l'échantillonneur (sampler), ils permettent de contrôler en direct l'évolution des paramètres dynamiques et spectraux et d'un indice temporel correspondant à la position lue dans l'enregistrement. Le calcul de cet indice est déterminé par un geste capté en continu, comme dans le cas d'une performance de scratch, ou par un geste capté de façon discrète, comme dans le cas d'une performance de Finger Drumming. Mono-Replay propose et intègre plusieurs méthodes pour contrôler l'indice temporel. Ces méthodes tiennent compte d'éléments structurels de l'enregistrement, comme la position des battues d'une pulsation, la position des transitoires d'attaques d'une note, le début d'une mesure ou le début d'une structure musicale. Diverses méthodes sont intégrées dans Mono-Replay afin d'indexer automatiquement ou manuellement ces éléments structurels. Le logiciel permet de varier le déroulement de l'enregistrement source, de transposer sa hauteur, son enveloppe spectrale, de filtrer son spectre, de jouer avec sa dynamique et de lui appliquer toutes sortes d'effets audionumériques. Chaque paramètre de contrôle est défini via le protocole OSC, qui est compatible avec la plupart des interfaces de contrôle gestuel actuelles. Le musicien peut ainsi utiliser une interface gestuelle externe pour configurer l'instrument et jouer en direct avec les paramètres spectraux et temporels de l'enregistrement. Toute œuvre sur support audionumérique peut être animée, après avoir été enrichie par des indices temporels. Ces indices temporels marquent le contenu de l'œuvre source, en fonction des intentions musicales particulière du compositeur. Les données de contrôle gestuel et les paramètres de contrôle du moteur audio doivent être défini à l'aide du mapping gestuel pour constituer l'instrument. Le corpus musical étudié comprend des enregistrements du *Trio n°2 op. 100 pour piano et cordes* de Franz Schubert, de la *Gymnopédie n°1* pour piano d'Érik Satie, une pièce électroacoustique *Accidents/Harmoniques* de Bernard Parmegiani et un enregistrement multipiste de *Supersition* de Stevie Wonder. Ce répertoire éclectique démontre l'application de Mono-Replay à différents styles musicaux. Des interfaces gestuelles comme le Méta-instrument 4, la manette de jeu dual Analog 4, le mini-méta, des contrôleurs MIDI (fader, potentiomètre, pads) ont servi d'interfaces gestuelles expressives. Mono-Replay a été évalué par la pratique grâce à des commandes de création à 3 compositeurs de musique électronique. Ces créations sont accompagnées d'entretiens semi-directifs avec les artistes, qui ont découvert puis intégré le logiciel dans leur environnement habituel de composition et de performance. Ces entretiens retranscrits et analysés permettent de souligner l'apport créatif de Mono-Replay dans une démarche comparative avec la boîte à outils d'instruments et de logiciels audionumérique de chacun. Parmi les trois musiciens interrogés deux ont utilisé Mono-Replay dans un contexte de création en studio et dans un contexte de concert et ont abouti à des pièces enregistrées et jouées en public, validant ainsi des utilisations musicales possibles de notre application. Cette thèse démontre la variété et l'efficacité du concept d'indice temporel pour l'animation de la musique sur support audionumérique.

**Mots clés :** Système musical interactif, indexation audio, mappings gestuels



**Abstract :** Animating music on a medium consists in interpreting, with the help of a gestural control device, a musical piece recorded on a medium (vinyl record, magnetic tape, digital audio file, midi file). The musician can thus propose a reading path different from the one fixed during the recording by modifying, through gesture, the temporal and spectral morphology of the original work. This thesis describes the design, development and evaluation of Mono-Replay, an application for animating music on digital audio media. Since the beginning of sound recording, Edison's phonograph has been considered, beyond the simple reproduction of the fixed work, as a new instrument for sound expression (the phonograph poetry of Apollinaire and Barzun). Since then, many devices allowing to transform and play a recorded sound in an expressive way have been invented. Whether they are analog, such as the vinyl turntable, or digital, such as the sampler, they allow the evolution of dynamic and spectral parameters and a temporal index corresponding to the position read in the recording to be controlled live. The calculation of this index is determined by a gesture captured continuously, as in the case of a scratch performance, or by a gesture captured in a discrete way, as in the case of a Finger Drumming performance. Mono-Replay proposes and integrates several methods to control the temporal index. These methods take into account structural elements of the recording, such as the position of beats of a pulse, the position of transients of attacks of a note, the beginning of a measure or the beginning of a musical structure. Various methods are integrated in Mono-Replay in order to index automatically or manually these structural elements. The software allows to vary the course of the source recording, to transpose its pitch, its spectral envelope, to filter its spectrum, to play with its dynamics and to apply all kinds of digital audio effects. Each control parameter is defined via the OSC protocol, which is compatible with most current gesture control interfaces. The musician can use an external gesture interface to configure the instrument and play live with the spectral and temporal parameters of the recording. Any work on digital audio can be animated, after being enriched with temporal cues. These temporal cues mark the content of the source work, according to the particular musical intentions of the composer. The gestural control data and the control parameters of the audio engine must be defined using gestural mapping to constitute the instrument. The musical corpus studied includes recordings of Franz Schubert's *Trio No. 2 op. 100 for piano and strings*, Erik Satie's *Gymnopédie No. 1* for piano, an electroacoustic piece *Accidents/Harmoniques* by Bernard Parmegiani and a multitrack recording of Stevie Wonder's *Supersition*. This eclectic repertoire demonstrates the application of Mono-Replay to different musical styles. Gestural interfaces such as the Meta-Instrument 4, the Analog 4 dual joystick, the mini-meta, MIDI controllers (fader, potentiometer, pads) were used as expressive gestural interfaces. Mono-Replay was evaluated by practice thanks to creation orders to 3 electronic music composers. These creations are accompanied by semi-directive interviews with the artists, who discovered and integrated the software in their usual environment of composition and performance. These transcribed and analyzed interviews allow underlining the creative contribution of Mono-Replay in a comparative approach with the toolbox of instruments and digital audio software of each. Among the three musicians interviewed, two used Mono-Replay in a studio creation context and in a concert context and resulted in pieces recorded and played in public, thus validating possible musical uses of our application. This thesis demonstrates the variety and efficiency of the temporal index concept for the animation of music on digital audio media.

**Key words :** Human machine interfaces, new interfaces for musical expression, gesture mappings



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>13</b>
1.1	Animer la musique sur support . . . . .	15
1.2	Plusieurs méthodes pour contrôler par le geste le déroulement d'une musique sur support . . . . .	18
1.2.1	Sampling et Plunderphonics . . . . .	18
1.2.2	Le scratch . . . . .	20
1.2.3	"Active Listening Interface" . . . . .	22
1.2.4	Le Méta-Piano . . . . .	24
1.2.5	La famille des instruments numériques Voks . . . . .	25
1.3	Trois concepts clés pour développer des interactions basées sur la manipulation d'un enregistrement . . . . .	27
1.3.1	Le substitut gestuel . . . . .	28
1.3.2	L'énaction . . . . .	32
1.3.3	Le concept de re-enactment . . . . .	33
1.4	Contexte de travail : PUCE MUSE et les Mono . . . . .	34
1.5	Bilan et Enjeux . . . . .	37
<b>2</b>	<b>Concevoir une interaction musicale basée sur la manipulation d'un enregistrement</b>	<b>41</b>
2.1	Les applications audio interactives . . . . .	42
2.2	Indexation . . . . .	45
2.2.1	Indexation de transitoire d'attaque . . . . .	48
2.2.2	Indexation de pulsation . . . . .	58
2.2.3	Indexation manuelle . . . . .	63
2.3	Contrôle temporel . . . . .	64



2.3.1	Contexte . . . . .	65
2.3.1.1	Expression mathématique des modes de contrôle temporel . . . . .	67
2.3.2	Contrôle temporel continu . . . . .	68
2.3.3	Contrôle temporel discret . . . . .	69
2.3.3.1	Mode asynchrone linéaire . . . . .	72
2.3.3.2	Mode asynchrone linéaire continu . . . . .	73
2.3.3.3	Mode synchrone linéaire . . . . .	74
2.3.3.4	Mode synchrone parabolique . . . . .	76
2.4	Contrôle spectral . . . . .	77
2.4.1	Contrôle continu linéaire . . . . .	78
2.4.2	Contrôle continu linéaire par morceaux . . . . .	78
2.4.3	Contrôle spectral discret . . . . .	80
2.4.4	Contrôle discret par calcul d'enveloppe . . . . .	80
<b>3</b>	<b>Mono-Replay</b>	<b>83</b>
3.1	Architecture générale . . . . .	85
3.2	Choix des langages de programmation . . . . .	88
3.2.1	PUCE MUSE et Max . . . . .	88
3.2.2	Accéder aux algorithmes de MIR dans Max à l'aide de Node.js . . . . .	89
3.3	Système d'indexation assistée . . . . .	90
3.3.1	Paramétrer les algorithmes de <i>essentia.js</i> dans Max/MSP . . . . .	91
3.3.1.1	SuperFluxExtractor . . . . .	91
3.3.1.2	RhythmExtractor2013 . . . . .	94
3.3.2	Trois niveaux d'indexation . . . . .	95
3.4	Gestion des modes de contrôle temporel . . . . .	99
3.4.1	Scrub, Speed et TapTempo . . . . .	101
3.4.1.1	Scrub . . . . .	101
3.4.1.2	Speed . . . . .	102
3.4.1.3	TapTempo . . . . .	105
3.4.2	Naviguer dans l'enregistrement selon trois niveaux d'indexation . . . . .	109
3.5	Gestion des transformations spectrales . . . . .	110
3.5.1	Contrôles des paramètres SuperVP . . . . .	111
3.5.2	Éditeur d'enveloppe de transformation . . . . .	112

3.6	Gestion des mémoires . . . . .	113
3.6.1	Pattern storage . . . . .	113
3.6.2	Interpolation de mémoires . . . . .	113
3.7	Création de projet . . . . .	115
3.8	Dictionnaire OSC et mapping . . . . .	117
<b>4</b>	<b>Cas d’usage de Mono-Replay</b>	<b>119</b>
4.1	Contrôler Mono-Replay à l’aide d’une manette de jeu vidéo . . . . .	120
4.1.1	Ateliers Mono-Replay . . . . .	123
4.1.2	Animer un enregistrement multipiste du morceau <i>Superstition</i> par Stevie Wonder . . . . .	125
4.1.3	Création du morceau <i>Bar Mono-Replay</i> par Pehoz . . . . .	127
4.2	Contrôler Mono-Replay à l’aide du Méta-Instrument 4 . . . . .	129
4.2.1	Le Méta-Instrument 4 . . . . .	129
4.2.2	Animer un enregistrement du <i>Trio n°2</i> de Franz Schubert par le Beaux Arts Trio en battant une pulsation . . . . .	131
4.2.2.1	Étiquetage . . . . .	133
4.2.2.2	Mapping . . . . .	135
4.2.3	Animer <i>Accidents/Harmoniques</i> de Bernard Parmegiani en contrôlant de façon continue la position de lecture . . . . .	136
4.2.3.1	Étiquetage . . . . .	137
4.2.3.2	Mapping . . . . .	137
4.3	Animer deux pièces de musiques électroniques composées par Nicolas Rivière . . . . .	138
4.3.1	<i>SUPERNOME DIVA 2 - Contemplation du vide</i> . . . . .	139
4.3.2	<i>SUPERNOME DIVA 2.5 - Anatomie du vide</i> . . . . .	139
4.3.3	Dispositif et mappings . . . . .	141
<b>5</b>	<b>Élaboration et analyse d’entretiens semi-directifs visant à évaluer qualitativement le logiciel Mono-Replay</b>	<b>147</b>
5.1	Présentation des sujets . . . . .	149
5.1.1	Nicolas Rivière . . . . .	149
5.1.2	Paul Denis Kingue-Manjole (Pehoz) . . . . .	150
5.1.3	Basile Chassaing . . . . .	151
5.2	L’entretien exploratoire . . . . .	151

5.3	Élaboration du guide d'entretien . . . . .	155
5.4	À propos du contrôle temporel . . . . .	158
5.5	Au sujet du ressenti global de l'expérience . . . . .	159
5.6	À propos des interfaces gestuelles . . . . .	164
5.7	À propos de l'interface graphique . . . . .	166
5.8	Les perspectives de Mono-Replay selon les musiciens . . . . .	168
5.9	Synthèse . . . . .	170
<b>6</b>	<b>Conclusion</b>	<b>173</b>
6.1	Bilan . . . . .	173
6.2	Perspectives . . . . .	177
6.2.1	Vers une évaluation quantitative . . . . .	177
6.2.2	Intégrer Mono-Replay dans d'autres formes de création . . . .	178
6.2.3	Rhythmic Control Unit : une librairie de fonctions pour le contrôle temporel . . . . .	179
6.2.4	Développer un projet pédagogique autour du contrôle temporel	180
	<b>Bibliographie</b>	<b>183</b>
	<b>Annexes</b>	<b>190</b>
<b>A</b>	<b>Article de la conférence Journées d'Informatique Musicale 2020</b>	<b>193</b>
<b>B</b>	<b>Article de la conférence New Interfaces for Musical Expression 2021</b>	<b>201</b>
<b>C</b>	<b>Manuel du logiciel Mono-Replay</b>	<b>221</b>
<b>D</b>	<b>Transcription de l'entretien semi-directif avec Nicolas Rivière</b>	<b>247</b>
<b>E</b>	<b>Transcription de l'entretien avec Paul-Denis Kingue Manjole</b>	<b>259</b>
<b>F</b>	<b>Transcription de l'entretien avec Basile Chassaing</b>	<b>269</b>
<b>G</b>	<b>Instruction de l'atelier SMAC pour créer un projet à l'aide de Mono-Replay</b>	<b>275</b>

H Dictionnaire OSC des variables de Mono-Replay	281
I Questionnaire d'évaluation portant sur l'ergonomie des logiciels Mono	289

# Chapitre 1

## Introduction

### Sommaire

---

<b>1.1</b>	<b>Animer la musique sur support . . . . .</b>	<b>15</b>
<b>1.2</b>	<b>Plusieurs méthodes pour contrôler par le geste le déroulement d'une musique sur support . . . . .</b>	<b>18</b>
1.2.1	Sampling et Plunderphonics . . . . .	18
1.2.2	Le scratch . . . . .	20
1.2.3	"Active Listening Interface" . . . . .	22
1.2.4	Le Méta-Piano . . . . .	24
1.2.5	La famille des instruments numériques Voks . . . . .	25
<b>1.3</b>	<b>Trois concepts clés pour développer des interactions basées sur la manipulation d'un enregistrement . . . . .</b>	<b>27</b>
1.3.1	Le substitut gestuel . . . . .	28
1.3.2	L'énaction . . . . .	32
1.3.3	Le concept de re-enactment . . . . .	33
<b>1.4</b>	<b>Contexte de travail : PUCE MUSE et les Mono . . . . .</b>	<b>34</b>
<b>1.5</b>	<b>Bilan et Enjeux . . . . .</b>	<b>37</b>

---

Depuis l'apparition des premières flûtes il y a 35000 ans, les instruments de musique n'ont jamais cessé d'évoluer. Les motivations de ces évolutions sont souvent liées à la volonté d'améliorer la facilité d'utilisation de l'outil. Leur mise en oeuvre est, quant à elle, indissociable des avancées techniques de l'époque. La révolution industrielle, par exemple, a permis une nette évolution de la lutherie traditionnelle occidentale [15]. Le développement des arts mécaniques, du commerce colonial et de

la métallurgie ont permis de créer de nouveaux instruments mais aussi d'améliorer les anciens. Ainsi le saxophone fut inventé et le système de clef Boehm fut adapté vers 1839 à la clarinette pour la rendre plus facile à jouer et plus juste.

L'arrivée de l'informatique a aussi révolutionné notre rapport au geste instrumental. Alors que les premiers logiciels de synthèse sonore n'étaient contrôlables qu'avec des claviers classiques, des centaines d'autres interfaces gestuelles furent conçues pour explorer de nouveaux gestes musicaux [56]. Aujourd'hui une communauté internationale de chercheurs travaillent sur les problématiques liées à ces systèmes que l'on appelle couramment NIME, en référence à la conférence New Interfaces for Musical Expression créée en 2001. Parmi ces interfaces on peut citer le Méta-Instrument [18] [14] conçu pour contrôler des algorithmes de spatialisation sonore de plusieurs enregistrements en temps réel ou encore le Karlax [38] qui prend la forme d'un bâton sur lequel une multitude de capteurs retranscrivent le geste du musicien.

La plupart de ces interfaces sont programmables et compatibles avec les logiciels de création dont elles deviennent, en quelque sorte, les manettes. Les "mapping" gestuels représentent les blocs algorithmiques qui lient les variables de ces interfaces aux contrôles du logiciel qu'elles pilotent. Ces algorithmes peuvent se décliner à l'infini et il est compliqué d'évaluer leur pertinence musicale. Certains imitent le fonctionnement d'instruments déjà existants comme le clavier MIDI mais d'autres, comme le Méta-instrument, proposent des façons uniques de contrôler les hauteurs, durées et autres paramètres d'une note.

Pour évaluer si ces nouveaux gestes sont musicalement pertinents, il n'existe pas réellement de méthode. C'est souvent depuis l'expérience subjective d'un utilisateur/musicien ou d'un spectateur que l'on parvient à comprendre si le geste musical associé au contrôle d'une transformation sonore a été suffisamment expressif. Certains musicologues nous offrent néanmoins des outils méthodologiques pointus pour décomposer le geste instrumental en plusieurs sous-fonctions [56] [9]. Ces outils peuvent nous guider dans notre démarche de réflexion sur la qualité "musicale" d'un geste qui permet d'interagir avec un système informatique, mais l'expérience subjective reste toujours la plus déterminante.

Dans notre travail, nous nous intéresserons à la qualité expressive de ce geste, pour le cas très précis des systèmes qui permettent de contrôler temporellement la lecture d'un enregistrement. Le but est de proposer une méthode scientifique pour concevoir un système dédié au contrôle temporel dans lequel s'intègrent toutes les

fonctionnalités nécessaires pour qu'il soit un parfait outil expressif.

## 1.1 Animer la musique sur support

*Comment concevoir un geste expressif pour jouer un enregistrement ?*

On appelle musique sur support toute musique issue d'un enregistrement physique sur lequel l'information audio nécessaire pour sa restitution sonore est contenue. Les premiers supports sont nés en même temps que les premiers systèmes capables de les créer comme le phonotaugraphe d'Édouard-Léon Scott de Martinville inventé en 1853 ou le phonographe de Thomas Edison, représenté figure 1.1, inventé en 1877 qui permirent pour la première fois de retranscrire les ondes sonores sur un support physique. Le phonotaugraphe inscrit les ondes sonores sur du papier et le phonographe sur de l'étain.

Bien qu'il fallût attendre les travaux du "Lawrence Berkeley National Laboratory" en 2008 pour pouvoir enfin entendre les ondes sonores retranscrites sur papier par le phonotaugraphe<sup>1</sup>, l'invention de Thomas Edison permit quant à elle de restituer dès 1877 le son préalablement gravé sur le cylindre d'étain. On comprend dès le début de ces inventions l'enjeu du choix du support utilisé pour contenir la donnée.

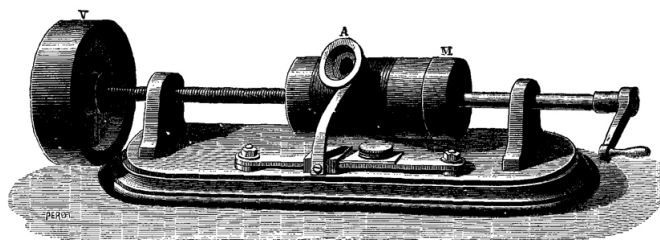


FIGURE 1.1 – Le phonographe ou machine parlante d'Edison, parut dans Nature, revue des sciences le 23 mars 1878

La pensée d'utiliser ces outils d'enregistrement et de diffusion comme des instruments de création artistique arriva quelques dizaines d'années plus tard avec notamment Apollinaire qui en enregistrant sa voix déclamant un poème

---

1. Le plus vieil enregistrement de voix humaine a été enregistré en 1860 par Léon Scott de Martinville. Il aura fallu attendre l'année 2010 pour qu'il soit enfin audible : <http://www.firstsounds.org/sounds/scott.php>

an 1914 vit dans le gramophone un outil pour réciter de manière simultanée plusieurs vers d'une poésie [3]. Cette volonté d'expérimenter de nouvelles formes de poésie était accompagnée à l'époque d'une intense recherche sur la représentation graphique de celle-ci. Le calligramme représenté figure 1.2 en donne un bon exemple. La mise en page du poème sous la forme d'un dessin redéfinit le sens de lecture de celui-ci.

Le compositeur Carol-Bérard pense quant à lui dès 1929, soit 20 ans avant l'existence de la musique concrète, qu'en regroupant et associant attentivement des bruits enregistrés sur un gramophone on pourrait obtenir des symphonies reconnaissables à l'oreille [3].

Le parti pris de composer avec des fragments d'enregistrements et de les utiliser comme des "éléments prélevés sur le donné sonore expérimental"<sup>2</sup> arriva avec la musique concrète. Schaeffer fonda en 1951 le Groupe de Recherche de Musique concrète (GRM) où toutes sortes d'expérimentations autour des concepts liant création artistique, enregistrement et interaction virent le jour. Le support musical devient alors un véritable matériel de création sonore avec lequel on expérimente, on crée et on interagit.

Parmi les systèmes inventés par ce groupe de recherche on peut citer les Phonogènes qui, conçus par l'ingénieur Jacques Poullin, permettent de varier la vitesse de lecture d'un enregistrement et sa hauteur par intervalle de demi-ton. La deuxième génération de ces instruments permet d'effectuer des variations continues de vitesse. On peut ainsi glisser d'une hauteur à l'autre sans discontinuité. La dernière génération de ces instruments permet de modifier la vitesse de lecture sans transposer la hauteur [8]. Les inventions du GRM s'inscrivent dans la continuité des premières pensées artistiques de Carol-Bérard et Appolinaire face à l'invention d'Edison.

Aujourd'hui les outils permettant de contrôler de manière expressive le déroulement temporel d'enregistrements se sont largement popularisés. Parmi les plus connus on peut nommer la platine de DJ, le sampler numérique ou encore les Digital Audio Workstation. Chacun de ces systèmes propose des gestes adaptés à l'utilisation particulière que l'on fait d'eux. Dans le cas des samplers

---

2. citation extraite du livre *À la recherche d'une musique concrète* paru en 1952 chez Éditions du Seuil dans la collection Pierre vives



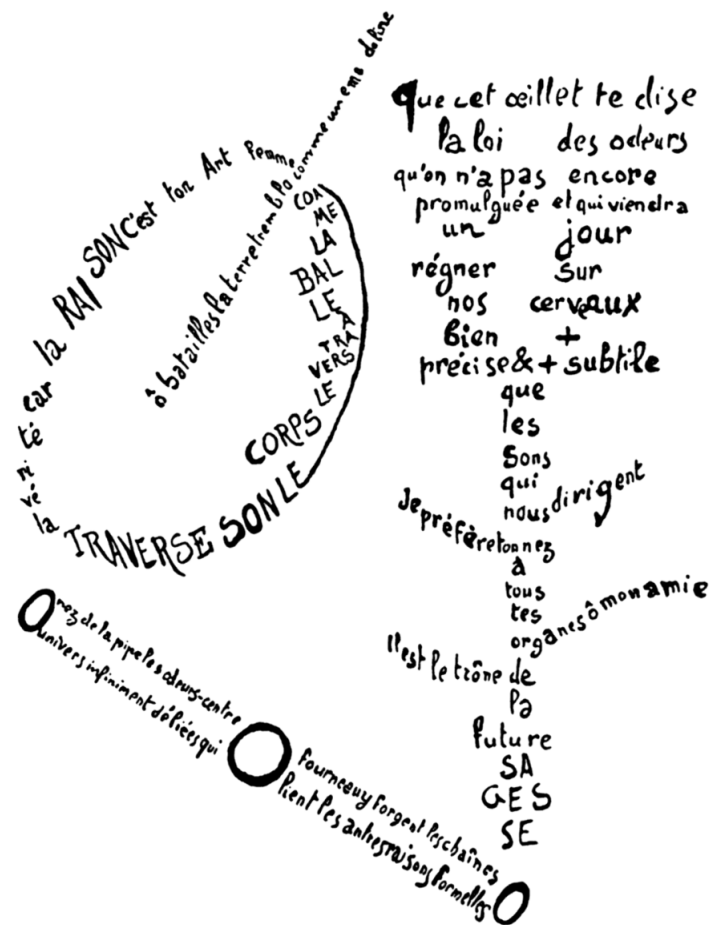


FIGURE 1.2 – *La mandoline, l'œillet et le bambou*, Guillaume Apollinaire, 1914 - 1915. Les calligrammes sont des poèmes dont la mise en page s'apparente à des dessins. Une telle disposition redéfinie le sens de lecture.

MPC représenté sur la figure 1.4 par exemple, l'interface gestuelle se constitue d'une matrice de touches grâce auxquelles l'utilisateur peut déclencher simultanément la lecture de différentes portions préalablement définies d'un même enregistrement. La lecture se lance lorsque l'on enfonce une touche et s'arrête lorsqu'on la relâche. Il existe quelques fonctionnalités qui permettent de lire l'échantillon en boucle et de paramétrer le début et la fin du segment audio, mais la lecture de l'échantillon se fait toujours selon le même scénario : en appuyant sur une touche on déclenche la lecture du fichier d'une position initiale vers une position finale selon une progression linéaire. Si on souhaite changer de scénario pour par exemple contrôler de manière directe la position de lec-

ture il sera nécessaire de changer de système. Le logiciel que nous présentons ici explore ce concept de scénario. Il a été conçu pour permettre d'utiliser un nombre important de méthodes pour contrôler en direct la position de lecture d'un enregistrement.

Dans ce travail de recherche on désigne par mode de contrôle temporel toutes les fonctions qui prennent en entrée un geste capté et donnent en sortie une position de lecture dans l'enregistrement. Dans le cas de la platine de DJ, par exemple, le mode de contrôle temporel correspond au déplacement de la tête de lecture sous l'action du musicien sur le vinyle. Les mouvements du diamant vont suivre la forme d'onde gravée sur le vinyle et la vitesse de son mouvement va déterminer la hauteur du son.

On désigne ici par indexation l'étape qui consiste à placer des repères temporels sur un enregistrement en vue de le manipuler. Dans le cas du sampler par exemple, cette étape permet d'attribuer à chaque bouton de l'interface gestuelle visible figure 1.4 une portion de l'enregistrement qui sera lue.

## 1.2 Plusieurs méthodes pour contrôler par le geste le déroulement d'une musique sur support

### 1.2.1 Sampling et Plunderphonics

Le sampling désigne la pratique musicale qui consiste à utiliser des portions d'un enregistrement déjà existant pour créer un nouveau morceau. Le support musical est utilisé ici comme un matériau dans lequel l'artiste vient chercher, agencer et transformer des textures, des motifs rythmiques ou encore des hauteurs au service d'un discours musical qu'il souhaite développer. Dans la musique hip-hop le sampling est une pratique très courante. À l'aide d'un échantillonneur ou sampler, comme celui représenté figure 1.4, Le "beatmarker" déclenche la lecture d'une portion d'un enregistrement afin de l'inclure dans une orchestration instrumentale plus complète. Ce segment est souvent transformé et joué en boucle. Dans le morceau *Kick in the Door*<sup>3</sup> produit par

---

3. Le morceau *Kick in the Door* de Notorious BIG est conçu à partir d'un segment de l'enregistrement du morceau *I put a spell on you* interprété par Screaming Jaw Hawkins : [https://www.youtube.com/watch?v=aNwvHEme\\_JE](https://www.youtube.com/watch?v=aNwvHEme_JE)

DJ Premier et interprété par Notorious BIG, un segment de l'enregistrement de *I put a spell on you* interprété par Screaming Jaw Hawkins est joué en boucle. Une orchestration s'additionne pour développer un discours musical Hip-Hop et accentuer les temps forts. Cette boucle sert de base instrumentale pour que le rappeur y développe son discours musical.

Dans le cas des "plunderphonics" [42] de John Oswalds, le support musical est utilisé comme matériel exclusif pour la composition. Le compositeur choisit des segments sonores issus d'enregistrements célèbres de la musique pop pour construire son discours musical<sup>4</sup>. Contrairement au sampling le but ici est d'identifier clairement l'enregistrement dont sont issus les segments pour accentuer l'effet d'entendre quelque chose de nouveau depuis quelque chose de déjà existant. Les échantillonneurs ou sampler sont des outils spécialement

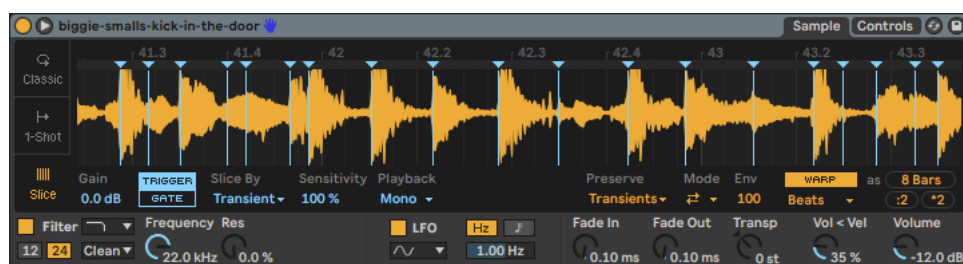


FIGURE 1.3 – Interface graphique de l'échantillonneur intégré à Ableton Live. Un algorithme de détection automatique de transitoire d'attaque y est intégré.

conçus pour ce type de réalisation. Popularisés en 1979 par la commercialisation du synthétiseur Fairlight CMI ils permettent de placer des ancrs temporelles sur un enregistrement afin de déclencher la lecture de différentes portions de celui-ci. Le placement des ancrs se fait la plupart du temps manuellement mais peut-être assisté par des algorithmes de détection de transitoires pour déterminer automatiquement le début des notes dans un enregistrement. C'est le cas par exemple de l'échantillonneur simplier qui est disponible dans le logiciel Ableton Live et dont la représentation de l'interface est disponible figure 4.16. Les échantillonneurs sont des outils très complets pour indexer le contenu rythmique d'un enregistrement. Néanmoins ils proposent souvent une faible variété de mode de contrôle temporel. La lecture de l'échantillon s'effectue

4. Le morceau DAB de John Oswald est conçu exclusivement d'échantillons sonores issus du morceau Bad de Michael Jackson : <https://www.youtube.com/watch?v=Rhck5dVP9AA>

la plupart du temps de façon linéaire entre le début et la fin d'une portion à indexer. Il existe des alternatives permettant de lire en boucle ou à l'envers un segment audio mais la vitesse de lecture demeure systématiquement constante du début à la fin. Introduire plus de diversité de mode de contrôle temporel au sein d'un même système offre de nouvelles possibilités créatives. C'est ce que nous souhaitons démontrer dans ce travail de recherche.



FIGURE 1.4 – Le sampler AKAI MPC permet de déclencher la lecture de segments audio issus d'un enregistrement en tapant sur des touches.

### 1.2.2 Le scratch

Le scratch désigne la pratique musicale qui consiste à contrôler de façon synchronisée la vitesse de lecture et le bouton de volume d'une platine vinyle. Ici le contrôle temporel est continu et consiste à déplacer la tête de lecture sur le sillon du vinyle. On fait ainsi varier la vitesse de lecture et la hauteur du son rejoué. Les platines vinyles sont souvent reliées à des tables de mixage sur lesquelles figure un fader de volume qui permet de couper et remettre le son. Le célèbre DJ philippin DJ Q-Bert recense dans un DVD [49] plus d'une centaine de techniques de scratch qui combinent variation de vitesse de lecture et modulation du volume. Cette diversité de techniques montre à quel point il est possible de développer plusieurs gestes expressifs autour d'un même mode de contrôle temporel. Si on imagine un système permettant d'en intégrer une grande variété alors les possibilités gestuelles deviennent immenses.

Pour aider le DJ à positionner la tête de lecture sur la bonne portion de l'enregistrement plusieurs techniques d'indexation sont possibles. Dans les logiciels d'émulation de platine tel que Serato ou Traktor il est possible de poser manuellement ou automatiquement ces repères temporels directement sur la forme d'onde de l'enregistrement. Comme on le voit dans la figure 1.6 l'interface graphique de ces systèmes ressemblent beaucoup à l'interface des samplers. Une fois ces positions déterminées, le DJ s'en sert pour positionner la tête de lecture sur ceux-ci à l'aide d'un contrôleur MIDI. Avant l'apparition des logiciels d'émulation et des contrôleurs MIDI cette indexation pouvait être réalisée directement sur le vinyle à l'aide d'autocollants directement déposés dessus comme représenté figure 1.5. Tout comme pour le sampling, le scratch



FIGURE 1.5 – Certains DJ placent des repères temporels directement sur le vinyle pour indexer les éléments de l'enregistrement qu'ils souhaitent manipuler. (source : <https://www.youtube.com/watch?v=tr3ftsCVXhc>)

nécessite une indexation préalable de l'enregistrement. Cette indexation aide ensuite le DJ à positionner la tête de lecture sur la portion de l'enregistrement qu'il souhaite rejouer. Les méthodes d'indexations sont les mêmes que celles utilisées dans les échantillonneurs, néanmoins le mode de contrôle temporel est très différent. D'un côté on contrôle de façon discrète la lecture d'une portion de l'enregistrement en appuyant sur la touche d'un pad et de l'autre on contrôle de manière continue la position de lecture à l'aide d'une platine sur laquelle tourne un vinyle. Serait-il possible de créer un système qui permettrait d'indexer un enregistrement de la même façon que ces deux systèmes tout en intégrant leurs modes de contrôle temporel respectifs ? Quel serait l'intérêt

créatif d'un tel dispositif ?

Le logiciel Skip proof [30] a été conçu pour étudier les gestes utilisés par les DJ lors des performances de scratches. Il permet de réaliser des scratchs sur des fichiers numérisés en utilisant des techniques d'étirement temporel. Le logiciel intègre certaines des techniques recensées dans DJ Q-bert 2007 et permet à l'utilisateur de les appliquer directement sur l'échantillon sonore sans avoir de geste continu à effectuer pour le générer. On parle alors de contrôle discret.

Ce logiciel est intéressant pour nous car il permet d'effectuer un contrôle temporel discret et continu de la position de lecture d'un échantillon.

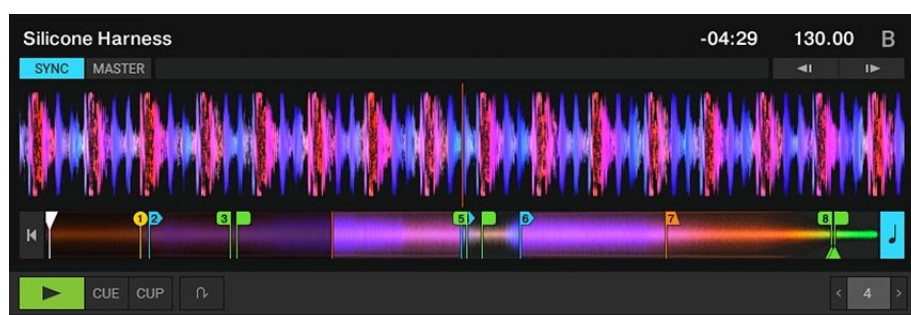


FIGURE 1.6 – L'interface du logiciel Traktor permet de positionner des repères temporels sur la forme d'onde d'un enregistrement. Le Dj peut ensuite se servir de ces repères pour notamment scratcher.

### 1.2.3 "Active Listening Interface"

Au début des années 2000, la démocratisation de l'accès haut débit à internet permet à de nombreuses personnes d'écouter, de partager et de stocker des fichiers audio sur les serveurs web. Ainsi les bases de données s'accroissent de façon exponentielle et il devient urgent de réfléchir à de nouvelles façons de les organiser. C'est alors qu'une communauté de chercheurs, composée de musicologues, de développeurs informatiques et d'ingénieurs en traitement du signal, réfléchit à de nouvelles façons de renseigner un fichier audio. Ils font ainsi naître un domaine de recherche interdisciplinaire appelé *Musical Information Retrieval* (MIR). Ces personnes élaborent des méthodes de traitement du signal permettant de récupérer des informations directement depuis le contenu audio d'un fichier sans passer par une phase préalable d'écoute. Cela permet de récupérer rapidement des informations tel que l'énergie globale, l'évolution

harmonique ou encore les variations de tempo d'un enregistrement sur toute une base de données [48].

Le chercheur Masataka Goto travaille sur des méthodes de ce type et les applique à la conception d'outils informatiques qui proposent des interactions inédites entre un auditeur et une application audio interactive [27] [24] [25] [26]. Il nomme ces applications "Active Listening Interfaces". Le but premier de ces applications est de sortir l'utilisateur d'un système d'écoute de son simple rôle d'auditeur en le faisant interagir activement avec le déroulement de la musique. Par "activement" Masataka Goto désigne le caractère physique de l'interaction avec le système. Cela va, selon lui, augmenter l'expérience auditive de l'utilisateur.

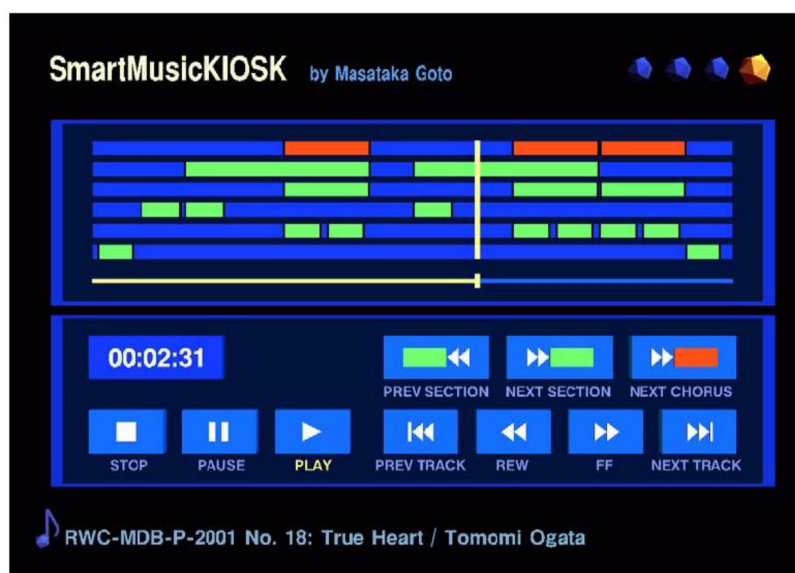


FIGURE 1.7 – Interface graphique de *SmartMusicKIOSK* de Masataka Goto est un logiciel qui permet à un utilisateur de naviguer dans un enregistrement compte tenu de la structure musicale de celui-ci. Ainsi l'utilisateur peut se déplacer directement au début d'un refrain ou d'un couplet.

SmartMusicKIOSK dont l'interface est présentée figure 1.7 fait partie de ces systèmes. Il permet à un utilisateur de naviguer dans un enregistrement compte tenu de la structure musicale de celui-ci. Grâce à un système d'indexation intelligent, le système renseigne automatiquement le début des refrains et des couplets d'un enregistrement. On peut alors positionner la lecture de l'enregistrement directement sur ces ancres, pour ne lire qu'une partie de celui-ci.



Si l'on observe ce système de façon globale, on constate qu'il est composé d'un système d'indexation, d'une interface graphique grâce à laquelle l'utilisateur contrôle le logiciel et d'un moteur audio qui génère le son lu. Ici le mode de contrôle rythmique est très basique, l'utilisateur indique de façon discrète des positions temporelles dans un enregistrement. Malgré sa simplicité, ce système utilise le même paradigme de contrôle rythmique que des systèmes permettant des contrôles beaucoup plus complexes.

#### 1.2.4 Le Méta-Piano

Le Méta-Piano est un instrument développé par Jean Haury [31] permettant de contrôler en direct l'articulation, le tempo, le rythme et le phrasé dynamique et agogique d'un morceau de piano. Grâce à cet instrument le musicien peut se focaliser directement sur l'interprétation du morceau sans passer par l'apprentissage des relations mélodiques, harmoniques et contrapuntiques entre les notes.

Le Méta-Piano, qui est représenté figure 1.8, est constitué d'un clavier MIDI, d'un système pour annoter une partition MIDI et d'un moteur audio synthétisant un son de piano en temps réel. À chaque fois qu'une touche est enfoncée une nouvelle unité rythmique du fichier MIDI est jouée. L'unité rythmique peut être constituée d'une note ou d'un accord. La combinaison des touches enfoncées sur le clavier MIDI permet de définir l'articulation du passage d'une unité rythmique à l'autre. Par exemple si le musicien appuie sur une touche alors qu'une autre touche est déjà enfoncée les notes présentes dans la première unité seront liées aux notes présentes dans la seconde.

Contrairement aux autres systèmes présentés dans cet état de l'art le Méta-Piano n'indique pas un indice temporel correspondant à la position lue dans l'enregistrement. Ici le support "animé" n'est pas un enregistrement. Il s'agit d'un fichier MIDI sur lequel la progression mélodique et harmonique est renseignée. Le geste du musicien permet de définir la durée et l'intonation des éléments présents dans ce fichier.

Ce système est particulièrement intéressant pour notre travail car il permet de contrôler en direct l'avancement rythmique d'un morceau de piano. Ce mode de contrôle temporel fait partie de ceux que nous souhaitons intégrer





FIGURE 1.8 – Le Méta-Piano de Jean Haury [31] permet de contrôler le tempo, le rythme et le phrasé dynamique d’une partition MIDI pour piano.

dans notre logiciel. En nous inspirant du Méta-Piano nous avons conçu un mode permettant de contrôler en direct l’avancement rythmique d’un enregistrement de la Gymnopédie n°1 de Érik Satie. Nous présentons cette interaction dans la suite de ce manuscrit.

### 1.2.5 La famille des instruments numériques Voks

Les instruments Voks sont conçus pour la synthèse performative de voix. Ils permettent de contrôler en direct à l’aide d’interfaces gestuelles expressives la hauteur, l’effort vocal, la tension vocale, la taille du conduit vocal apparent, les composantes harmoniques et inharmoniques d’une voix enregistrée. Tout comme le système que nous souhaitons concevoir ces instruments possèdent des fonctions permettant de calculer un indice temporel  $\tau(t)$  correspondant à la position lue dans l’enregistrement 1.9. Une étape d’indexation préalable est nécessaire pour déterminer la position des syllabes dans l’enregistrement. Cette indexation se fait manuellement et en dehors du système.

T-Voks [34][57] est un instrument de cette famille. Il s’agit d’un thérémine augmenté qui permet de contrôler la hauteur, l’effort vocal et le déroulement syllabique d’un enregistrement vocal. Les paramètres de hauteur et d’effort vocal sont contrôlés de façon continue par la distance des mains droite et gauche par rapport aux antennes du thérémine comme présenté dans la partie

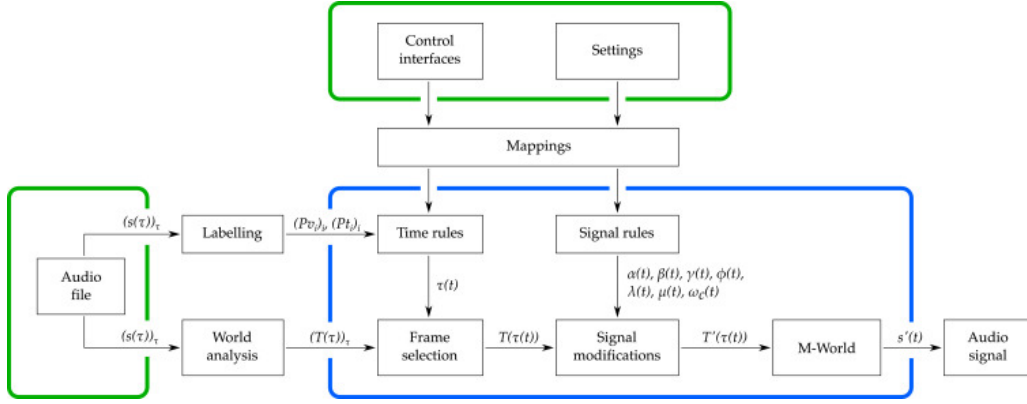


FIGURE 1.9 – Schéma fonctionnel de la famille d'instruments numériques Voks [34]

gauche de la figure 1.10. Le déroulement syllabique s'effectue par contrôle discret à l'aide d'un capteur FSR comme présenté dans la partie droite de la figure 1.10. Ce système intègre un mode de contrôle rythmique complexe qui calcule une fonction parabolique à chaque fois que l'utilisateur déclenche le passage à la syllabe suivante. Cette parabole sert de guide pour l'évolution de la vitesse de lecture entre la position dans l'enregistrement au moment où l'utilisateur appuie sur le capteur FSR et la position du début de la syllabe suivante. Si l'utilisateur déclenche le passage à la syllabe suivante, alors que le système est au milieu de la syllabe actuelle, la parabole calculée aura une pente plus forte au début pour rattraper le retard. Grâce à ce système on peut lire un échantillon segment par segment sans discontinuité lors du passage au segment suivant. L'intégralité de l'enregistrement est lu, seul le profil de vitesse évolue pour adapter le déroulement syllabique à la performance de l'utilisateur.

Le système de captation gestuelle récupère les données captées par le geste de l'utilisateur (ici le thérémine et le capteur FSR) et les transcrit en indice temporel  $\tau(t)$  correspondant à la position lue dans l'enregistrement. La transformation sonore est effectuée ici par le vocodeur de phase WORLD[40] qui permet d'étirer et de compresser temporellement la durée de l'enregistrement. Le vocodeur récupère en entrée la position temporelle calculée par la fonction de captation gestuelle et synthétise par Transformée de Fourier inverse l'enregistrement selon la position indiquée.

Les applications présentées dans cette introduction permettent des interactions très différentes les unes des autres. Néanmoins certaines similarités ont

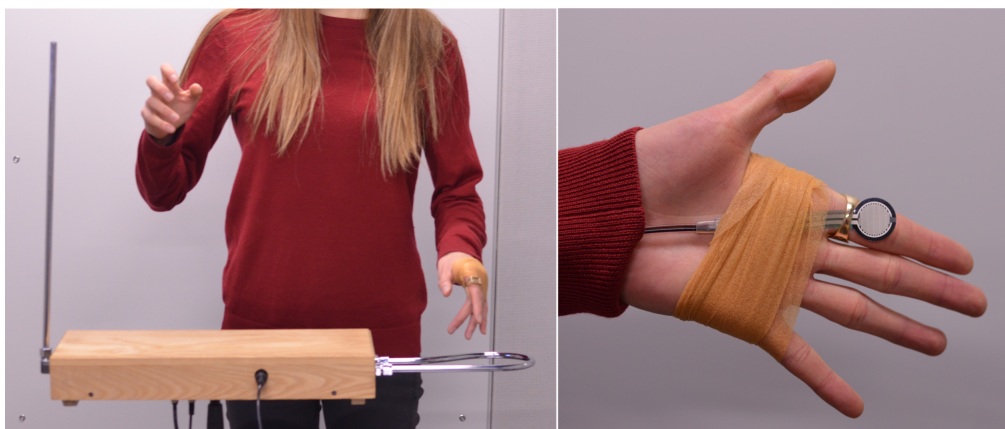


FIGURE 1.10 – T-Voks est un thérémine augmenté permettant de contrôler le déroulement syllabique, la hauteur et l’effort vocal d’une voix enregistrée à l’aide des antennes de l’instrument et d’un capteur FSR.

étés soulignées : elles récupèrent toutes en entrée un fichier d’indexation ainsi que des données captées par le geste de l’utilisateur et donnent en sortie une position temporelle dans l’enregistrement initial. Nous présentons les particularités de chacun de ces systèmes dans le tableau récapitulatif disponible figure 1.11. Cette observation va nous guider dans toute la suite de ce manuscrit. Serait-il possible d’en dégager un système compatible avec n’importe quel mode de contrôle temporel ? Comment cela peut-il concrètement s’intégrer dans un logiciel ? Dans ce travail nous présentons ce système ainsi que son implémentation logicielle. Nous mettons ensuite ce logiciel à disposition de différents artistes afin qu’ils le prennent en main. Au travers d’entretiens semi-directifs menés avec chacun d’entre eux nous récupérons et analysons l’expérience subjective du musicien face à l’outil.

### 1.3 Trois concepts clés pour développer des interactions basées sur la manipulation d’un enregistrement

La perception d’un son est un processus cognitif déterminant dans notre manière d’interagir avec notre environnement. Le but de cette section est de comprendre, au travers de plusieurs concepts liés aux sciences cognitives, comment l’écoute d’un son enregistré influe sur la représentation gestuelle qu’on

Système	Support(s) de l'enregistrement	Système(s) d'étiquetage	Mode(s) de contrôle temporel	Interface(s) gestuelle(s)
<b>Phonographe d'Edison</b>	<ul style="list-style-type: none"> <li>• Cylindre d'étain</li> <li>• Cylindre de cire</li> </ul>	<ul style="list-style-type: none"> <li>• Pas d'outil d'étiquetage</li> </ul>	<ul style="list-style-type: none"> <li>• Contrôle temps réel de la vitesse de lecture</li> </ul>	<ul style="list-style-type: none"> <li>• Manivelle</li> </ul>
<b>Échantillonneur Akai MPC (Sampling)</b>	<ul style="list-style-type: none"> <li>• Fichier audionumérique</li> </ul>	<ul style="list-style-type: none"> <li>• Indexation automatique des transitoires</li> <li>• Indexation manuelle</li> </ul>	<ul style="list-style-type: none"> <li>• Contrôle temps réel de la position de déclenchement</li> <li>• Réglage préalable de la vitesse de lecture</li> </ul>	<ul style="list-style-type: none"> <li>• Grille de pad</li> </ul>
<b>Platine Vinyle (Scratch)</b>	<ul style="list-style-type: none"> <li>• Disque Vinyle</li> </ul>	<ul style="list-style-type: none"> <li>• Pas d'outil d'étiquetage automatique</li> <li>• Certains musiciens se servent d'autocollants pour repérer les positions d'éléments sonores</li> </ul>	<ul style="list-style-type: none"> <li>• Contrôle temps réel de la position de lecture</li> </ul>	<ul style="list-style-type: none"> <li>• Platine Vinyle</li> </ul>
<b>Méta-Piano</b>	<ul style="list-style-type: none"> <li>• Fichier MIDI</li> </ul>	<ul style="list-style-type: none"> <li>• Partition MIDI</li> </ul>	<ul style="list-style-type: none"> <li>• Contrôle temps réel de l'avancement rythmique d'une note/accord à l'autre</li> </ul>	<ul style="list-style-type: none"> <li>• Méta-touche</li> </ul>
<b>Voks</b>	<ul style="list-style-type: none"> <li>• Fichier audionumérique</li> </ul>	<ul style="list-style-type: none"> <li>• Indexation manuelle de la position des syllabes</li> </ul>	<ul style="list-style-type: none"> <li>• Contrôle temps réel de l'avancement syllabique</li> </ul>	<ul style="list-style-type: none"> <li>• Thérémine</li> <li>• Tablette</li> </ul>
<b>SmartMusicKIOSK</b>	<ul style="list-style-type: none"> <li>• Fichier audionumérique</li> </ul>	<ul style="list-style-type: none"> <li>• Indexation automatique de la position des refrains</li> </ul>	<ul style="list-style-type: none"> <li>• Contrôle temps réel de la position de déclenchement de lecture</li> </ul>	<ul style="list-style-type: none"> <li>• Souris/clavier</li> </ul>
<b>Mono-Ovisir</b>	<ul style="list-style-type: none"> <li>• Fichier audionumérique</li> </ul>	<ul style="list-style-type: none"> <li>• Indexation manuelle des positions de début et de fin de boucle</li> </ul>	<ul style="list-style-type: none"> <li>• Contrôle temps réel de la vitesse de lecture et de la position de déclenchement</li> </ul>	<ul style="list-style-type: none"> <li>• Souris/clavier</li> </ul>

FIGURE 1.11 – Tableau récapitulatif des systèmes présentés dans cet état de l'art

lui associe. En synthétisant ces différentes approches nous pourrions orienter nos travaux vers le développement d'outils intégrant ces comportements afin de rendre l'expérience de jeu la plus intuitive possible.

### 1.3.1 Le substitut gestuel

Selon Denis Smalley, lorsque nous écoutons une musique enregistrée, nous associons l'évolution du son que nous entendons à une action que nous imaginons être à la source de celui-ci. Par cet acte plus ou moins conscient de la pensée nous augmentons notre expérience d'écoute en la reliant au geste qui est à l'origine du son entendu : "le savoir intuitif du geste humain physique impliqué au moment de la création d'un son est inextricablement lié à notre savoir de la musique en tant qu'activité" <sup>5</sup> [54].

Selon l'approche de ce chercheur, l'expérience pour un auditeur d'écouter un enregistrement d'instrument de musique est un processus conditionné culturellement : inconsciemment il associe le son qu'il perçoit à un geste instrumental qu'il a déjà vu. Ainsi l'expérience auditive est augmentée d'une

5. citation extraite de *Spectromorpholy : explaining sound-shapes* parut dans la revue *Organised sound*, 1997, Cambridge University Press, p. 109 "intuitive knowledge of the human physical gesture involved is inextricably bound up with our knowledge of music as an activity"

représentation concrète de l'esprit quant à la source sonore perçue. C'est ce qui arrive par exemple à l'écoute d'un enregistrement de piano où l'auditeur perçoit dans son expérience auditive le geste du pianiste associé à la création d'un son. Dans les musiques électroacoustiques il est plus difficile de connecter la création d'un son à sa perception du fait que les corps sonores créés ne sont pas issus d'une source acoustique facilement identifiable. Il s'agit bien souvent d'une combinaison de "gestes de travail" qui ne contiennent pas les mêmes informations perceptives que le savoir intuitif du geste physique de la production sonore traditionnelle. Ces différents degrés de représentation sont appelés "Substitut gestuel" (*gestural surrogacy* en anglais) et s'organisent en quatre catégories :

**Substitut gestuel du premier ordre :** Ce premier niveau de représentation concerne les gestes invoqués dans la création d'objets sonores bruts avant leur instrumentalisation, sous entendu leur incorporation dans un contexte musical. Ce premier niveau de représentation n'est donc pas associé à un geste instrumental mais plus à un geste de création du son. C'est par exemple les gestes employés lorsque l'on entrechoque deux pierres ensemble. On peut considérer qu'à ce niveau le geste impliqué peut être utilisé comme un matériau de composition mais il ne pourra jamais atteindre pleinement le statut culturel d'un instrument. Si le son est transformé et incorporé dans un processus de composition il ne s'agira plus du même substitut gestuel.

**Substitut gestuel de deuxième ordre :** Il s'agit de tous les gestes d'instruments de musique traditionnels où des compétences de performances reconnaissables sont utilisées pour développer un jeu articulé. Dans une musique acousmatique, ce deuxième ordre de représentation gestuelle concerne seulement les enregistrements d'instruments que l'on peut identifier. On peut aussi observer ce degré de représentation lors de l'utilisation d'un outil numérique de simulation d'un instrument existant. Lorsque le joueur produit le son par ces instruments, il intègre le geste à l'origine du son qu'il reconnaît au moment de l'écoute de sa performance. Cette représentation englobe aussi tous les synthétiseurs qui créent des sons dont les sources sonores peuvent être associées

à une pratique instrumentale connue. Le geste de création du son est évoqué de façon claire dans l'esprit de l'utilisateur.

**Substitut gestuel de troisième ordre :** Cette représentation gestuelle intervient lorsque le geste est imaginé ou "déduit" au moment de l'écoute du son. Ce troisième ordre concerne toutes les sources dont il est difficile d'identifier les causes mais dont on arrive néanmoins à déduire certains aspects. Par exemple si le son entendu est un bruit sec et bref l'auditeur se représentera plutôt un geste d'impact mais il n'arrivera pas à se représenter clairement la source sonore à l'origine du son.

**Substitut gestuel éloigné :** Ce substitut gestuel évoque une représentation gestuelle partielle. À ce niveau, les sources et les causes sont inconnues et pas connaissables du fait que toute action humaine derrière la source a disparu. L'auditeur se concentre à la place, sur les attributs "spectromorphologiques" du son perçu, c'est-à-dire l'évolution temporelle de paramètres spectraux. Il y a néanmoins des vestiges de gestes liés à une action physique à la source du son toujours présents. Pour les trouver il faudrait selon Smalley se référer aux propriétés de tension proprioceptives et aux caractéristiques d'effort et de la résistance perçue dans la trajectoire du geste. Ainsi la représentation gestuelle éloignée, bien que distante de la représentation gestuelle de premier ordre, peut être liée à une représentation de degré inférieur. Pour ressentir ça, il faut qu'il y ait suffisamment d'énergie dirigée, propagée ou réinjectée dans la "spectromorphologie". Par exemple dans les musiques drones, de longues nappes musicales évoluent très lentement dans le temps et l'on parvient difficilement à identifier un geste concret à l'origine du son que l'on entend. On peut néanmoins facilement relier l'évolution du son entendu à l'évolution des propriétés spectrales et temporelles de celui-ci. Le mouvement que l'auditeur associe au son n'est donc plus lié au geste à l'origine de celui-ci mais à l'évolution de sa "spectromorphologie".

La théorie de Denis Smalley montre que la conception d'un geste d'interaction avec la lecture d'un enregistrement commence dès la phase d'écoute. Dans le cas d'une écoute acousmatique, c'est-à-dire sans que la source sonore

	perceptions				
substituts gestuels	reconnais- sance de la source	invention de la source	matériau de base	jeu musi- cal de type instrumen- tal	jeu musi- cal acous- matique
premier geste	✓		✓		
deuxième ordre	✓	✓		✓	
troisième ordre	✓	✓			✓
éloigné		✓			✓

FIGURE 1.12 – Tableau synthétique des différents degrés de *substitut gestuel* selon Pierre Couprie [11]

soit visible, on relie le son entendu à un geste que l'on connaît. On peut donc orienter la conception de notre geste compte tenu des propriétés du son que l'on entend. Si l'on choisit par exemple de manipuler un enregistrement de percussion on peut prédire que l'auditeur qui entendra ce son se représentera un geste lié à la nature de celui-ci : le choc d'un matériau avec un autre. Si l'on choisit de développer une interaction basée sur un geste qui n'est pas percussif alors on sait que l'expérience n'ira pas dans la direction de sa représentation gestuelle.

Dans le cadre des pratiques liées à la musique électroacoustique il arrive très souvent de manipuler des sons dont le substitut gestuel est éloigné. Dans ce cas il est difficile de prédire un comportement. Il est alors utile d'observer le son selon ses propriétés "spectromorphologiques" pour concevoir notre interaction selon les évolutions de celles-ci. Par exemple, si la hauteur du son évolue lentement sur une grande plage de valeurs, alors on peut concevoir ou utiliser une interface gestuelle qui permet de faire des gestes lents et amples pour rendre l'expérience de l'utilisateur plus intuitive. De plus du côté spectateur si le geste est corrélé à la représentation gestuelle du son qu'il entend, le caractère expressif du geste sera d'autant plus compris.

Certains systèmes comme le méta-piano [31] proposent des interactions cohérentes avec le substitut gestuel de second ordre que le son généré évoque. Dans le cas de ce système il s'agit de contrôler le rythme, le phrasé et l'accentuation d'une pièce pour piano à l'aide d'un clavier midi. Celui-ci pilote

l'avancement dans la partition préprogrammée qui est jouée par un synthétiseur de piano. Ainsi, l'utilisateur se focalise essentiellement sur les paramètres d'interprétation de la pièce rejouée, en contrôlant le rythme sans se soucier des hauteurs jouées. Dans ce cas-là l'expérience va dans le sens de la représentation gestuelle du son entendu : l'interface est un clavier et le substitut gestuel associé est celui du geste du pianiste.

### 1.3.2 L'énaction

Le concept d'*énaction* établi par le neurobiologiste Francisco Varela et Humberto Maturana en 1992 [55] donne un point de vue novateur sur la théorie de l'évolution des espèces. En observant l'environnement propre à une espèce, on parvient à comprendre comment celui-ci influe sur son évolution. Du point de vue de la conception d'interaction musicale, ce concept souligne l'existence d'un savoir incarné, qui conditionne la façon dont nous interagissons avec un système. Ce savoir incarné s'acquiert de par nos expériences sensorielles avec le monde physique qui nous entoure. Par exemple si l'on observe l'acte de taper du doigt une table en bois, la réaction qui se produira à la suite de cette action est parfaitement anticipée par notre cerveau. Avant même d'avoir percuté la table on connaît déjà la nature du son qui sera émis ainsi que la réaction de notre doigt à ce choc. C'est grâce à nos expériences passées que l'on a développé ce savoir. Mais que se passe-t-il si on développe une interaction en se basant sur ce type de théorie ? L'expérience est-elle plus intuitive ? L'instrument est-il plus facile à prendre en main ?

Georg Essl est un informaticien et compositeur autrichien qui a travaillé sur ces théories. Il conçoit des instruments nouveaux en se basant sur ce savoir incarné. Par exemple [23] grâce à la Pebble Box, il transforme le son obtenu lorsque l'on secoue de la main des cailloux disposés dans une boîte en différents sons de synthèse. Selon lui si l'on change complètement les paramètres de synthèse sonore, la dynamique de l'interaction sera toujours appropriée. Effectivement la dynamique de son système repose sur le comportement des cailloux qui s'entrechoquent au contact de la main. Cette sensation est bien connue de tous et la réaction du système à cette interaction est parfaitement familière à l'utilisateur. En récupérant le signal produit par un capteur piézo-



électrique situé sous la boîte, il relie l'évolution de ce signal à des paramètres de contrôles de la synthèse sonore. Ainsi l'utilisateur génère du son en remuant les cailloux contenus dans la boîte. Essl explique que le degré de corrélation entre l'expérience haptique et le son produit permet d'étendre le rang des couplages de ceux-ci. Le fait que le son ne soit pas produit directement par la main au contact du caillou, mais par les cailloux qui s'entrechoquent entre eux, permet de garder la nature intuitive de l'expérience haptique malgré une réponse sonore décorrélée de cette sensation.

Du point de vue de notre travail de recherche, l'approche éenactiviste met en lumière les facteurs cognitifs qui vont impacter l'utilisation du système que l'on propose. D'après Essl en intégrant dans l'interaction musicale que l'on développe une expérience déjà vécue avec l'environnement qui nous entoure, on peut décorréler plus facilement le geste et la réponse sonore associée. L'interaction gardera un aspect intuitif pour l'utilisateur qui retrouvera dans celle-ci une expérience qu'il a déjà vécue. Alors que Denis Smalley soutient qu'à chaque son correspond une représentation gestuelle, Essl, lui, soutient qu'à chaque geste correspond un savoir incarné. Bien loin d'être opposées, ces deux visions sont tout à fait complémentaires et permettent de comprendre un peu mieux nos comportements face à une expérience musicale auditive et haptique. Le son que l'on entend évoque un geste et l'interface que l'on utilise évoque une expérience déjà vécue.

### **1.3.3 Le concept de re-enactment**

Le chercheur Norbert Schnell définit ce terme dans son travail de thèse [53]. Il s'en sert pour décrire de façon conceptuelle les mécanismes mis en place par un utilisateur au moment de l'interaction avec un système musical interactif basé sur des enregistrements sonores. Il aborde ces questions du point de vue du concepteur d'application.

Le concept de "re-enactment" peut être traduit littéralement par "reconstitution de l'évènement". Il suppose que dans les processus cognitifs d'interaction homme machine restituant un son enregistré, toutes nos actions et notre perception sont connectées à un réseau entier d'actions que nous avons expérimentées auparavant. Dans ce contexte, de la même manière que l'approche

énactiviste de Essl, les gestes d'interactions peuvent se référer à des actions concrètes déjà expérimentées. Elles peuvent aussi se référer à des notions plus abstraites d'actions, comme les techniques, les événements historiques et la pratique plus liée au contexte de l'événement. La représentation gestuelle à l'écoute d'un son, n'est ici pas perçue simplement comme le fruit d'un acte physique plus ou moins identifiable produisant une onde sonore. Il est aussi identifié comme un phénomène lié à un contexte social, technique ou historique au moment de la captation du son. En quelque sorte lorsque l'on manipule un son enregistré dans un système interactif on cherche aussi à reconstituer le contexte autour du son au moment de sa captation. Pour illustrer cela, on peut imaginer un système permettant d'interagir avec l'enregistrement d'un concert de rock, alors la ré-énaction représenterait tous les mécanismes mis en place par l'utilisateur pour reproduire, en plus de la musique, le contexte dans lequel le son qu'il manipule a été enregistré.

Cette idée de "reconstitution" fait directement écho à la théorie sur le substitut gestuel développée par Smalley décrite plus haut. L'expérience auditive induit une représentation gestuelle corrélée à l'acte physique à l'origine du son. Dans l'approche de Schnell, c'est la reconstitution du contexte dans lequel cet acte a lieu qui améliore l'expérience cognitive. Manipuler un son fait intervenir des mécanismes cognitifs forts qui vont influencer notre comportement face à une application audio interactive. Prendre en considération tous ces mécanismes, lors de la conception d'un système, s'avère être une tâche difficile. Il est néanmoins utile de se questionner sur chacun des concepts vus précédemment si l'on souhaite anticiper le comportement d'un utilisateur pour une interaction en particulier.

## **1.4 Contexte de travail : PUCE MUSE et les Mono**

Ce travail a été réalisé en grande partie au sein d'une structure privée nommée PUCE MUSE. Spécialisée dans la création artistique audiovisuelle, cette association a été fondée en 1982 par le musicien Serge de Laubier avec pour but de développer des créations artistiques et des projets pédagogiques dans le domaine des musiques visuelles. Avec une forte composante autour des

pratiques musicales collectives, ce centre de création audiovisuelle intervient dans de nombreux collèges, lycées et conservatoires d'île de France dans le cadre d'atelier autour de la prise en main d'outils informatiques pour la création musicale [43] [17] [19]. Ces outils sont distribués gratuitement par PUCE MUSE sur le site meta-librairie.com<sup>6</sup> en visant un public large d'utilisateurs, allant du néophyte à l'expert. L'enjeu majeur de ce travail de recherche pour la structure PUCE MUSE a été la production d'un logiciel informatique multiplateformes intégrant tous les aspects abordés dans cette thèse. Le développement de cet outil a permis de valider les concepts abordés dans ce travail par la réalisation concrète d'un système les intégrant.

Les Mono sont des logiciels développés et distribués par l'association PUCE MUSE. Ils sont utilisés essentiellement dans le cadre d'ateliers pédagogiques dans des collèges, lycées et conservatoires. Ils ont été pensés pour permettre une utilisation "stand-alone" des logiciels qui composaient autrefois la méta-mallette [17]. Ils permettent de manipuler des transformations sonores et visuelles sur des sons et vidéo enregistrés.



FIGURE 1.13 – Interface utilisateur de Mono-Ovisir. Cette application développée par PUCE MUSE permet de contrôler de façon expressive le déroulement temporel d'un enregistrement audio et d'une séquence vidéo en simultanée.

6. <https://www.meta-librairie.com/fr>

Les Mono prennent la forme de lecteurs audio et vidéo grâce auxquels on peut contrôler des transformations pendant la lecture du fichier. Mono-Trans1 et Mono-Trans2 permettent de contrôler des effets de delay. Mono-Ovisir dont l'interface est présentée figure 1.13 permet de contrôler la vitesse et la position de lecture de clips vidéo ainsi que des transformations sur l'image.

Chaque Mono dispose d'un dictionnaire OSC qui permet d'avoir un accès externe à tous les paramètres du logiciel. Ainsi on peut relier des variables captées par une interface gestuelle à des paramètres de transformation dans le logiciel. Cette fonctionnalité nous permettra de concevoir rapidement des interactions faisant intervenir une interface captant un geste continu ou discret pour contrôler le logiciel.

Les Mono intègrent tous des fonctionnalités de base comme la sauvegarde de mémoire, l'enregistrement de séquences audio ou vidéo ou encore le contrôle du volume. Ces fonctionnalités ont été conçues de façon à faciliter les mises à jour sur l'ensemble des logiciels de cette suite. Ainsi tous les Mono disposent d'un répertoire sur les serveurs de PUCE MUSE dans lequel un sous-module commun à tous les Mono est mis à jour régulièrement. Une des autres particularités de ces logiciels est qu'ils intègrent un système de gestion de mémoire très axé sur l'expressivité. En glissant la souris sur un axe horizontal il est possible de passer d'une mémoire à l'autre en interpolant les valeurs. Cette "grille d'interpolation" est intégrée à tous les Mono et permet de changer le réglage de plusieurs paramètres durant une performance, sans discontinuité entre une valeur et une autre.

Mono-Ovisir s'organise de la même façon que les logiciels présentés précédemment. Le système d'indexation est ici manuel et permet à un utilisateur de renseigner la position de différents points de bouclage. La fonction de contrôle gestuel est gérée dans le logiciel par le dictionnaire OSC qui donne accès à tous les paramètres du logiciel. Ainsi l'utilisateur peut envoyer des commandes à Mono-Ovisir par le biais d'interfaces gestuelles externes.

Ces outils étant destinés aussi bien à des utilisateurs débutants qu'à des utilisateurs experts, implémenter notre système dans cet environnement nous permettrait de faciliter son utilisation.

## 1.5 Bilan et Enjeux

Comme nous venons de le voir précédemment, interagir avec un enregistrement est un processus qui peut prendre de nombreuses formes. De l'interaction simple consistant à lancer la lecture d'un enregistrement à une interaction plus complexe qui consiste à avancer dans un enregistrement vocal, syllabe par syllabe, chacun des systèmes que nous avons vus précédemment propose un contrôle gestuel particulier et unique bien qu'ils aient à première vue les mêmes fonctionnalités, à savoir : une fonction d'indexation, une fonction de captation gestuelle et une fonction de rendu sonore.

Ces observations nous ont poussé à nous questionner sur d'éventuelles compatibilités entre tous ces modes. Serait-il possible de tous les intégrer dans un même système ? Est-ce qu'une telle intégration aurait un intérêt pour la création artistique ? Ces deux questions sont fondamentales dans ce travail et nous tentons d'y apporter une réponse tout au long du manuscrit.

La théorie de Denis Smalley [54] sur le substitut gestuel implique qu'il existerait une représentation gestuelle pour chaque son enregistré. Si on souhaite construire une interaction selon cette représentation gestuelle cela sous-entend qu'il existe au moins autant de gestes d'interaction possibles qu'il existe de sons enregistrés. Ce constat pousse à imaginer un système laissant la possibilité pour un utilisateur de concevoir autant de gestes qu'il souhaite compte tenu de l'enregistrement qu'il manipule.

Les théories de l'énaction et de la re-énaction vont aussi dans ce sens. À chaque son perçu on associe une expérience vécue qui vient augmenter notre expérience d'auditeur [23]. Concevoir une interaction qui permettrait de manipuler par le geste un enregistrement, serait une expérience plus intuitive s'il était possible d'intégrer facilement ces mécanismes cognitifs. Mais comment s'assurer que notre système soit suffisamment ouvert pour qu'un utilisateur puisse concevoir son propre geste et ainsi intégrer sa propre représentation gestuelle lorsqu'il manipulera l'enregistrement ?

Les applications que nous avons décrites dans cette introduction présentent toutes une fonction d'indexation qui permet de renseigner le système sur la position temporelle d'ancres qui seront manipulées dans la suite de l'interac-

tion. Pour le logiciel T-Voks[57] la fonction d'indexation se fait manuellement et indique au système les positions temporelles de chaque syllabe de l'enregistrement. Pour l'application SmartMusicKIOSK cette fonction se fait automatiquement et renseigne le système sur les positions des différentes structures musicales qui composent l'enregistrement. Le choix de l'information contenue dans cette indexation dépend complètement de l'interaction que l'on souhaite concevoir : renseigner sur la position des syllabes d'un enregistrement de voix n'aurait *a priori* aucun intérêt pour le logiciel SmartMusicKIOSK. Néanmoins on pourrait imaginer un système de contrôle rythmique plus général permettant à la fois de faire du contrôle syllabique sur une voix enregistrée comme T-Voks et de lire un enregistrement musical compte tenu de la structure de celui-ci comme SmartMusicKIOSK. Il faudrait alors pouvoir intégrer les fonctions d'indexation des deux logiciels ainsi que leur mode de contrôle rythmique. On pourrait alors changer d'indexation et de mode de contrôle rythmique en fonction du type d'enregistrement que l'on souhaite manipuler. Si l'on souhaite avoir autant de flexibilité sur la fonction d'indexation de nos systèmes, il faudrait la penser le plus en amont possible dans le développement afin qu'elle puisse intégrer aussi bien des méthodes manuelles que des méthodes automatiques.

Notre constat est le suivant : tous les modes de contrôle temporel déjà existants s'organisent de la même façon : ils prennent en entrée un fichier d'annotation et un geste capté et donnent en sortie un indice temporel correspondant à la position de lecture dans l'enregistrement à synthétiser. Comment doit s'organiser ce fichier d'annotation ? Peut-on exprimer plusieurs modes de contrôles temporels selon les mêmes variables ? Serait-il possible de créer un système compatible avec la plupart de ceux-ci ? Quel serait l'intérêt créatif d'un tel dispositif ? Pour répondre à ces questions nous étudierons en détail deux fonctions constitutives du système que nous souhaitons développer.

La première de ces fonctions sert à placer des repères temporels sur l'enregistrement afin d'indexer sa structure interne. Ces repères doivent être organisés selon différents niveaux d'importance qui serviront ensuite à articuler les modes de contrôle temporel. Nous décrirons plusieurs méthodes d'analyse numérique qui facilitent cette tâche. Il faut néanmoins laisser la possibilité de

corriger manuellement chaque repère temporel pour adapter au mieux l'indexation de l'enregistrement à l'interaction que nous souhaitons développer.

La deuxième fonction permet de calculer un indice temporel correspondant à la position à synthétiser dans l'enregistrement. Cette fonction prend en entrée des données de contrôle ainsi que les ancrs temporelles préalablement déterminées. Dans notre cas, nous intégrons dans cette fonction plusieurs méthodes de calcul de l'indice temporel de manière à laisser le choix à l'utilisateur de celle qui semble la plus adaptée à l'interaction qu'il souhaite concevoir.

La troisième fonction est constituée d'un moteur audio qui génère le son lu dans l'enregistrement compte tenu de la position indiquée par les précédentes méthodes de calcul. Ce moteur audio intègre des fonctions de transformations spectrales qui permettent d'étirer la durée d'un enregistrement, de le transposer et de filtrer certains composants de son spectre. Dans notre étude, nous présentons plusieurs méthodes de paramétrage de ces transformations qui tiennent compte de l'indexation de notre enregistrement.

Mono-Replay [35] [36] est un logiciel qui a été conçu durant ce travail de thèse. Il s'intègre dans l'environnement Mono des logiciels développés par l'association PUCE MUSE et comporte toutes les fonctions décrites précédemment. Nous présentons en détail chacune de ces fonctionnalités dans le chapitre 3.

Pour illustrer les possibilités créatives de ce système, nous présentons au chapitre 4 plusieurs cas d'usage du logiciel pour contrôler la lecture d'un enregistrement à l'aide d'une interface gestuelle. Le répertoire musical utilisé ici a été choisi volontairement composite pour montrer l'adaptabilité du système. Nous présentons pour chaque cas la manière dont l'enregistrement est indexé ainsi que l'interface gestuelle utilisée. Chaque interface est reliée au système par un mapping gestuel que nous détaillons.

Pour évaluer ce travail nous soumettons le logiciel à plusieurs artistes compositeurs qui le prennent en main afin de créer une œuvre avec ou pour que celle-ci soit utilisée par le logiciel. Ces compositeurs ont des pratiques musicales différentes les unes des autres. Elles sont décrites et analysées dans le chapitre 5 de ce travail. Dans ce même chapitre nous détaillons la méthodologie des entretiens semi-directifs qui ont été menés *a posteriori* de cette

expérience. Leur but est de récupérer l'expérience subjective de l'artiste avec le système pour ensuite les discuter. Nous pourrions alors conclure sur l'apport créatif de notre logiciel comparé aux outils qu'ils utilisent habituellement. Ce chapitre introduit aussi un travail d'évaluation quantitative sous forme de questionnaires élaborés dans le cadre d'un atelier. Cet atelier a pour but d'explorer les différents modes de contrôle temporel présents dans Mono-Replay dans le cadre d'une pratique collective du logiciel.



## Chapitre 2

# Concevoir une interaction musicale basée sur la manipulation d'un enregistrement

### Sommaire

---

<b>2.1</b>	<b>Les applications audio interactives . . . . .</b>	<b>42</b>
<b>2.2</b>	<b>Indexation . . . . .</b>	<b>45</b>
2.2.1	Indexation de transitoire d'attaque . . . . .	48
2.2.2	Indexation de pulsation . . . . .	58
2.2.3	Indexation manuelle . . . . .	63
<b>2.3</b>	<b>Contrôle temporel . . . . .</b>	<b>64</b>
2.3.1	Contexte . . . . .	65
2.3.2	Contrôle temporel continu . . . . .	68
2.3.3	Contrôle temporel discret . . . . .	69
<b>2.4</b>	<b>Contrôle spectral . . . . .</b>	<b>77</b>
2.4.1	Contrôle continu linéaire . . . . .	78
2.4.2	Contrôle continu linéaire par morceaux . . . . .	78
2.4.3	Contrôle spectral discret . . . . .	80
2.4.4	Contrôle discret par calcul d'enveloppe . . . . .	80

---

Le but de ce travail de recherche est de concevoir un système permettant de contrôler en direct et selon plusieurs scénarios d'interaction expressif les

paramètres temporels et spectraux d'un enregistrement. Pour aboutir à ce résultat il est important de répondre à ces questions :

- Quelles sont les fonctions essentielles qui composent une interaction basée sur le contrôle temporel et spectral d'un enregistrement ?
- Comment doivent être organisées ces fonctions pour que notre système soit compatible avec un grand nombre d'interaction ?

Pour y répondre nous proposons, dans la suite de notre travail, une description globale puis détaillée de chacune des briques algorithmiques qui composent notre système. Cela sert de canevas pour le concepteur d'interaction basée sur le contrôle temporel d'un enregistrement. En respectant l'organisation décrite dans ce chapitre, le concepteur s'assure de développer un système intégrant toutes les fonctionnalités nécessaires pour concevoir des scénarios de contrôles temporels expressifs.

## 2.1 Les applications audio interactives

Le but de cette section est de présenter de manière globale chaque fonction qui compose notre système. Nous faisons une description générale de chacune d'entre elles et nous présentons les relations qui les lient les unes aux autres. Le terme application audio interactive désigne tout système numérique qui permet d'interagir avec la production d'un son. Dans notre cas nous nous intéressons aux applications qui permettent d'interagir de façon musicale avec la lecture d'un enregistrement. Le développement de ces applications a été théorisé par le chercheur Norbert Schnell [53]. Selon lui la conception d'une application audio interactive basée sur la manipulation d'un enregistrement s'organise autour de trois éléments représentés figure 2.1 :

- Enregistrer, analyser et étiqueter un son
- Capturer et analyser un geste
- Générer de façon interactive un signal audio temps réel

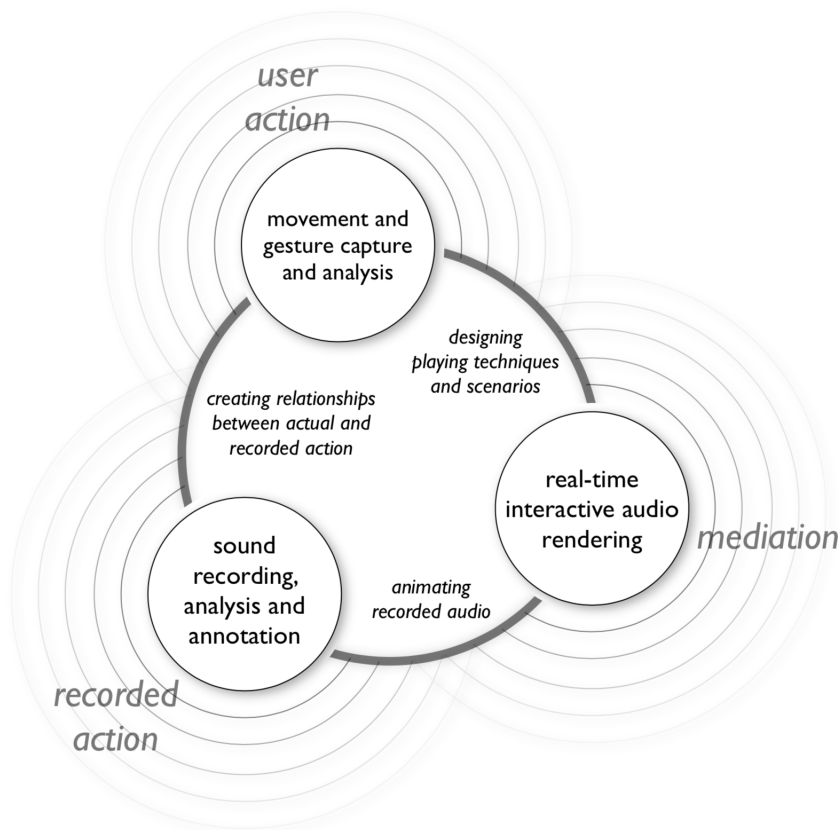


FIGURE 2.1 – Une vue schématique des principaux éléments intervenant dans la conception d’applications audio interactives basées sur des enregistrements sonores [53]

Cette définition donne une parfaite représentation des systèmes que nous étudions ici. En effet, chaque interaction permettant de contrôler temporellement la lecture d’un enregistrement nécessite l’indexation préalable de celui-ci, la captation d’un geste pour le contrôler et un rendu audio temps-réel qui corresponde aux gestes qu’il effectue.

Dans le cas d’une performance de scratch par exemple on observe bien ces différentes fonctions : le musicien relève des positions temporelles sur le vinyle correspondant aux sons qui l’intéressent (phase d’indexation). À l’aide de sa main il fait tourner le vinyle autour de l’axe de la platine pour composer son discours musical (captation gestuelle). Le mouvement de la tête de lecture génère un son de façon instantanée (rendu sonore temps-réel).

- Selon Norbert Schnell l’élément *enregistrer, analyser et étiqueter un son* peut être vu comme une extension de l’écoute [53]. Il permet de

cartographier les éléments saillants d'un enregistrement en vue d'offrir à l'utilisateur des ancres temporelles utiles pour l'interaction musicale. Par exemple, dans le cadre de l'application Mono-Replay décrite chapitre 3, l'indexation permet d'informer automatiquement le système des différentes positions de transitoire d'attaque. Ainsi il est possible de positionner la tête de lecture au début de chaque note qui compose l'enregistrement.

Dans le cas de T-Voks décrit section 1.2.5, cette indexation se fait manuellement et permet d'informer le système des positions de chaque syllabe qui compose l'enregistrement de voix [57].

La représentation visuelle de ces indexations n'est pas forcément accessible à l'utilisateur. C'est le cas par exemple du système Cindy [24] qui permet d'animer un danseur virtuel au rythme d'une musique pulsée. L'indexation se fait automatiquement et le résultat est transmis directement au modèle de synthèse sans être préalablement visualisé par l'utilisateur.

- L'élément *capter et analyser un geste*, quand à lui, a pour but de convertir le geste effectué par l'utilisateur en données de contrôle pour le système. Ce geste peut être, par exemple, celui d'un chef d'orchestre qui bat une mesure. C'est un des scénarios que l'on décrit dans la section 4.2.2. Ce geste de battement est capté puis converti en vitesse de lecture et l'utilisateur peut ainsi faire varier ce paramètre compte tenu de la pulsation qu'il bat. Ici le traitement est relativement simple mais nous verrons dans d'autres exemples que la complexité de ces fonctions est directement liée au nombre de variables que l'on souhaite contrôler lors de la performance.
- L'élément *générer de façon interactive de l'audio en temps réel* peut être vu comme la transcription sonore des deux précédents. Il prend la forme d'un moteur audio qui permet de générer le son selon le geste capté et analysé de l'utilisateur. Dans le cas de Mono-Replay décrit chapitre 3 cette fonction est occupée par le vocodeur de phase SuperVP [50]. Les données étiquetées informent le système des positions d'élé-

ments rythmiques saillants de l'enregistrement et servent à définir les segments joués lors de la performance. Les gestes de l'utilisateur sont captés puis transcrits en valeur discrète ou continue de position. Le vocodeur de phase SuperVP récupère cet ensemble de données pour étirer ou compresser temporellement les différents segments de l'enregistrement joué.

La majorité des applications audio interactives basées sur des enregistrements musicaux peuvent se représenter selon le schéma fonctionnel vu ci-dessus. Cette vue générale nous permet de définir des bases solides à la conception de notre système. En le schématisant de la sorte on souligne les enjeux de chaque élément les uns par rapport aux autres. Dans la suite de notre travail nous verrons plus en détail le contenu de chacun de ces éléments et comment il est possible d'en tirer un formalisme.

## 2.2 Indexation

Dans les systèmes que nous étudions, l'indexation peut être vu comme une extension de l'écoute. Cela permet d'informer le système sur le contenu musical d'un enregistrement mais aussi d'offrir une représentation visuelle les intégrant. Dans le cas de l'application SmartMusicKIOS [26] par exemple, l'indexation informe sur la position temporelle des refrains d'un morceau. Cette indexation est faite de façon automatique à l'aide de l'algorithme RefraiD [25] qui détecte la position d'éléments répétés dans un enregistrement. Une fois l'analyse effectuée l'utilisateur dispose des ancres temporelles nécessaires pour naviguer d'un refrain à l'autre pendant la lecture du fichier. Le module d'indexation est intégré au logiciel et son utilisation se fait de façon transparente et automatique.

Dans beaucoup d'autres cas l'indexation se fait de façon manuelle. Dans le logiciel EAnalysis [13] développé par le chercheur Pierre Couprie, on place des formes géométriques sur l'enregistrement audio pour représenter de façon symbolique son contenu musical comme présenté figure 2.2. L'utilisateur peut aussi s'aider d'algorithmes de traitement du signal pour tracer l'évolution de

paramètres sonores et placer ses ancres. Néanmoins les résultats de ces analyses ne sont pas directement liés à des modes de lecture expressif. On peut facilement imaginer l'intérêt artistique d'un outil d'indexation aussi complet que celui de EAnalysis pour le développement de notre système. L'utilisateur aurait accès à une représentation complexe du contenu musical d'un enregistrement et pourrait s'en servir comme retour visuel lors de sa performance.

L'indexation prend des formes très variées compte tenu de l'application que l'on souhaite concevoir. Elle peut être automatique en se basant sur des techniques de traitement du signal ou manuel et à la charge de l'utilisateur. Dans les prochains paragraphes nous expliquons au travers d'exemples plus détaillés, pourquoi il est important d'intégrer ces deux approches dans le système que l'on souhaite concevoir.

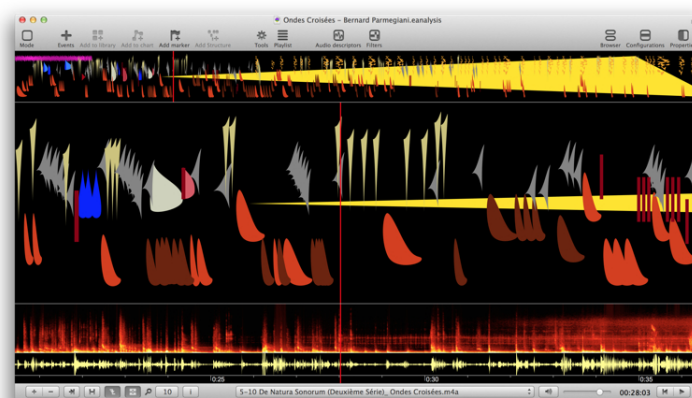


FIGURE 2.2 – Une représentation de l'œuvre *Ondes croisées* composée par Bernard Parmegiani à l'aide du logiciel EAnalysis. Ce logiciel permet de placer des formes géométriques sur l'enregistrement audio pour représenter de façon graphique son contenu musical. (source :<https://logiciels.pierrecouprie.fr/>)

Nous appelons ici "indexation automatique" toute méthode basée sur le traitement du signal permettant de déterminer, à l'aide d'un algorithme, les positions temporelles d'éléments le composant. Ces méthodes ont été élaborées pour la plupart par les chercheurs en MIR.

Pour avoir un ordre d'idée des possibilités d'indexation offertes par ces technologies on peut se référer au travail taxinomique du chercheur Geoffroy Peeters qui a classifié en 2003 plus d'une quarantaine de descripteurs audio

[48]. Ces descripteurs peuvent informer sur des aspects fréquentiels d'un son tels que l'harmonicité ou des aspects temporels tels que la durée des transitoires d'attaque. Il est possible de combiner plusieurs de ces descripteurs pour en obtenir d'autres ce qui augmente encore plus les possibilités.

Dans le cadre de la conception d'application audio interactive ces méthodes d'analyses sont très utiles à l'élaboration de fichiers d'indexation. Elles donnent rapidement accès à des informations contenues dans un enregistrement. Ces informations permettent ensuite de concevoir des collections d'ancres temporelles qui seront manipulées pendant la performance. Si l'on reprend le travail de Mastaka Goto décrit section 2.1, la plupart des outils qu'il développe intègre ce type de méthode. Des algorithmes de traitement du signal récupèrent de façon automatique la position temporelle d'éléments musicaux comme un refrain par exemple. L'avantage de l'automatisation de cette étape est qu'elle permet instantanément de donner accès aux informations nécessaires à l'utilisation du système.

Dans le cas des samplers il est très fréquent de trouver ce type de méthode notamment pour la détection d'événement rythmique. Comme montré sur la figure 1.3 le sampler natif d'Ableton Live propose une indexation selon la position des transitoires de l'enregistrement. Cette fonction permet à l'utilisateur d'avoir rapidement accès à un découpage de l'enregistrement pour commencer son exploration. Ces fonctions sont souvent paramétrables et permettent d'affiner la sensibilité de détection de l'algorithme.

L'indexation automatique ne garantit pas toujours la fiabilité des informations récupérées. Dans le cas d'une détection de transitoire par exemple, la taille de la fenêtre d'analyse, le réglage du seuil de détection ou encore le niveau de bruit de fond de l'enregistrement sont des paramètres qui influent sur la qualité des résultats. Il est souvent nécessaire d'accéder à des fonctionnalités d'édition pour corriger celui-ci. Cela permet par exemple d'ajouter ou de retirer des ancres temporelles non désirées. Dans la suite de ce paragraphe, nous présentons en détail les possibilités offertes par deux approches d'indexation automatique, la détection de transitoire et la détection de pulsation.

Ces deux approches sont très couramment utilisées dans les outils de contrôle rythmique déjà existants. Ils permettent de récupérer les principales

informations sur le contenu rythmique d'un enregistrement. La détection de transitoire renseigne sur la position des débuts de note et la détection de pulsation sur la position des pulsations ressenties dans l'enregistrement. Comme nous le démontrons dans les paragraphes suivants, leur intégration permet d'obtenir des informations nécessaires pour le fonctionnement d'un grand nombre de modes de contrôle rythmique.

### 2.2.1 Indexation de transitoire d'attaque

Un transitoire est, selon la définition du Larousse, un "phénomène physique intervenant entre deux régimes établis". En musique le transitoire d'attaque correspond à la partie du son faisant transition entre l'excitation de la corde, la percussion d'une peau, la vibration de l'anche et la tenue d'une note ou sa résonance. Chercher des transitoires dans un enregistrement revient donc à déterminer la position des débuts de note dans celui-ci. Si on regarde la représentation schématique du transitoire d'attaque (figure 2.3) on voit que d'un point de vue temporel, l'évolution de son amplitude se décompose en deux parties distinctes :

- Un transitoire d'excitation correspondant à l'évolution de l'amplitude d'une onde sonore vers un maximum.
- Un transitoire de résonance correspondant à l'après-excitation et au retour vers un régime pseudo-périodique.

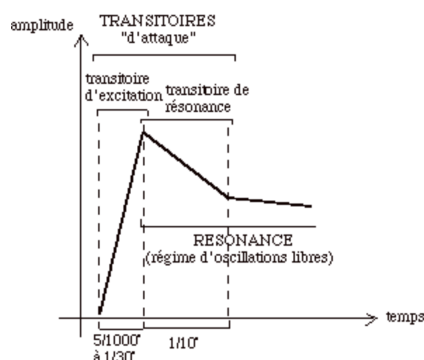


FIGURE 2.3 – représentation schématique du transitoire d'attaque [41]



Si l'on adopte maintenant un point de vue spectral, la partie transitoire d'une note correspond au temps pendant lequel le contenu spectral est rapidement variable. Pour le cas de la corde tendue frappée par le marteau d'un piano par exemple, la représentation spectrale du transitoire d'excitation prend la forme d'une très courte tranche de bruit blanc. Le son passe ensuite sous une forme plus tenue où la répartition fréquentielle prend la forme d'un peigne comportant plus ou moins d'harmoniques.

	PN data			PP data			NP data			CM data		
	P	R	F	P	R	F	P	R	F	P	R	F
SF*	0.914	0.871	<b>0.892</b>	0.984	0.949	<b>0.966</b>	0.945	0.816	<b>0.876</b>	0.896	0.804	<b>0.848</b>
SF	0.938	0.968	<b>0.952</b>	0.981	0.988	<b>0.984</b>	0.959	0.975	<b>0.967</b>	0.882	0.882	<b>0.882</b>
PD*	0.957	0.957	<b>0.957</b>	0.997	0.955	<b>0.976</b>	0.945	0.807	<b>0.871</b>	0.753	0.801	<b>0.776</b>
PD	0.654	0.935	<b>0.770</b>	0.482	0.865	<b>0.619</b>	0.750	0.933	<b>0.831</b>	0.663	0.749	<b>0.704</b>
WPD	0.937	0.957	<b>0.947</b>	0.899	0.925	<b>0.912</b>	0.974	0.958	<b>0.966</b>	0.843	0.830	<b>0.836</b>
NWPD	0.909	0.968	<b>0.938</b>	0.961	0.981	<b>0.971</b>	0.950	0.966	<b>0.958</b>	0.916	0.845	<b>0.879</b>
CD	0.946	0.946	<b>0.946</b>	0.971	0.984	<b>0.978</b>	0.948	0.924	<b>0.936</b>	0.941	0.819	<b>0.876</b>
RCD	0.948	0.978	<b>0.963</b>	0.983	0.979	<b>0.981</b>	0.944	0.983	<b>0.963</b>	0.945	0.819	<b>0.877</b>

FIGURE 2.4 – Tableau comparatif réalisé par Dixon [21] sur 6 méthodes de détection de transitoires (spectral flux (SF et SF\*), phase deviation (PD et PD\*), complex domain (CD), weighted phase deviation (WPD), rectified complex domain (RCD), normalized weighted phase deviation (nwpd)). Pour chaque méthode on calcule la précision  $P$ , la sensibilité (ou rappel)  $R$  et la mesure  $F$ . Cette étude a été réalisée sur des bases de données incluant des sons percussifs à hauteur définie (PN) ; des sons percussifs à hauteur non définie (PP) ; des sons non-percussifs à hauteur non définie (NP) et des sons mélangeant les trois précédentes catégories (CM)

Il existe une multitude de méthodes permettant de déterminer ces transitoires. Chacune se base sur l'analyse de propriétés spectrales ou temporelles d'un enregistrement pour déterminer la position de chaque transitoire. Dans le tableau récapitulatif de l'étude comparative de [21] on voit les performances de plusieurs méthodes de détection de transitoires sur différents types de son. Les bases de données utilisées dans cette étude comportent des sons percussifs à hauteur définie (piano, guitare...) ; des sons percussifs à hauteur non définie (batterie, cymbale, percussions...) ; des sons non percussifs à hauteur définie (cordes frottées, violon, contrebasse..) et des sons mélangeant les trois précédentes catégories.

Les méthodes utilisées dans cette étude n'ont pas la même performance compte tenu du type de son qui est analysé. Alors que la méthode de détection par analyse du flux spectral ( $SF$ ), que nous détaillerons plus tard, est la

plus performante pour les sons percussifs à hauteur définie (*PP*), elle est la troisième plus performante pour les sons percussifs à hauteur non définie (*PN*).

Les systèmes que nous souhaitons élaborer ici ne sont pas conditionnés pour un certain type de son. Pour rappel la motivation première de notre démarche est de penser un système compatible avec n'importe quel mode de contrôle temporel pour interagir avec n'importe quel type d'enregistrement. Compte tenu des remarques faites précédemment, il faut donc prévoir la possibilité d'intégrer plusieurs méthodes d'analyse dans notre système. Ainsi, si une méthode ne permet pas de récupérer des informations sur un type d'enregistrement il est possible d'en essayer une autre.

La figure 2.5 illustre bien ce propos. Ici nous souhaitons déterminer la position des transitoires dans une portion de la pièce *Accidents/Harmoniques* composée par Bernard Parmegiani. Ce morceau est composé des trois catégories de transitoires vues précédemment. Il est donc nécessaire d'essayer plusieurs méthodes de détection de transitoires pour voir celle qui est la plus appropriée compte tenu de la section que l'on souhaite analyser. Dans la figure 2.5 on voit que la méthode dite de déviation de phase [21] ne détecte que deux transitoires sur cette portion. La méthode par différence spectrale [21] quant à elle en détecte plus d'une vingtaine. La dernière est plus appropriée pour déterminer des transitoires pour des sons percussifs alors que la deuxième basée sur la déviation de phase est plus adaptée pour détecter les transitoires plus doux. Dans le cas de cette portion il est donc plus avantageux d'utiliser la méthode par différence spectrale.

Dans les deux cas on est obligé de passer par une phase de correction manuelle pour obtenir un résultat intégrant tous les débuts de notes compris dans l'enregistrement. Dans le premier cas seuls les transitoires suffisamment marqués dans le spectre sont détectés alors que dans le deuxième cas on détecte plus de transitoires qu'il n'existe de notes jouées dans l'enregistrement. Il faut donc, dans un cas, ajouter manuellement les transitoires plus doux qui n'ont pas été détectés par la méthode par différence spectrale et, dans le deuxième cas, il faut retirer manuellement les transitoires qui ont été détectés à de mauvais moments.

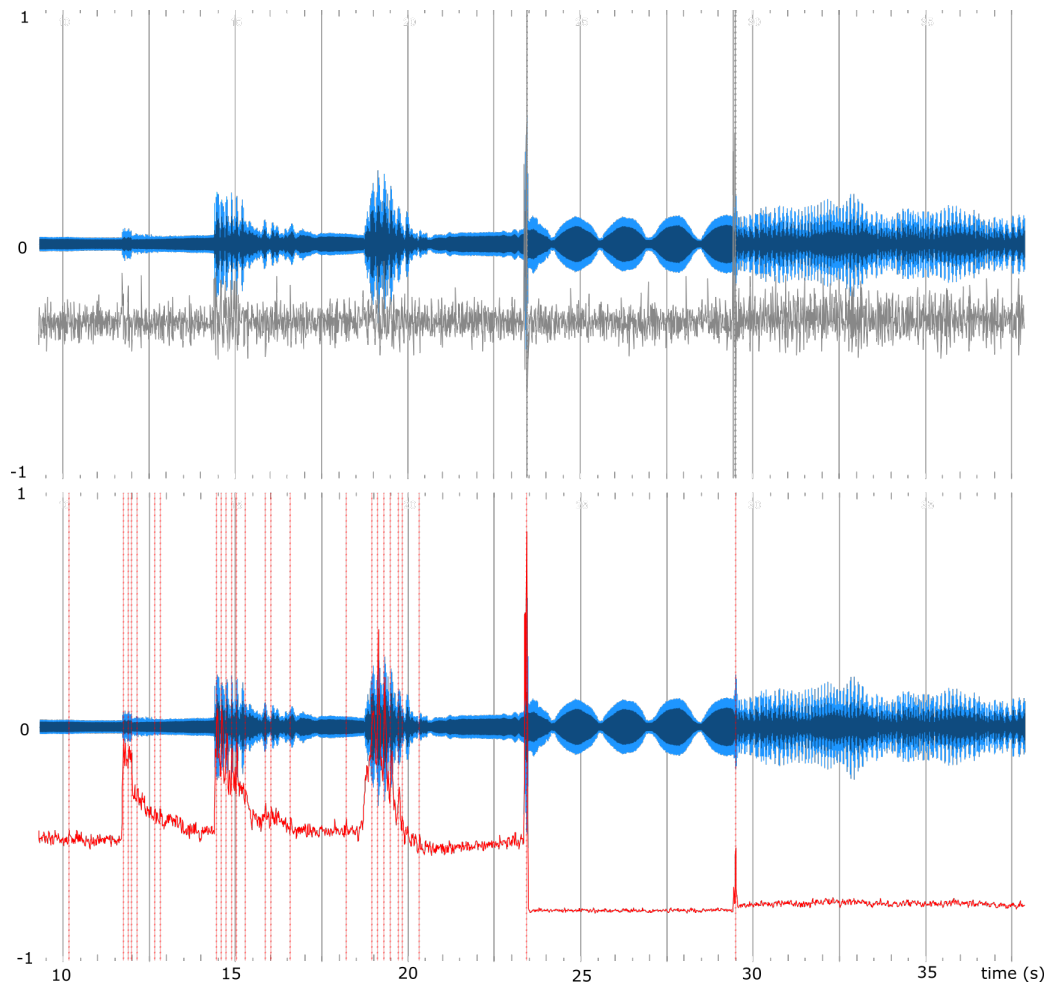


FIGURE 2.5 – Détection des transitoires d’attaque selon la méthode par déviation de phase (en haut) et selon la méthode de différence spectrale (en bas)

Les algorithmes de détection de transitoires permettent d’obtenir de façon discrète l’évolution des paramètres physiques propres à l’apparition d’un nouveau son dans un enregistrement. Comme nous l’avons vu précédemment ces paramètres physiques peuvent se manifester de plusieurs façons : valeur maximale de la somme des différences de magnitude spectrale ; augmentation de la déviation de phase...

D’après la littérature [21] et l’exemple de *Accidents/Harmoniques* décrit précédemment, pour un même enregistrement il est fréquent d’obtenir des résultats différents compte tenu de la méthode utilisée. C’est pour cette raison qu’il est important d’intégrer dans notre système, une architecture compa-

tible avec un grand nombre d'algorithmes de traitement du signal. L'idée est de pouvoir facilement changer d'algorithme compte tenu du son que l'on souhaite analyser. Aujourd'hui de nombreux algorithmes d'analyse sont disponibles dans des langages informatiques très populaires. C'est le cas par exemple de la librairie Essentia [7] qui intègre plus d'une centaine de méthodes originellement conçues en C++. Aujourd'hui portée en Python et en Javascript elle représente une véritable opportunité pour les systèmes que nous souhaitons concevoir. En résumé, l'indexation automatique de transitoire d'attaque est une opération qui peut être optimisée selon plusieurs paramètres :

- *Une connaissance suffisante de l'enregistrement* : Selon que l'enregistrement possède des sons percussifs, non percussifs ou mixtes, le transitoire d'attaque ne s'exprimera analytiquement pas de la même façon. Il est important d'ajouter qu'il est parfois nécessaire de faire un pré-traitement de l'enregistrement pour augmenter les chances de détecter tous les transitoires. Si l'enregistrement est très bruité par exemple il faudra filtrer ce bruit pour qu'il ne perturbe pas la détection.
- *Un large choix de méthodes d'analyse* : Le choix de la méthode s'effectue compte tenu de la nature des sons que l'on souhaite détecter. Si l'enregistrement comporte des sons provenant uniquement d'une source sonore il est plus facile de sélectionner l'algorithme qui correspondra. Si l'enregistrement possède plusieurs familles de son, on peut choisir la méthode en fonction des sons les plus présents dans l'enregistrement ou alors une méthode composite faisant intervenir plusieurs types de fonction de détection de transitoire [59].
- *La possibilité de paramétrer chaque méthode* : Chaque méthode possède un ensemble de variables permettant de paramétrer l'analyse. Du plus simple choix de la fenêtre d'analyse au plus complexe réglage des seuils de détection ces paramètres jouent un rôle essentiel dans la qualité des résultats que l'on obtient.
- *La possibilité de corriger manuellement les résultats* : Comme nous l'avons vu il y a toujours une marge d'erreur entre le résultat obtenu

et le contenu réel de l'enregistrement. Parfois un transitoire n'est pas détecté là où il devrait y en avoir un et parfois c'est l'inverse. Pour obtenir une indexation rigoureuse il est important de donner accès à des outils d'édition post-analyse afin de corriger ces résultats.

Pour bien penser notre outil d'indexation il est important de le concevoir en tenant compte des éléments listés ci-dessus. Dans ce paragraphe nous illustrons notre démarche avec des fonctions de détections de transitoires, mais les éléments que nous citons ici sont nécessaires pour toutes les autres méthodes d'indexation automatique.

Afin d'illustrer notre propos, nous détaillons dans la suite de ce chapitre deux scénarios d'indexation automatique basés sur des méthodes de détection de transitoires différentes. Le but est de montrer comment les différents éléments listés ci-dessus permettent de développer un outil d'indexation approprié au système que nous souhaitons développer.

#### **indexations de la *Gymnopédie n°1* d'Érik Satie par analyse du flux spectral :**

La *Gymnopédie n°1* est le premier morceau d'un ensemble de trois pièces pour piano solo composées par Érik Satie en 1888. Devenu populaire une dizaine d'années après leur première édition<sup>1</sup>, les *Gymnopédies* comptent aujourd'hui parmi les œuvres les plus célèbres du compositeur. Utilisée comme bande originale pour de nombreuses œuvres de la culture populaire comme le film *Corrina, Corrina* de Jessie Anderson ou encore le jeu vidéo *Persona 2 : Innocent Sin*, la *Gymnopédie n°1* est un morceau de piano dont l'air est connu par de nombreuses personnes. C'est ce côté populaire qui nous a poussé à choisir cet enregistrement. Nous souhaitons concevoir une interaction basée sur le contrôle temporel d'un enregistrement dont le rythme est connu par l'utilisateur. Ainsi les personnes qui utilisent notre système peuvent facilement parvenir à rejouer le rythme original de l'enregistrement dès la première utilisation. En étiquetant la position temporelle de chaque note/accord, on peut concevoir une interaction basée sur le contrôle temporel de l'enregistre-

---

1. source : [https://fr.wikipedia.org/wiki/Gymnopédies\\_\(Satie\)](https://fr.wikipedia.org/wiki/Gymnopédies_(Satie))

ment "note par note".

Le son de piano se caractérise par des transitoires percussifs et peut être détecté efficacement par les méthodes basées sur le calcul du flux spectral qui correspond à la somme trame par trame de la différence des amplitudes de chaque coefficient d'une transformée de Fourier à court terme :

$$SF(n) = \sum_{k=-N/2}^{N/2-1} H(|X(n, k)| - |X(n-1, k)|) \quad (2.1)$$

Avec  $X$  l'amplitude du spectre pour une trame  $n$  et une fréquence  $k$ ;  $N$  la taille de la fenêtre glissante et  $H$  un redresseur simple de telle sorte que  $H(x) = \frac{x-|x|}{2}$ . [21]

Comme présenté figure 2.6, on observe des pics d'amplitude au niveau des transitoires de chaque note ou groupe d'accords joués. Ces transitoires indiquent le début de chaque événement rythmique constituant l'enregistrement.

L'enregistrement utilisé est celui du pianiste Pyotr Godunov<sup>2</sup>. Cet enregistrement est très peu bruité, il n'y a donc pas besoin d'appliquer un traitement particulier pour différencier le contenu utile du signal de celui du bruit de fond.

Pour réaliser l'indexation de cet enregistrement de la *Gymnopédie* nous utilisons la méthode de flux spectral [6] implémentée dans la librairie `essentia.js` [10]. Dans cette méthode une fonction permet de paramétrer l'écart temporel minimum entre deux transitoire d'attaque consécutifs. Cela nous permet de filtrer les détections de transitoires trop proches correspondant aux notes d'un même accord. Sauf dans le cas d'un accord joué de façon déliée, les notes d'un accord doivent être jouées au même moment. Dans notre cas nous réglons ce paramètre à 100 ms pour qu'il n'y ait qu'une position temporelle par événement rythmique comme représenté figure 2.6.

La possibilité de paramétrer ce filtre peut permettre d'obtenir des résul-

---

2. L'enregistrement utilisé ici est une interprétation de la *Gymnopédie n°1* par le pianiste Pyotr Godunov <https://www.youtube.com/watch?v=eMnxjdGK4w>

tats d'indexation plus "musicaux". Si l'on connaît en amont le tempo de la pièce que l'on souhaite rejouer on peut régler ce paramètre pour obtenir une indexation selon différentes divisions rythmiques.

Dans le cas de l'enregistrement de la *Gymnopedie n°1* utilisé en exemple, le tempo à la noire est de  $62BPM$ . La durée d'un temps est donc de  $62/60 = 0.97secondes$ . La signature rythmique du morceau étant  $3/4$  la durée d'une mesure est de  $3 * 0.97 = 2.91secondes$ . Le morceau ne contient que des divisions rythmiques supérieures à la croche qui représente ici un demi-temps, soit 0,48 seconde. Si on configure un filtrage pour n'avoir que des transitoires consécutifs espacés au minimum de  $2.91 - 0.5 = 2.41secondes$  on obtient alors une indexation à la mesure comme celle représentée figure 2.6. Ainsi on peut appliquer cette logique pour filtrer toute sorte de divisions rythmiques.

### **indexation d'un enregistrement de la *Symphonie n°5* de Ludwig van Beethoven pour créer du "gel" audio :**

Les créations de PUCE MUSE sont souvent basées sur le contrôle temporel de séquences préenregistrées. Des transformations pour "geler" le son sont fréquemment utilisées notamment dans l'application Mono-Ovisir dont l'interface est présentée figure 1.13. Le "gel" audio est un effet qui donne la sensation que le son se "fige" sur une portion de l'enregistrement.

Pour réaliser cette transformation, il existe plusieurs méthodes basées sur des transformations temporelles ou spectrales du son. Ici on propose une méthode de gel basée sur des transformations temporelles. Un système dans lequel a été implémentée cette méthode est présenté dans la vidéo dont le lien est en bas de page.<sup>3</sup>

Certaines fonctions de détections de transitoire permettent de mettre en évidence les brusques variations spectrales d'un enregistrement. Comme montré figure 2.6, ces fonctions présentent des maximums d'amplitude lorsqu'une nouvelle note est jouée. Les minimums quand à eux indiquent les zones avec peu d'évolution spectrale. La courbe de nouveauté est une fonction [29] qui permet d'observer de telles évolutions. Voici sa définition :

---

3. Dans cette vidéo on voit une démonstration du système de "gel" audio décrit ici appliqué sur les points d'orgues du premier mouvement de la *Symphonie n°5* de Ludwig Van Beethoven : <https://www.youtube.com/watch?v=QAodPOKaqxw>

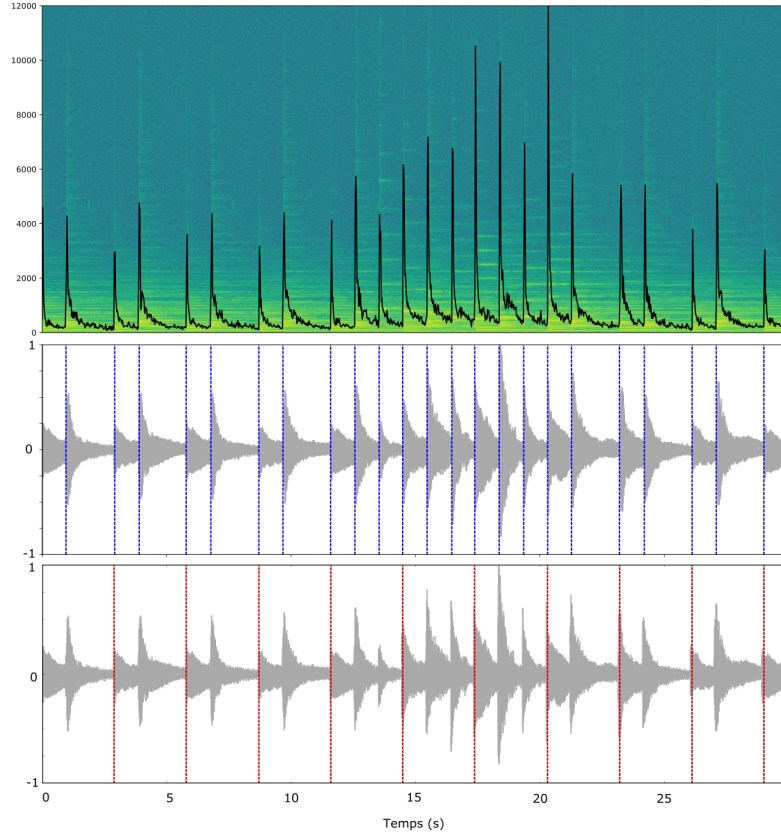


FIGURE 2.6 – Détection des transitoires d’attaque basée sur la détermination du flux spectral [6] sur un enregistrement de la *Gymnopédie n°1* composée par Érik Satie. En paramétrant le filtre des écarts temporels entre deux transitoires consécutifs, on peut obtenir une indexation selon différentes divisions rythmiques : ici au rythme (en bleu) ou à la mesure (en rouge)

$$\Delta(t) =: \sum_{k=1}^K |Y(k, t+1) - Y(k, t)|_{\geq 0} \quad (2.2)$$

avec  $|x|_{\geq 0} := x$  si  $x$  est un nombre réel non-négatif,  $|x|_{\geq 0} := 0$  si  $x$  est un nombre réel négatif.  $Y =: \log(1 + 1000 \cdot |X|)$  et  $X$  l’amplitude du spectre pour une fréquence  $k$  à la trame  $t$ . En appliquant un logarithme sur l’amplitude du spectrogramme, on accroît la dynamique des valeurs prises par cette courbe. Ainsi les variations spectrales sont plus marquées et les zones où le spectre évolue peu aussi. En appliquant une fonction de seuillage, on récupère les valeurs de  $t$  pour lesquelles  $\Delta(t) < S$  avec  $S$  une valeur de seuil déterminée



par lecture graphique.

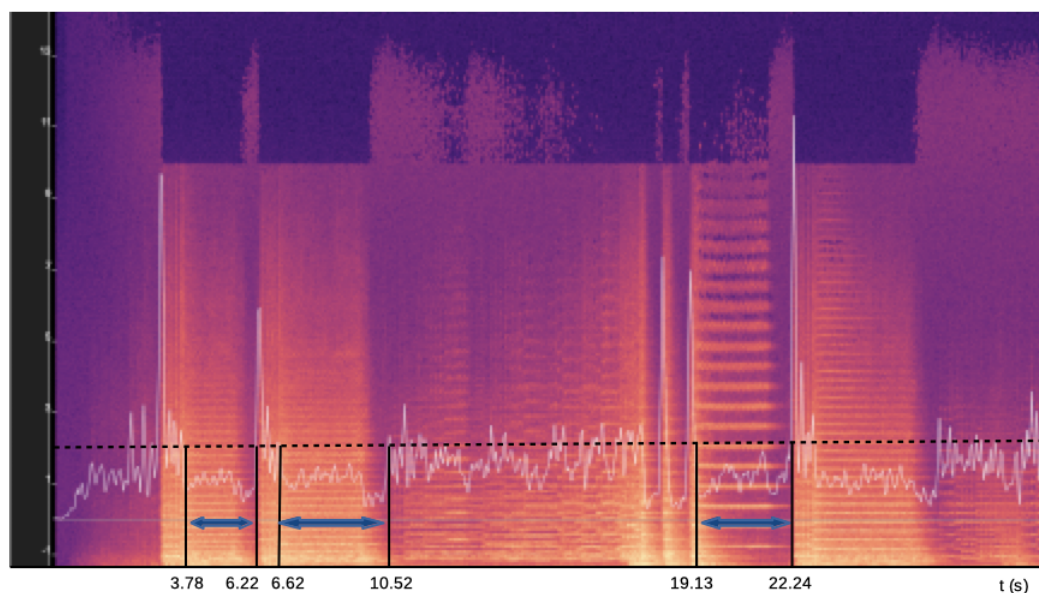


FIGURE 2.7 – Tracé de la courbe de nouveauté [29] sur un enregistrement de l'ouverture de la 5<sup>ème</sup> symphonie de Ludwig van Beethoven. Les plages de minima (zones fléchées) de la courbe de nouveauté représentent les zones où il y a peu d'évolution spectrale. Dans le cas d'un enregistrement de l'ouverture de la *Symphonie n°5* de Ludwig van Beethoven, ces zones représentent les notes tenues par l'orchestre ou des silences.

Sur la figure 2.7 on peut observer les zones détectées par la méthode décrite précédemment sur un enregistrement de la *Symphonie n°5* de Ludwig van Beethoven. Le réglage de l'algorithme qui trace la courbe de nouveauté permet de mettre en évidence les parties tenues des notes de ce passage. Pour obtenir un effet de gel, on peut tout simplement boucler la lecture de l'enregistrement sur ces zones. Ainsi un système de trois boucles a été mis en place. Ces trois boucles sont de tailles différentes. On les joue simultanément en contrôlant l'enveloppe d'amplitude selon une fenêtre de Hann. En divisant l'amplitude générale par la somme des trois valeurs d'enveloppe à l'instant  $t$ , on supprime l'impact de ces enveloppes sur le niveau final.

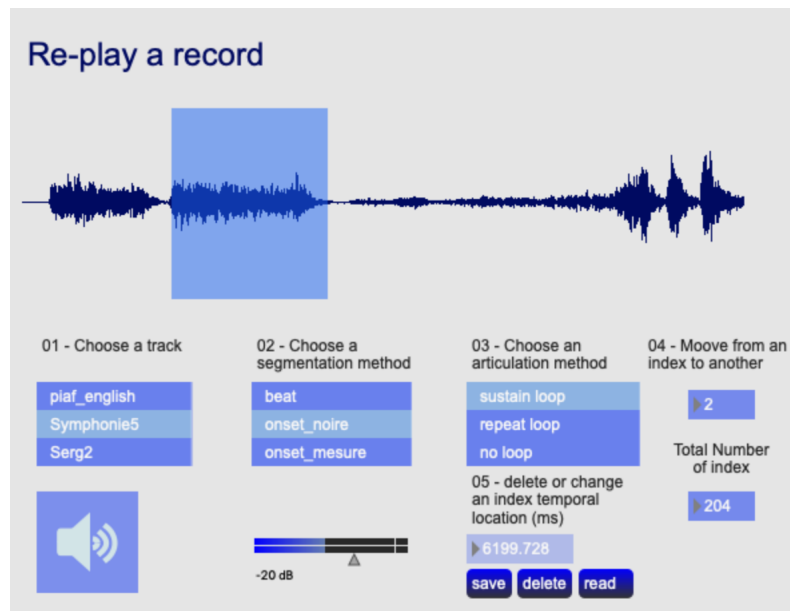


FIGURE 2.8 – Interface graphique d'un système permettant de produire un effet de gel audio sur les zones de minima de la courbe de nouveauté. Un lien vers une vidéo de démonstration de ce système est disponible en bas de page p.52

## 2.2.2 Indexation de pulsation

"La pulse (ou tactus) est une pulsation périodique induite par la perception qui se manifeste souvent par l'action de "taper du pied" sur la musique. " <sup>4</sup>. Dans la plupart des musiques "pulsées", la pulsation correspond à la battue du tempo que l'on exprime numériquement en BPM.

Le contrôle du tempo est une donnée importante de l'expressivité musicale. Le rubato, par exemple, consiste à faire varier le tempo d'une phrase musicale de façon continue afin de la souligner. Cet effet, fortement popularisé par les musiciens romantiques dont notamment Frédéric Chopin et Franz Liszt, est aujourd'hui très souvent utilisé. Intégrer un système permettant d'étiqueter un enregistrement selon sa pulsation permettrait d'intégrer ce type d'effet. Si "la pulsation périodique induite par la perception" d'un enregistrement est au cœur d'un scénario d'interaction, il est évident que les possibilités expressives du système deviennent nombreuses. Pour accéder à une telle information, il

4. "The beat (or tactus) is a perceptually induced periodic pulse that is best described by the action of "foot-tapping" to the music" citation extraite de la thèse de Tristan Jehan, *Creating Music by Listening*, MIT, Massachusetts, 2000, page 55

faut préalablement étiqueter un enregistrement selon sa pulsation.

Parmi les systèmes de contrôle temporel déjà existants, ceux de DJing intègrent souvent des systèmes d'indexation de pulsation. Cela permet de régler la vitesse de lecture d'un enregistrement compte tenu d'une valeur de tempo. Ainsi le DJ peut synchroniser la pulsation de deux morceaux au moment de leur transition.

Très souvent utilisés en informatique, les algorithmes de détection de pulsation ont été considérablement améliorés durant ces vingt dernières années. Parmi les méthodes existantes nous nous intéresserons ici à la méthode mise au point par José R. Zapata et al. [59] car elle est conçue pour tenir compte des différentes propriétés d'un son à l'autre. Effectivement les algorithmes de détection de pulsation nécessitent au préalable la position temporelle des transitoires d'attaque. Or chaque son possède un transitoire qui s'exprime physiquement d'une façon différente. Il serait donc réducteur de vouloir utiliser un unique algorithme de détection de transitoire en entrée de la méthode de détection de pulsation. Cette méthode calcule préalablement 8 fonctions de détection de transitoires différents avec laquelle il détermine, pour chacune d'entre elles, la position des pulsations selon l'algorithme de Degara [20]. Ces méthodes de détections sont celles dites du flux spectral [37], de la déviation de phase [22], du domaine complexe [5], de la déviation de phase pondérée [4], du domaine complexe rectifié [59] et par sélection de transitoire [4]. Une fois les résultats obtenus, il les compare selon leur coefficient de similarité. Celle qui a le coefficient de similarité le plus élevé est retenue. Pour illustrer notre propos nous présentons dans la suite de ce paragraphe un cas d'usage d'une indexation à la pulsation selon l'algorithme Multi-feature Beat tracking [59]. Nous montrons en quoi le paramétrage de cette fonction joue un rôle important dans la conception de notre scénario.

### **Indexation à trois niveaux d'un enregistrement multipiste de *Superstition* composé par Stevie Wonder**

Dans ce paragraphe nous détaillons l'indexation d'un enregistrement multipiste dans le cadre d'une pratique collective du contrôle rythmique. Nous ne nous focaliserons ici que sur les étapes d'indexation nécessaires pour rendre

possible ce scénario de contrôle temporel. Un détail plus général de l'installation est décrit à la section 4.1.2. Ce dispositif permet à chaque participant de contrôler la vitesse et la position de lecture d'une piste appartenant à un même enregistrement. En jouant toutes les pistes ensemble le résultat obtenu est l'enregistrement dans sa version masterisé. L'enregistrement multipiste choisi ici est la version du morceau de funk rock *Superstition*, composé par Stevie Wonder et Jeff Beck, présente sur l'album *Talking Book*. Cette version est disponible au format stem. Cela consiste en un fichier regroupant les différents groupes instrumentaux de l'enregistrement en pistes séparées. Ainsi on a accès à la piste de batterie, la piste de voix, la piste de basse, la piste pour la section cuivre et la piste pour les claviers.

Le format stem a été créé pour que les DJ puissent jouer séparément chaque partie instrumentale d'un même enregistrement. Dans notre cas, les musiciens rejouent chacun une des pistes qui composent ce fichier pour faire entendre la version complète du morceau.

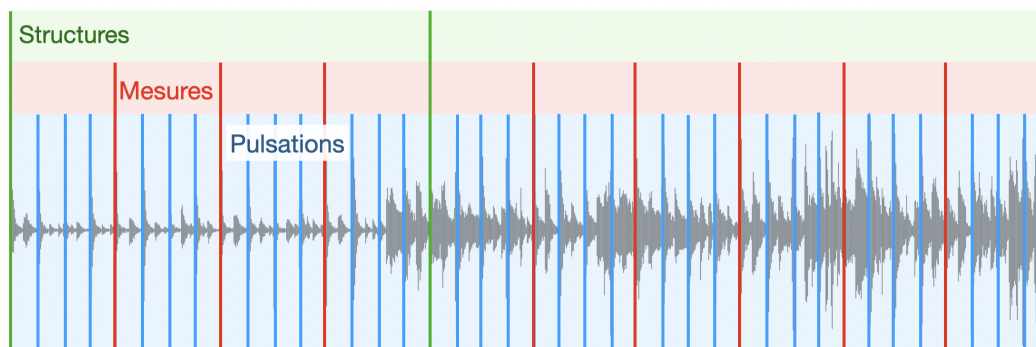


FIGURE 2.9 – Indexation d'un enregistrement du morceau *Superstition* composé par Stevie Wonder. Les pulsations sont représentées en bleu, les mesures sont représentées en rouge et les structures en vert. La première structure représente l'introduction à la batterie. La deuxième structure indique le début du premier couplet.

L'indexation que nous détaillons ici permet à posteriori de se déplacer de façon synchronisée dans chaque piste qui compose le format stem. Elle permet aussi à l'utilisateur de disposer de repères temporels nécessaires pour se placer rapidement dans une section de l'enregistrement en cas de désynchronisation. Une représentation de cette indexation est disponible figure 2.9. Elle est organisée comme suit :

- *La pulsation* : ici on détermine la position des pulsations à l'aide de

l'algorithme Multi-Feature Beat Tracking [59]. Comme vu précédemment, il existe une multitude de façons différentes de déterminer leur position compte tenu de la nature des sons contenus dans l'enregistrement. En utilisant 8 méthodes de détections de transitoire différentes on balaye les différentes fonctions qui permettent de détecter ce phénomène et ainsi sélectionner celle qui fonctionne le mieux pour le son étudié. Dans le cadre de notre exemple, on souhaite déterminer une indexation qui permettrait à plusieurs joueurs d'interagir avec les différentes pistes d'un même enregistrement de façon synchronisée. Il est donc important que chaque joueur ait le même découpage temporel et c'est pour cela que l'on choisit la piste de batterie comme référence pour le découpage en pulsation de chaque piste constituant le fichier multipiste. La piste de batterie étant constituée essentiellement de transitoires percussifs, c'est la fonction de détection par différence spectrale qui présente la meilleure ressemblance mutuelle selon la méthode de détection de pulsation "Multi-feature Beat tracking". À l'aide de cette méthode, on obtient une indexation de l'enregistrement répartie selon les durées présentées dans l'histogramme 2.10.

- *La mesure* : lorsque l'on souhaite synchroniser le déroulement temporel de plusieurs enregistrements, il est important de pouvoir déplacer rapidement leur position de lecture selon différents niveaux temporels. Le premier niveau vu précédemment permet de se mouvoir dans l'enregistrement selon la pulsation. Pour permettre à l'utilisateur de rattraper facilement son retard par rapport aux autres joueurs, on propose un autre niveau d'indexation, dit "à la mesure". Pour réaliser cette indexation, on choisit une valeur correspondant au nombre de segments du premier niveau inclus dans ce deuxième niveau d'indexation. Pour le cas de *Superstition* on règle cette valeur à 4 ce qui veut dire que l'on crée une indexation à la mesure toutes les 4 pulsations. Ici le choix de cette valeur s'est fait en fonction de la signature rythmique du morceau, de façon à obtenir un découpage cohérent avec la partition initiale.
- *La structure* : Ce dernier niveau d'indexation permet à l'utilisateur de

placer des ancres temporelles sur des éléments de la structure musicale du morceau. Dans le cadre de l'indexation de l'enregistrement multi-piste de *Superstition* cette indexation "à la structure" est effectuée manuellement et indique le début de l'introduction, du premier couplet, du premier refrain etc. contenu dans l'enregistrement. Dans l'utilisation collective que l'on décrit ici, ce type d'indexation permet de donner des rendez-vous au début d'une partie musicale pour déclencher la lecture d'une piste de l'enregistrement. Dans le cadre d'une performance de DJ, ce type d'indexation permet de déclencher le début d'un enregistrement directement sur le début d'un refrain, par exemple pour créer des transitions personnalisées.

Cette indexation est réalisée entièrement depuis la première analyse selon la méthode Multi-feature Beat tracking [59]. La possibilité de pouvoir paramétrer et corriger le résultat nous a permis d'obtenir ces trois niveaux d'indexations.

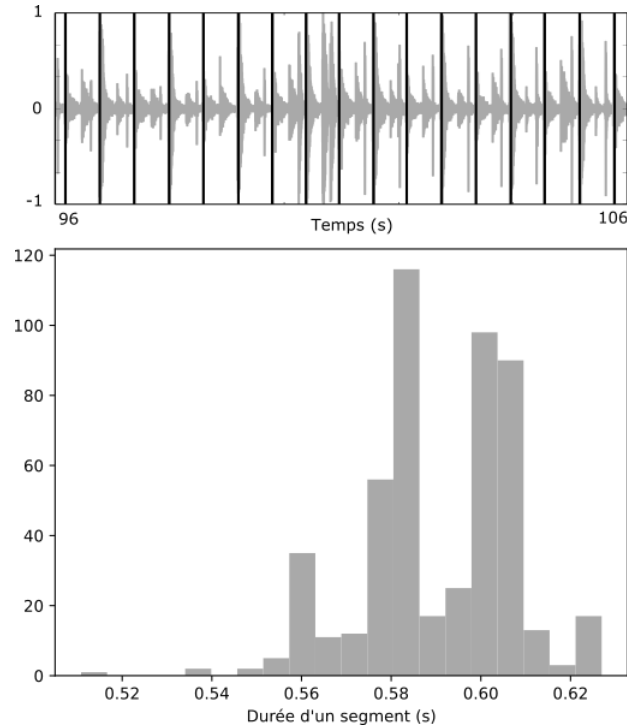


FIGURE 2.10 – Histogramme de répartition des durées de segment obtenues par détection de pulsation selon [59]

### 2.2.3 Indexation manuelle

Comme nous l'avons vu dans la sous-section précédente, il y a plusieurs avantages à ce que l'étape d'indexation soit assistée par des algorithmes de MIR (Musical Information Retrieval). Pour obtenir les meilleurs résultats, ces algorithmes nécessitent un paramétrage précis qui demande une connaissance de l'enregistrement analysé en amont. Néanmoins, il arrive souvent que le paramétrage de nos outils d'analyse ne suffise pas pour obtenir les fichiers d'indexation souhaités. C'est le cas par exemple des détections de transitoires trop nombreuses dans *Accidents/Harmoniques* qui sont visibles sur la figure 2.5. Il faut alors corriger le résultat obtenu par analyse pour obtenir la bonne indexation. C'est aussi le cas de l'indexation à la pulsation de la *Gymnopédie n°1*. Certains rallentandos sont mal interprétés par l'algorithme Multi-feature Beat tracking [59], il est alors nécessaire de revenir corriger à la main ces portions. Il est donc important d'intégrer des outils d'indexations manuelles pour corriger les résultats obtenus par détection automatique. Il faut pouvoir ajouter, effacer et corriger chaque ancre temporelle déterminée par l'analyse.

L'indexation manuelle est aussi une solution dans le cas où les éléments que l'on souhaite déterminer ne sont pas détectables par des algorithmes. Dans le système T-Voks que nous décrivons à la section 1.2.5 par exemple, l'indexation de syllabe se fait de façon manuelle. Il existe des méthodes pour déterminer automatiquement la position de syllabe dans un enregistrement [52] mais elles nécessitent des entrées supplémentaires comme la langue parlée, le sexe de l'interlocuteur ce qui complexifie un peu le système.

L'indexation manuelle peut aussi offrir des solutions dans le cas où les informations que l'on souhaite étiqueter sont composites, c'est-à-dire qu'elles ne désignent pas toujours le même élément sonore. Par exemple, dans le scénario de contrôle temporel de l'ouverture de la *Symphonie n°5* de Ludwig van Beethoven que l'on décrit à la section 2.2.1, on marque des temps de gel audio sur les points d'orgue de la partition. Pour déterminer les zones avec suffisamment peu d'évolution spectrale, on emploie un algorithme qui trace la courbe de nouveauté [59]. Une fois ces zones détectées, il nous faut aussi placer des ancres temporelles au début de chaque phrase qui précède le point d'orgue comme dans la vidéo montrée en bas de page 56 . Pour les ajouter on utilise

un algorithme spécifique. On ajoute manuellement le début de chaque phrase dans le fichier d'indexation. L'indexation globale de l'enregistrement est alors composite. Elle englobe à la fois une indexation automatique des zones de faible évolution spectrale et une indexation manuelle des débuts de phrase.

On comprend bien, à travers ces exemples, l'importance d'avoir accès à des outils d'édition manuelle pour corriger ou créer un fichier d'indexation adéquat avec le scénario de contrôle rythmique que l'on souhaite concevoir. Ces outils d'indexation manuelle peuvent être simples comme celui de Mono-Replay [36] qui permet de retirer ou d'ajouter des ancrs temporelles de même nature sur un fichier audio. Ils peuvent aussi être beaucoup plus complexe comme ceux disponibles dans le logiciel Eanalysis [12] qui permet de choisir la forme, la couleur et la durée de la représentation visuelle de cette indexation. Ce type d'indexation est visible sur la figure 2.2.

## 2.3 Contrôle temporel

L'un des enjeux de ce travail de recherche est de présenter des fonctions qui permettent au développeur d'application musicale interactive de concevoir et d'améliorer facilement des systèmes basés sur le contrôle temporel d'un enregistrement. Après avoir présenté les fonctionnalités nécessaires pour créer une indexation adaptée à ce type d'application, nous abordons ici l'organisation des modes de contrôle gestuel. Comme nous l'avons introduit dans la section 2.1, ce module prend en entrée les données captées par une interface gestuelle et les transforme en paramètres de contrôle pour la synthèse. Ces paramètres de contrôle peuvent temporels ou spectraux. Nous proposons ici de les exprimer sous forme mathématique.

Dans les systèmes que nous souhaitons modéliser le contrôle rythmique désigne la fonction qui permet de transformer un geste en position de lecture. Cette fonction se situe entre l'interface gestuelle externe et le moteur audio du système. L'enjeu ici est d'exprimer différents modes de contrôle temporel avec les mêmes variables. En les exprimant de la sorte on facilite leur intégration, leur gestion ainsi que l'ajout de nouveaux modes, notamment dans le cas d'un système comme Mono-Replay [36] qui en intègre 11.



Nous présentons dans un premier paragraphe plusieurs systèmes de contrôle rythmique déjà existants afin de souligner leurs similarités et leurs différences. Nous verrons qu'ils génèrent tous un même indice temporel correspondant à la portion de l'enregistrement qui est synthétisée. Nous exprimons ensuite plusieurs modes de contrôle temporel en fonction de ce paramètre.

### 2.3.1 Contexte



FIGURE 2.11 – Le phonogène chromatique (*Source : inagrm.com*)

Comme vu dans la section 1.2 le GRM a permis de concrétiser les pensées artistiques du début du XXe siècle portant sur la manipulation temporelle du support musical. Ses inventions ont permis à des artistes de contrôler la vitesse et la position de lecture d'enregistrement pour intégrer de nouveaux sons dans leur composition. Ces nouveaux instruments ont ensuite servi de modèle pour le développement de nombreuses autres interfaces d'expression musicale.

Le Phonogène chromatique est une des inventions du GRM. Représenté figure 2.11, il est reconnu aujourd'hui comme l'ancêtre analogique du sampler actuel. Il permet de contrôler la vitesse de lecture d'une bande audio en utilisant les touches d'un clavier. Si on modélise ce système mathématiquement, il peut être représenté sous la forme d'une fonction qui prend en entrée un contrôle discret, l'enfoncement d'une touche de clavier, et qui génère en sortie une position dans l'enregistrement. Cette position varie de façon plus ou moins

rapide compte tenu de la touche qui est enfoncée. Cette représentation très schématique se retrouve dans la plupart des systèmes de contrôle rythmique récents.

Le célèbre sampler 404SX, développé par la société Roland, peut schématiquement être représenté de la même façon. Un contrôle gestuel discret est capté en entrée du système lorsque l'utilisateur presse une des touches présentées figure 2.12. Une fois la touche enfoncée, le système indique une position de lancement dans l'échantillon et une vitesse de lecture. La position de lecture dans l'échantillon évolue de façon continue tant que la touche reste enfoncée. La lecture s'arrête lorsque la touche n'est plus enfoncée. Du point de vue du contrôle temporel, on peut souligner quelques différences entre le phonogène chromatique et le sampler Roland 404SX. D'un côté le geste de l'utilisateur permet d'ajuster la vitesse de lecture de l'enregistrement, de l'autre il permet de définir la position initiale de lecture. Néanmoins, dans les deux cas le geste génère le défilement de la position de lecture.

De façon générale les outils faits pour la manipulation temporelle d'un enregistrement sont toujours constitués d'un module de contrôle temporel qui convertit un geste continu ou discret en un paramètre temporel de la lecture. D'un système à l'autre ces opérations sont différentes mais elles peuvent toutes s'exprimer selon le même indice temporel correspondant à la position à synthétiser dans l'enregistrement.



FIGURE 2.12 – Le SP404-SX est un échantillonneur développé par la société Roland en 2009. Il permet de lire différents échantillons sonores d'un même enregistrement en tapant sur des pads. Il est devenu un instrument de référence pour les producteurs de Hip-Hop qui pratiquent le sampling et le finger-drumming

### 2.3.1.1 Expression mathématique des modes de contrôle temporel

Tous les systèmes décrits dans la section précédente peuvent se résumer selon le formalisme suivant : pour un enregistrement d'une durée  $T$ , ils déterminent, pour tout instant  $t$ , un indice temporel  $\tau(t)$  dans l'intervalle  $[0, T]$  qui correspond à la position lue dans l'enregistrement audio.

L'indice temporel  $\tau(t)$  fait référence à la double échelle temporelle qui existe lors d'une performance de type manipulation de support enregistré[32] : d'un côté s'écoule le temps  $t$  de la "performance" qui mesure en continu la durée de l'interaction et de l'autre le temps de la "synthèse" qui indique la position temporelle dans l'enregistrement qui est re-synthétisé. L'indice temporel  $\tau(t)$  [57] [16] [34] correspond à ce temps de synthèse. On peut le représenter sous la forme d'une fonction qui prend en entrée des contrôles gestuels temps réel  $t_i$ , captés lors d'une performance, ainsi que des données temporelles prédéterminées, comme la durée  $T$  de l'enregistrement ou les marqueurs temporels  $\lambda_i$  obtenus par indexation.

#### Les entrées temps réels :

Les entrées temps réel sont les données captées par une interface gestuelle pendant une performance. Elles peuvent être de deux types :

- *continues* : dans ce cas les entrées sont représentées sous forme de fonction  $f(t) \in \mathbb{R}^+$  .
- *discrètes* : dans ce cas les entrées sont représentées sous forme de séquence  $(t_i)_{i \in \mathbb{N}}$  de valeurs temporelles.

#### Les entrées prédéterminées :

Les entrées prédéterminées sont les valeurs récupérées en amont de la performance. Elles servent dans le calcul de l'indice temporel  $\tau(t)$  comme représenté figure 2.13 et figure 2.14. Ces valeurs peuvent être :

- La durée  $T$  de l'enregistrement manipulé.
- Une séquence de marqueurs temporels  $(\lambda_i)_{i \in \mathbb{N}}$  obtenue par indexation

de l'enregistrement. Dans notre travail ces marqueurs temporels renseignent sur la position d'éléments temporels saillants d'un enregistrement comme la pulsation ou le début de note.

- Une valeur qui renseigne sur un paramètre global de l'enregistrement comme par exemple la valeur du tempo, exprimée en battements par minute (BPM), ou encore la période correspondante  $T_{BPM}$ , exprimée en secondes, qui s'écoule entre deux pulsations :

$$T_{BPM} = \frac{60}{BPM} \quad (2.3)$$

### 2.3.2 Contrôle temporel continu

On dit qu'un mode de contrôle temporel est continu à partir du moment où il intègre une entrée temps réel continue. Ainsi à chaque instant  $t$  d'une performance, le geste de l'utilisateur a une incidence sur le calcul de l'indice temporel  $\tau(t)$ . Une vue schématique de ces modes est visible figure 2.13.

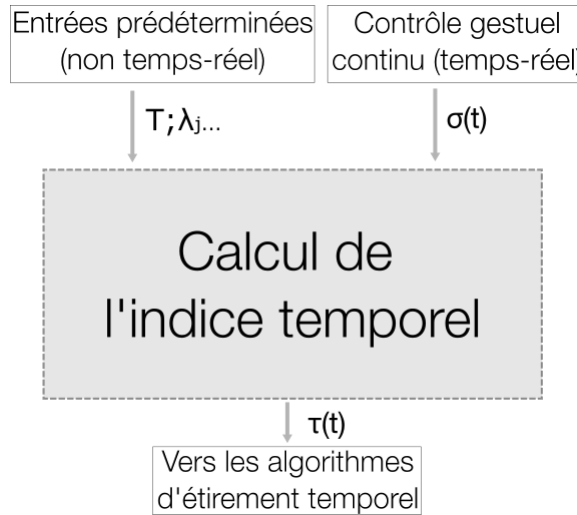


FIGURE 2.13 – Vue schématique d'un mode de contrôle rythmique continu. Pour établir la valeur de l'indice temporel  $\tau(t)$ , le système récupère des contrôles gestuels continus temps réel  $\sigma(t)$  ainsi que des valeurs prédéterminées non-temps réel comme la durée  $T$  de l'échantillon ou des marqueurs  $\lambda_i$  indiquant la position d'éléments rythmiques dans l'enregistrement.

Les modes décrits dans la suite de ce paragraphe utilisent des contrôles gestuels continus et temps réel ainsi que des valeurs prédéterminées pour calculer

à chaque instant  $t$  l'indice temporel  $\tau(t)$ .

#### **Contrôle continu de position :**

Ce mode permet de spécifier directement une position temporelle à synthétiser dans l'enregistrement initial. Il prend en entrée un paramètre de position  $p(t)$  dans l'intervalle  $[0, 1]$  et le multiplie par une durée  $T$  prédéterminée, communément la longueur de l'échantillon. On obtient alors la valeur de  $\tau(t)$  selon l'équation suivante :  $\forall t \in \mathbb{R}$ ,

$$\tau(t) = Tp(t) \quad (2.4)$$

Ce mode de contrôle rythmique est utilisé par les DJ pour scratcher. Pour le cas des platines vinyles, le musicien modifie la position de lecture en bloquant ou entraînant le vinyle sous la tête de lecture. Sur les systèmes analogiques le changement de vitesse induit un changement de pitch. Dans notre cas nous nous intéressons seulement au changement rythmique induit par ce mode.

#### **Contrôle continu de vitesse :**

Dans ce mode on modifie de façon continue la vitesse de lecture  $v(t)$  de l'enregistrement. L'indice temporel  $\tau(t)$  est la dérivée de cette valeur. On a donc :  $\forall t \in \mathbb{R}$ ,

$$\tau(t) = \int_0^t v(u) du \quad (2.5)$$

Ce mode de contrôle rythmique est aussi couramment utilisé par les DJ pour accélérer ou ralentir la vitesse de lecture d'un morceau. Sur de nombreux modèles de platine un fader permet de régler cette vitesse.

### **2.3.3 Contrôle temporel discret**

Les modes de contrôles discrets calculent  $\tau(t)$  à partir de contrôles discrets temps réel  $(t_i)_{i \in \mathbb{N}}$  et d'entrées prédéterminées non-temps réel. Une représentation schématique de ces modes est visible figure 2.14.

Les entrées prédéterminées non-temps réel sont les mêmes que celles des modes de contrôle temporel continus. Elles peuvent indiquer par exemple la durée  $T$  de l'enregistrement ou encore des marqueurs temporels dans celui-ci

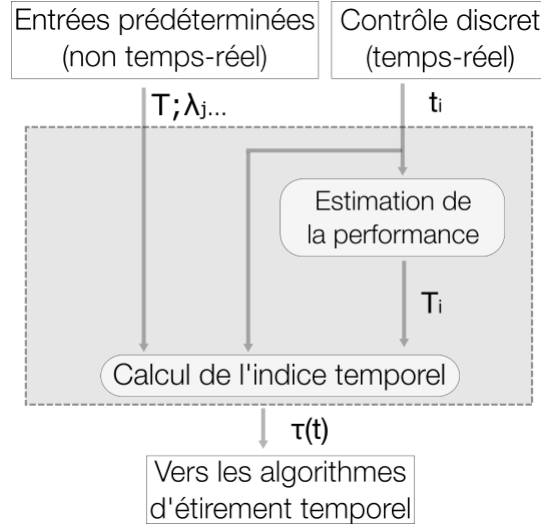


FIGURE 2.14 – Vue schématique d’un mode de contrôle rythmique discret. Pour établir la valeur de l’indice temporel  $\tau(t)$ , le système récupère des contrôles gestuels discrets  $t_i$  ainsi que des valeurs prédéterminées non-temps réel comme la durée  $T$  de l’échantillon ou des marqueurs  $\lambda_i$  indiquant la position d’éléments rythmiques dans l’enregistrement

$(\lambda_i)_{i \in \mathbb{N}}$ .

Dans les modes que nous décrivons ensuite, la liste de marqueurs  $\lambda_i$  indique les pulsations du morceau. Ici nous tenons compte des précédentes valeurs de contrôle discret  $(t_i)$  pour calculer  $\tau(t)$ . Le calcul de  $\tau(t)$  prend donc en plus des deux entrées décrites précédemment, une entrée  $(T_i)$  qui tient compte des précédentes performances. Dans notre cas ces entrées  $(T_i)$  estiment l’écart temporel entre plusieurs pulsations lors d’une performance. Il existe de très nombreuses façons d’estimer ces valeurs. Nous en présentons quelques-unes ici :

**Dernière pulsation :** La durée estimée de la pulsation battue par l’utilisateur est la même que celle de la dernière pulsation battue :  $\forall i \in \mathbb{N}$ ,

$$T_i = t_i - t_{i-1} \quad (2.6)$$

**Moyenne glissante :** La durée estimée de la pulsation battue par l'utilisateur est la moyenne des durées sur les  $N$  dernières pulsations battues :  $\forall i \in \mathbb{N}$ ,

$$T_i = \frac{t_i - t_{i-N}}{N} \quad (2.7)$$

Avec  $N$  un entier positif préalablement déterminé. Lorsque  $N$  vaut 1 la durée estimée de la pulsation est la même que celle de la dernière pulsation.

**Moyenne globale :** La durée estimée de la pulsation battue par l'utilisateur est la moyenne sur toutes les précédentes pulsations battues :  $\forall i \in \mathbb{N}$ ,

$$T_i = \frac{t_i - t_0}{i} \quad (2.8)$$

Dans cette méthode, les dernières performances ont de moins en moins d'influence sur le calcul du tempo battu par l'utilisateur à mesure que les itérations de  $i$  évoluent.

Cette méthode d'estimation est la plus couramment utilisée dans les outils d'estimation de pulsation disponibles en ligne<sup>5</sup> et dans les DAW. Elle est adaptée pour atteindre une valeur cible de tempo sur un grand nombre d'itérations.

**Simple filtre passe bas :** La durée estimée de la pulsation battue par l'utilisateur est la somme pondérée des dernières durées de pulsation estimées et de la dernière mesurée :

$$\forall i \in \mathbb{N},$$

$$T_i = (1 - a)(t_i - t_{i-1}) + aT_{i-1} \quad (2.9)$$

où  $a$  est un paramètre à choisir entre 0 et 1. Cela correspond à un filtre passe bas numérique appliqué sur les valeurs de pulsation estimées.

Cette équation peut être réécrite de cette façon :

$$T_i = (1 - a) \sum_{k \geq 0} a^k (t_{i-k} - t_{i-k-1}) \quad (2.10)$$

---

5. <https://www.all8.com/tools/bpm.htm>

Plus la valeur de  $a$  est proche de zéro et plus le système est réactif aux changements de tempo de l'utilisateur. À l'inverse, plus la valeur de  $a$  est proche de 1, et plus le système sera robuste aux variations de tempo lors de la performance de l'utilisateur. Lorsque  $a = 0$  la durée estimée de la pulsation battue par l'utilisateur est la même que celle de la dernière pulsation battue.

Dans les sections suivantes nous présentons plusieurs méthodes permettant de calculer à chaque instant  $t$  l'indice temporel  $\tau(t)$  depuis un contrôle gestuel discret. On donne à chaque fois une représentation mathématique selon les variables décrites précédemment ainsi qu'une représentation graphique sur laquelle est tracée l'évolution de cet indice temporel.

### 2.3.3.1 Mode asynchrone linéaire

Dans ce mode de contrôle l'évolution de  $\tau(t)$  suit une rampe linéaire calculée pour chaque nouvelle valeur de  $T_i$ . Ici  $T_i$  représente l'écart entre deux valeurs discrètes consécutives  $t_i$ . Cet écart peut être moyenné ou filtré selon les méthodes vues précédemment. La rampe d'évolution de  $\tau(t)$  part de la valeur d'indice temporel à l'instant  $t_i$  et évolue selon le coefficient directeur  $\frac{T_{BPM}}{T_i}$ .  $T_{BPM}$  correspond à la période entre deux pulsations selon la valeur de tempo globale du morceau. Pour le cas du trio pour piano et corde n°2 de Franz Schubert, dont l'évolution de l'indice temporel selon le mode asynchrone linéaire est représenté figure 2.15, la valeur de tempo estimée par l'algorithme Multi-feature Beat tracking [59] est de 95.8 BPM. Alors  $T_{BPM} = 0.63$  seconde. Mathématiquement ce mode s'exprime de la façon suivante :  $\forall i \in \mathbb{N}$

$$\tau(t) = \tau(t_i) + \frac{T_{BPM}}{T_i}t \quad (2.11)$$

On dit que ce mode est asynchrone car les contrôles discrets en entrée ne sont pas synchronisés à une position particulière dans l'enregistrement. À chaque nouvelle valeur de  $T_i$ , seule la vitesse évolue de façon discrète. Il n'y a pas de discontinuité dans l'évolution de la position.

Ce mode a été implémenté dans le logiciel Mono-Replay que nous décrivons section 3. Il s'agit du mode "direct".



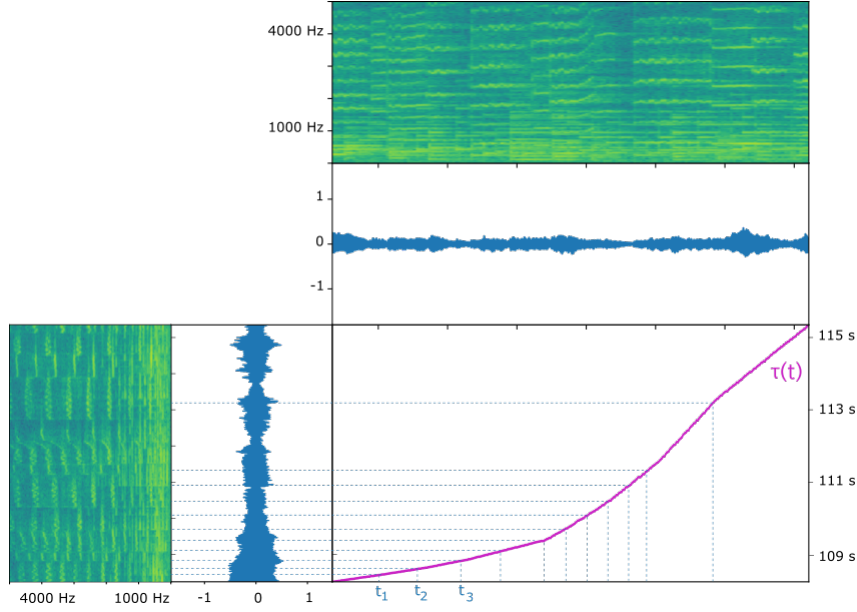


FIGURE 2.15 – Évolution de l'indice temporel  $\tau(t)$  (en violet) d'un enregistrement du Trio pour piano et corde n°2 de Franz Schubert selon le mode asynchrone linéaire. La forme d'onde à gauche représente l'enregistrement initial et la forme d'onde en haut représente l'enregistrement transformé. Quand le système reçoit un nouveau contrôle temporel discret  $t_i$  (en bleu) il calcule une nouvelle vitesse de lecture.

### 2.3.3.2 Mode asynchrone linéaire continu

Ce mode détermine une fonction linéaire pour calculer l'évolution de l'indice temporel  $\tau(t)$ . Cette fonction est la même que celle décrite précédemment à la différence que l'on remplace  $T_{BPM}$  par la différence  $\lambda_{\tau(t_{i-1})} - \lambda_{\tau(t_i)}$ .  $\lambda_{\tau(t_i)}$  correspond au marqueur temporel le plus proche de l'indice temporel  $\tau(t_i)$  au moment où l'utilisateur envoie le contrôle discret  $t_i$ .

Mathématiquement ce mode s'exprime de la façon suivante :  $\forall i \in \mathbb{N}$

$$\tau(t) = \tau(t_i) + \frac{\lambda_{\tau(t_{i-1})} - \lambda_{\tau(t_i)}}{T_i} t \quad (2.12)$$

Ce mode a été pensé pour tenir compte des rapides variations de tempo dans un enregistrement, dans la mesure où les marqueurs temporels  $\lambda_i$  représentent la position des pulsations dans un morceau. Dans le cas d'un rallentando par exemple les pulsations deviennent progressivement plus espacées. Il peut être intéressant d'intégrer ces variations dans un scénario d'interaction.

Ce mode a été intégré dans le logiciel Mono-Replay que l'on décrit à la section 3. Il porte le nom de "fit".

### 2.3.3.3 Mode synchrone linéaire

Dans ce mode à chaque fois que le système reçoit un nouveau point de contrôle  $t_i$  l'indice temporel  $\tau(t)$  suit une rampe partant du marqueur temporel  $\lambda_i$  jusqu'au marqueur temporel suivant  $\lambda_{i+1}$  en un temps  $t_i - t_{i-1}$ . Une fois que l'indice temporel atteint la valeur  $\lambda_{i+1}$  il reste constant jusqu'au prochain point de contrôle.

Lorsque  $\tau(t) \in [\lambda_i, \lambda_{i+1}]$ , si l'utilisateur envoie un point de contrôle avant que  $\tau(t)$  arrive au marqueur  $\lambda_{i+1}$ , l'indice temporel se synchronise directement sur  $\lambda_{i+1}$  et évolue selon une nouvelle rampe linéaire dont l'équation est décrite plus bas.

On dit que ce mode est synchrone car à chaque fois qu'il reçoit un nouveau point de contrôle il initialise sa position sur le marqueur temporel suivant.

Mathématiquement on peut écrire ce mode de la manière suivante :  $\forall i \in \mathbb{N}, \forall t \in [t_i, t_{i+1}]$ ,

$$\tau(t) = \begin{cases} \lambda_i + \frac{t-t_i}{T_i}(\lambda_{i+1} - \lambda_i) & \text{si } t < t_i + T_i \\ \lambda_{i+1} & \text{si } t \geq t_i + T_i \end{cases} \quad (2.13)$$

Ce mode de contrôle rythmique permet de faire durer la fin d'un segment audio avant de passer au segment suivant. Dans ce cas  $\tau(t)$  n'évolue plus jusqu'au prochain contrôle discret de l'utilisateur. Si on observe l'évolution de l'indice temporel d'un enregistrement de la *Gymnopédie n°1* représentée figure 2.16, on peut apercevoir un plateau un peu avant  $t_1$ . Il correspond au cas où l'utilisateur indique un nouveau contrôle gestuel après que  $\tau(t)$  atteigne la fin du segment en cours.

Par ailleurs, si  $\tau(t)$  est situé dans l'intervalle  $[\lambda_i; \lambda_{i+1}]$  et que l'utilisateur déclenche un point de contrôle avant que cet indice temporel n'atteigne la valeur cible  $\lambda_{i+1}$  alors  $\tau(t)$  subit une discontinuité et passe directement à la valeur cible suivante. Dans la figure 2.16 on peut observer cette discontinuité à  $t_5$ .

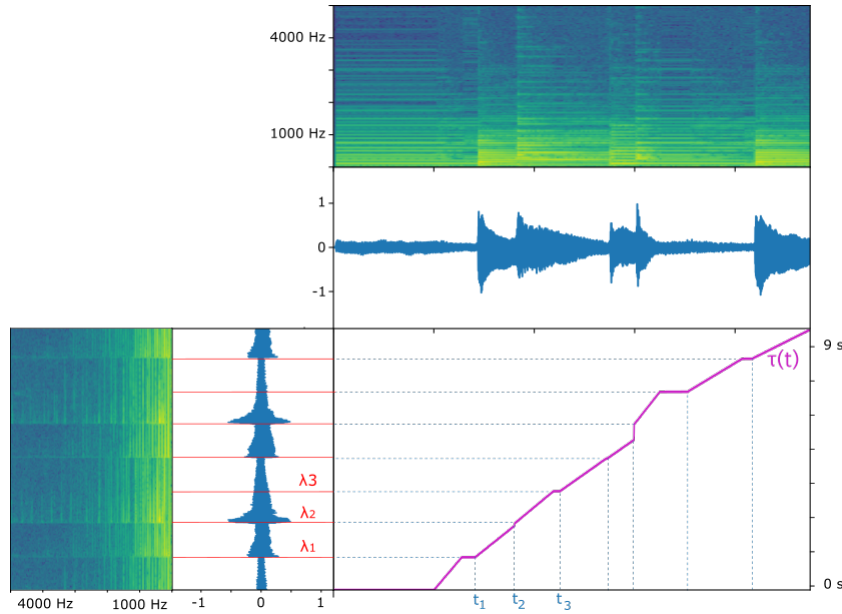


FIGURE 2.16 – Évolution de l'indice temporel  $\tau(t)$  (en violet) d'un enregistrement de la *Gymnopédie n°1* d'Erik Satie selon le mode synchrone linéaire. La forme d'onde à gauche représente l'enregistrement initial et la forme d'onde en haut représente l'enregistrement transformé. L'enregistrement est découpé en segments selon des marqueurs temporels  $\lambda_i$  (en rouge). Chaque marqueur indique une pulsation du morceau. Quand  $\tau(t)$  arrive sur un marqueur il reste à la même position. Il passe au segment suivant quand il reçoit un nouveau contrôle discret  $t_i$  (en bleu).

Dans le cadre d'une utilisation musicale comme le sampling, cette discontinuité peut être esthétiquement recherchée comme dans le cas d'usage du compositeur Pehoz présenté section 4.1.3. Cette discontinuité peut aussi être un avantage pour synchroniser le passage d'un segment à l'autre dans le cas d'une pratique collective comme celle décrite section 4.1.2. Néanmoins il existe beaucoup d'autres cas où la présence d'une discontinuité dans le déroulement d'un son peut s'avérer problématique. Dans le cas du déroulement d'une voix enregistrée, par exemple, cette discontinuité peut poser problème pour la compréhension globale de la phrase rejouée. [34]. Effectivement si l'on passe d'une valeur de  $\tau(t)$  à une autre sans avoir synthétisé l'intégralité de l'enregistrement contenu entre ces deux positions temporelles, il manquera une partie du fichier audio comme la fin d'un mot par exemple.

Ce mode a été intégré dans le logiciel Mono-Replay que l'on décrit à la section 3. Il porte le nom de "next".

#### 2.3.3.4 Mode synchrone parabolique

Dans ce mode l'indice temporel  $\tau(t)$  suit un profil parabolique d'évolution de la vitesse de lecture : La vitesse est plus rapide au début de la lecture du segment et de plus en plus lente à l'approche d'une valeur cible  $\lambda_i$  choisie parmi les marqueurs prédéterminés. Une fois que la valeur cible est atteinte l'indice temporel n'évolue plus et la lecture de l'enregistrement gèle à cette position temporelle.

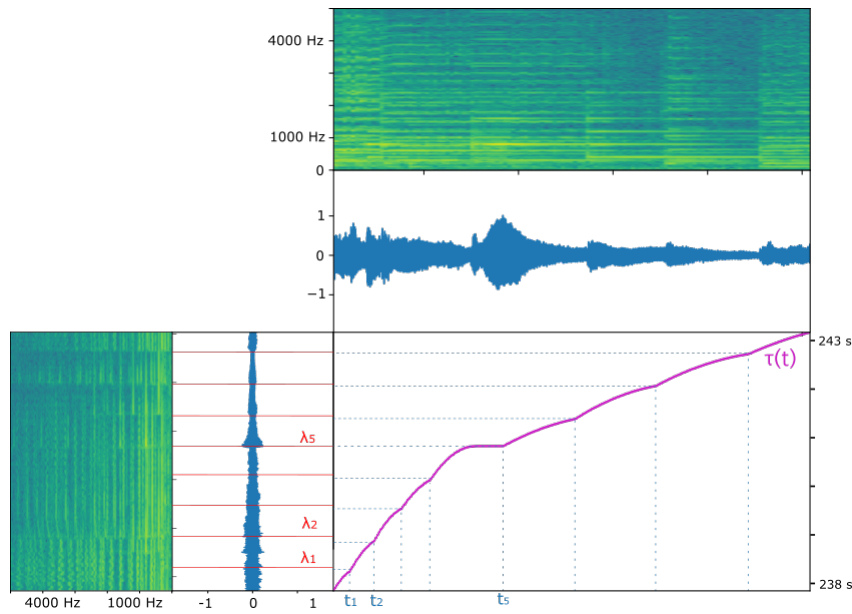


FIGURE 2.17 – Évolution de l'indice temporel  $\tau(t)$  (en violet) d'un enregistrement du trio pour piano et corde n°2 de Franz Schubert. La forme d'onde à gauche représente l'enregistrement initial et la forme d'onde en haut représente l'enregistrement transformé. À chaque contrôle discret  $t_i$  (en bleu) un nouveau profil convexe d'évolution de l'indice temporel  $\tau(t)$  est calculé. À chaque  $t_i$  l'indice temporel évolue vers une valeur cible  $\lambda_{i+1}$ . Lorsque la valeur cible est atteinte l'indice temporel n'évolue plus. On peut voir ce phénomène à  $\lambda_5$

Si l'utilisateur déclenche un contrôle discret avant d'avoir atteint la valeur cible  $\lambda_i$  alors la nouvelle valeur cible devient  $\lambda_{i+1}$ . Une nouvelle fonction convexe tenant compte de ce retard est ensuite calculée. On dit que ce mode est synchrone car à chaque contrôle discret  $t_i$  il initialise une nouvelle valeur cible. Ce mode se représente mathématiquement de la façon suivante :

$\forall i \in \mathbb{N}, \forall t \in [t_i, t_{i+1}],$

$$\tau(t) = \begin{cases} \tau(t_i) + (-\Theta^2 + 2.\Theta).(\lambda_{i+1} - \tau(t_i)) & \text{si } t < t_i + T_i \\ \lambda_{i+1} & \text{si } t \geq t_i + T_i \end{cases} \quad (2.14)$$

avec  $\Theta = \frac{t-t_i}{T_i}$  et  $T_i = t_i - t_{i-1}$

Ce mode de contrôle rythmique est intégré dans l'outil de synthèse vocale T-Voks qui permet de contrôler la synthèse de voix humaines pré-enregistrées à l'aide d'un thérémine [57]. Dans le cadre d'une utilisation musicale, la discontinuité peut être un effet artistique souhaité mais, pour le contrôle rythmique d'une voix de synthèse, il est important de garder la totalité de l'information phonétique de chaque mot afin que le mot prononcé soit entier et compris. Ce mode a été intégré dans le logiciel Mono-Replay que l'on décrit à la section 3. Il porte le nom de "parabolic".

## 2.4 Contrôle spectral

Les différents modes de contrôle temporel listés précédemment ont été intégrés dans un logiciel que nous présentons à la section 3. Pour augmenter l'expressivité de ce système nous avons par la suite intégré différentes méthodes pour contrôler des paramètres spectraux du signal. Dans le cas de Mono-Replay, ces paramètres sont accessibles via le vocoder de phase Super-VP qui sert de moteur audio. Ainsi on peut contrôler :

- La transposition de pitch
- La transposition de l'enveloppe spectrale
- Le timbre.

Lors de performances artistiques basées sur le déroulement temporel d'un enregistrement il arrive souvent que les musiciens contrôlent ce type de paramètre. Dans le cas d'une performance de DJ, par exemple, le musicien applique des filtres passe bas, passe haut ou passe bande pour créer des effets de transition dans le morceau. De la même manière que les contrôles temporels, les contrôles spectraux peuvent être continus ou discrets. Nous en présentons quelques-uns dans la suite de ce paragraphe.

### 2.4.1 Contrôle continu linéaire

Ce mode de contrôle définit à chaque instant  $t$  une nouvelle valeur d'un ou plusieurs paramètres de transformation spectrale. Ces paramètres évoluent selon une fonction affine. Mathématiquement,  $\forall t \in \mathbb{N}$ ,

$$\phi(t) = \Delta.\alpha(t) + S_{min} \quad (2.15)$$

Avec  $\Delta = S_{max} - S_{min}$ ,  $S_{max}$  la valeur maximale du paramètre de transformation spectrale,  $S_{min}$  la valeur minimale et  $\alpha(t)$  la valeur de contrôle de l'interface gestuelle, comprise dans l'intervalle  $[0;1]$ .

Par exemple, si on souhaite contrôler la transposition de hauteur du vocodeur de phase SuperVP avec un potentiomètre de telle sorte que l'on puisse transposer jusqu'à une octave au-dessous et une octave au-dessus de la valeur initiale, on choisit  $S_{max} = 1200$  cent et  $S_{min} = -1200$  cent.  $\alpha(t)$  représente les valeurs émises par le potentiomètre et comprises dans l'intervalle  $[0;1]$ . Ainsi la position minimale du potentiomètre règle la valeur de transposition de pitch sur -1200 cent et la valeur maximale sur 1200 cent.

### 2.4.2 Contrôle continu linéaire par morceaux

Sur des interfaces gestuelles comme les bras du méta-instrument 4 décrit section 4.2, il est difficile de régler le potentiomètre sur une valeur précise. Dans la vidéo dont le lien est en bas de page, on rejoue un enregistrement du trio n°2 pour corde et piano de Franz Schubert à l'aide du Méta-instrument 4 et de Mono-Replay. Le bras droit permet de définir la valeur de l'indice temporel en battant une pulsation selon le mode synchrone convexe décrit précédemment. Il permet aussi de contrôler la transposition de l'enveloppe spectrale du moteur audio SuperVP. Pour pouvoir facilement régler cette transposition sur zéro quand le bras est à mi-parcours, on utilise la fonction affine par morceaux représentées figure 2.18. Mathématiquement elle s'écrit de la sorte :

$\forall t \in \mathbb{N}$ ,

$$\phi(t) = \begin{cases} (S(t) - \Theta_{min}) \cdot \frac{S_{min} - \Theta}{S_{min} - \Theta_{min}} + \Theta & \text{si } S(t) \in (S_{min}, \Theta_{min}) \\ \Theta & \text{si } S(t) \in (\Theta_{min}, \Theta_{max}) \\ (S(t) - \Theta_{max}) \cdot \frac{S_{max} - \Theta}{S_{max} - \Theta_{max}} + \Theta & \text{si } S(t) \in (\Theta_{max}, S_{max}) \end{cases} \quad (2.16)$$

avec  $S_{min}$  et  $S_{max}$  les valeurs minimum et maximum que peut prendre le paramètre spectral que l'on souhaite contrôler ;  $\Theta_{min}$  et  $\Theta_{max}$  les valeurs seuils qui encadrent la valeur  $\Theta$  que l'on souhaite "viser" ;  $S(t)$  la fonction affine décrite dans le mode précédent telle que :  $S(t) = \Delta \cdot \alpha(t) + S_{min}$ , avec  $\Delta = S_{max} - S_{min}$  et  $\alpha(t)$  la fonction exprimant les données gestuelles sur l'intervalle  $[0,1]$ .

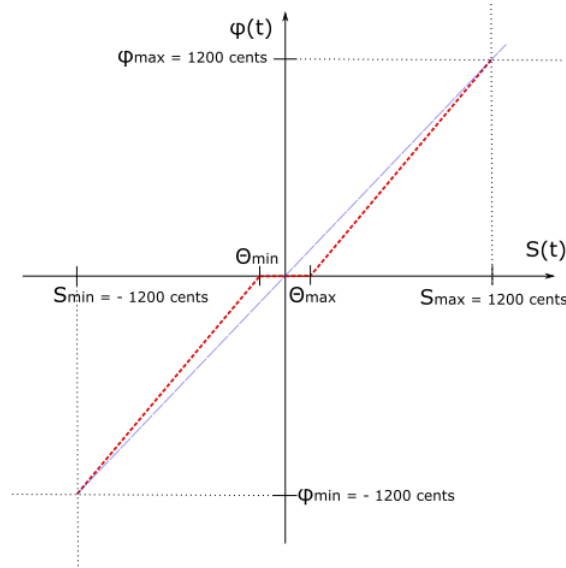


FIGURE 2.18 – Contrôle de la transposition de hauteur  $\phi(t)$  (en rouge) selon la fonction affine par morceaux décrite dans l'équation 2.16.  $S(t)$  exprime les valeurs de transpositions spectrales définies par l'utilisateur à l'aide d'un potentiomètre. Ces valeurs vont de  $S_{min} = -1200$  cents à  $S_{max} = 1200$  cents.  $\Theta$  représente la valeur cible autour de laquelle on souhaite définir un plateau.  $S(t)$  vaut zéro lorsqu'il est compris entre  $\Theta_{min} = -100$  cents et  $\Theta_{max} = 100$  cents. En bleu la droite  $\phi(t) = S(t)$

Pour pouvoir facilement régler cette transposition sur zéro quand le bras est à mi-parcours, on utilise une fonction affine par morceaux. La figure 2.18 illustre l'exemple cité en introduction de cette sous-section.

### 2.4.3 Contrôle spectral discret

Certains modes de contrôle temporel que nous étudions ici se basent sur une représentation indexée d'un enregistrement audio. Cette indexation est définie par des marqueurs  $\lambda_i$  qui indiquent le début de chaque segment. Les méthodes de contrôle spectral discrètes tiennent compte de cette indexation. Elles définissent pour chaque segment une nouvelle enveloppe d'une durée  $T$  (durée du segment) qui contrôlera des paramètres spectraux du signal. On dit que le contrôle est discret car son paramétrage est défini à chaque début de segment.

### 2.4.4 Contrôle discret par calcul d'enveloppe

Certains modes de contrôle temporel que nous étudions ici se basent sur une représentation indexée d'un enregistrement audio. Cette indexation est définie par des marqueurs  $\lambda_i$  qui indiquent le début de chaque segment. La méthode de contrôle spectral discret par calcul d'enveloppe tient compte de cette indexation. Elle définit pour chaque segment une nouvelle enveloppe d'une durée  $T$  (durée du segment) qui contrôlera des paramètres spectraux du signal. On dit que le contrôle est discret car son paramétrage est défini à chaque début de segment.

Pour un segment de durée  $T$  cette méthode détermine une nouvelle enveloppe qui paramètre les contrôles spectraux  $\sigma(t)$  d'un enregistrement.

Pour calculer cette enveloppe on utilise deux sous-enveloppes  $E_1$  et  $E_2$ , représentées figure 2.19, qui pilotent la modulation en amplitude et en fréquence d'un sinus.

Mathématiquement cela s'exprime de la manière suivante :  $\forall i \in \mathbb{N}, \forall t \in [0; T]$

$$\phi(t) = \Delta.E(t) + S_{min} \quad (2.17)$$

Avec  $\Delta = S_{max} - S_{min}$ ,  $S_{max}$  et  $S_{min}$  les valeurs maximale et minimale du paramètre de transformation spectrale.  $E(t)$  est une fonction sinusoïdale dont la fréquence et l'amplitude sont modulées selon l'équation suivante :

$\forall t \in [0; T] :$



$$E(t) = E_1(t) \sin\left(2\pi \sum_{i=0}^t E_2(t) \cdot \Delta t\right) \quad (2.18)$$

Avec  $E_1(t)$  et  $E_2(t)$  deux fonctions affines par morceaux prédéterminées en amont. Sur la figure 2.19 la fonction  $E_1(t)$  est représentée par la courbe du haut et la fonction  $E_2(t)$  par celle du milieu. La fonction  $E(t)$  résultante est représentée par la courbe du bas de cette même figure.

La durée de cette enveloppe va s'étirer temporellement entre le point d'entrée du segment lu et le point de sortie de celui-ci. Cela permet de reproduire à chaque nouveau segment la même forme d'enveloppe mais avec une durée différente. Si on prend le cas d'une indexation à la pulsation, par exemple, l'enveloppe fera varier les paramètres de transformation spectrale à chaque nouvelle pulsation.

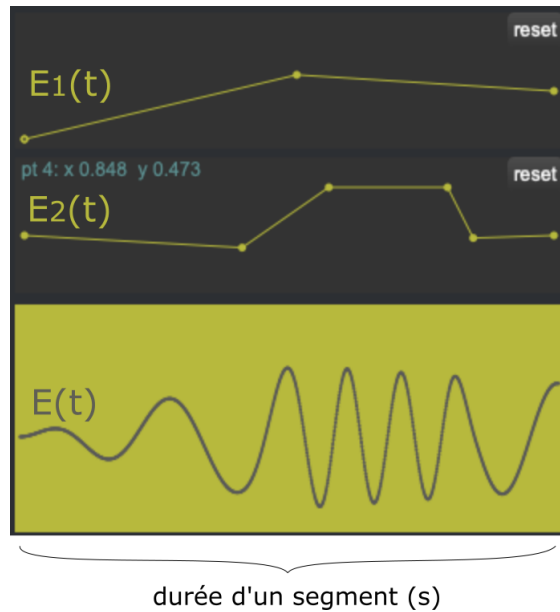


FIGURE 2.19 – Pour un segment audio d'une durée  $T$ , la fonction  $E_1(t)$  contrôle l'évolution de l'enveloppe d'amplitude de la fonction sinusoïdale  $E(t)$ , définie selon l'équation 2.18. La fonction  $E_2(t)$  contrôle l'évolution de la fréquence de la fonction sinusoïdale  $E(t)$ .

Ce système de création d'enveloppe a été créé comme un outil de paramétrage de l'effet vibrato. Effectivement, en paramétrant la fréquence et l'amplitude de la variation de pitch, on peut dessiner des formes d'enveloppe imitant le vibrato d'un instrumentiste comme représenté figure 2.20. On peut

choisir l'amplitude, la fréquence, la position de déclenchement et la durée de ce vibrato en paramétrant les enveloppes de modulation d'amplitude et de fréquence.

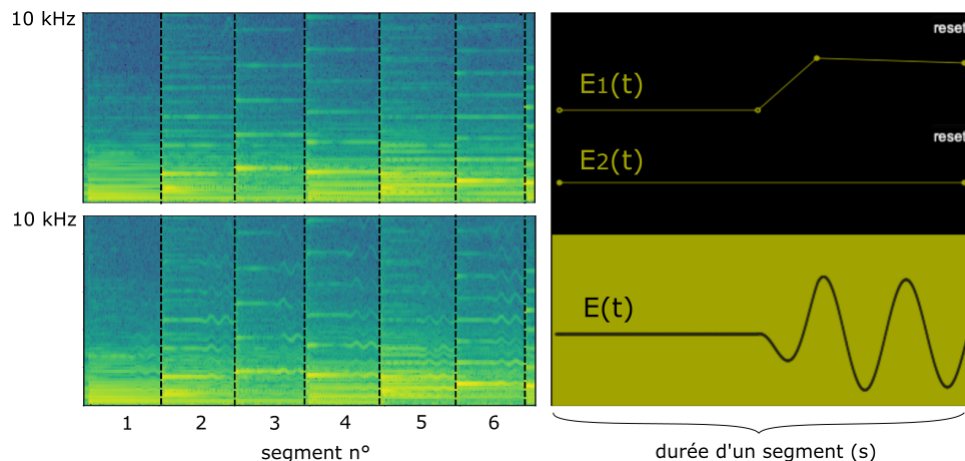


FIGURE 2.20 – Création d'une enveloppe  $E(t)$  à l'aide des fonctions  $E_1(t)$  et  $E_2(t)$ . La fonction  $E_1(t)$  contrôle l'évolution de l'enveloppe d'amplitude de la fonction sinusoïdale  $E(t)$ , définie selon l'équation 2.18. La fonction  $E_2(t)$  contrôle l'évolution de la fréquence de la fonction sinusoïdale  $E(t)$ . En bas à gauche, le spectrogramme du signal transformé sur lequel on applique l'enveloppe  $E(t)$  pour chaque segment. L'enregistrement est segmenté ici selon la position de ses transitoires d'attaque.

Le fait que l'enveloppe se déclenche en fonction de l'indexation du fichier audio permet d'obtenir des évolutions sonores tenant compte de l'information qui est indexée par cette indexation. Ce système n'est donc possible que dans le cadre d'une indexation préalable du fichier audio et pourrait s'appliquer à tous les systèmes audios interactifs permettant d'interagir avec des enregistrements. Bien que le modèle présenté ici n'affecte que les paramètres de transformation spectrale du vocodeur de phase SuperVp, on pourrait imaginer un tel système pour contrôler d'autres paramètres de synthèse d'un son enregistré, comme la vitesse de lecture par exemple.

## Chapitre 3

# Mono-Replay

### Sommaire

---

<b>3.1</b>	<b>Architecture générale . . . . .</b>	<b>85</b>
<b>3.2</b>	<b>Choix des langages de programmation . . . . .</b>	<b>88</b>
3.2.1	PUCE MUSE et Max . . . . .	88
3.2.2	Accéder aux algorithmes de MIR dans Max à l'aide de Node.js	89
<b>3.3</b>	<b>Système d'indexation assistée . . . . .</b>	<b>90</b>
3.3.1	Paramétrer les algorithmes de <i>essentia.js</i> dans Max/MSP . .	91
3.3.2	Trois niveaux d'indexation . . . . .	95
<b>3.4</b>	<b>Gestion des modes de contrôle temporel . . . . .</b>	<b>99</b>
3.4.1	Scrub, Speed et TapTempo . . . . .	101
3.4.2	Naviguer dans l'enregistrement selon trois niveaux d'indexation	109
<b>3.5</b>	<b>Gestion des transformations spectrales . . . . .</b>	<b>110</b>
3.5.1	Contrôles des paramètres SuperVP . . . . .	111
3.5.2	Éditeur d'enveloppe de transformation . . . . .	112
<b>3.6</b>	<b>Gestion des mémoires . . . . .</b>	<b>113</b>
3.6.1	Pattern storage . . . . .	113
3.6.2	Interpolation de mémoires . . . . .	113
<b>3.7</b>	<b>Création de projet . . . . .</b>	<b>115</b>
<b>3.8</b>	<b>Dictionnaire OSC et mapping . . . . .</b>	<b>117</b>

---

Pour démontrer la faisabilité des concepts présentés précédemment, nous avons développé un logiciel intégrant toutes les fonctionnalités nécessaires à la

conception d'interactions musicales basées sur le contrôle temporel d'un enregistrement. Dans ce chapitre nous détaillons les fonctionnalités de ce logiciel et comment elles ont été conçues pour valider les concepts abordés auparavant.

Pour l'association PUCE MUSE un des intérêts de ce travail de recherche est d'aboutir à la création d'un système multi-plateformes s'intégrant dans l'environnement de logiciels déjà existants. Cela permet à l'association de distribuer un nouveau logiciel<sup>1</sup> et d'ouvrir son activité [17] [19] [43] à de nouvelles utilisations informatiques. Ce développement a pris une part importante dans notre travail de recherche et a permis d'étendre notre démarche à la réalisation d'un cas d'usage concret.

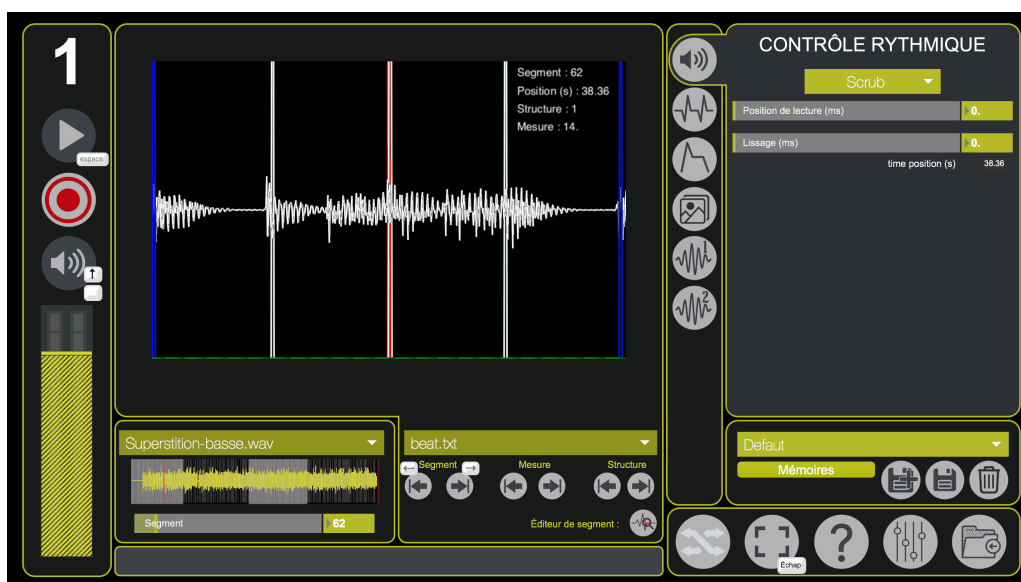


FIGURE 3.1 – Interface graphique du logiciel Mono-Replay

Ce logiciel s'appelle Mono-Replay. Il permet à un utilisateur de concevoir des interactions selon différents degrés de contrôle pour manipuler le déroulement rythmique d'un enregistrement. Le terme "Replay" fait référence à l'action de rejouer un enregistrement audio et le terme "Mono" fait référence à la suite de logiciel "Mono" dans laquelle ce logiciel figure. Afin d'avoir un aperçu général du logiciel, son interface graphique est présentée en figure 3.1. La charte graphique de cette interface est commune à tous les logiciels de la

1. Les logiciels développés par PUCE MUSE sont disponibles gratuitement sur le site : <https://www.meta-librairie.com/fr>

suite Mono, seul le contenu des onglets, l’affichage de la forme d’onde et les couleurs de l’interface diffèrent suivant les fonctionnalités de chaque Mono.

Mono-Replay intègre toutes les fonctions décrites dans le chapitre précédent ce qui en fait un outil complet pour créer des interactions basées sur les musiques enregistrées. De la phase d’indexation à la phase de contrôle gestuel, le logiciel possède des fonctionnalités qui permettent de paramétrer précisément l’interaction. Au croisement du sampler, de la platine de DJ et d’un outil de tap-tempo, Mono-Replay intègre en tout 11 modes de contrôle temporel qui peuvent être contrôlés en OSC par toutes les interfaces pouvant communiquer avec les systèmes d’exploitations pour ordinateurs développés par Apple et Microsoft.

Nous verrons ici les différentes fonctionnalités de ce logiciel et comment leur conception apporte des solutions aux problématiques abordées auparavant. Cette section s’inscrit dans la continuité de notre réflexion en illustrant chacune des problématiques soulevées dans le chapitre précédent par une solution pratique intégrée dans le logiciel.

### 3.1 Architecture générale

Mono-Replay suit l’architecture globale des applications audio interactives décrites dans la section 2.1. Il s’organise en trois blocs algorithmiques principaux, représentés schématiquement figure 3.2, qui permettent de réaliser ces fonctions :

- *Indexation* : cette fonction permet d’indexer un enregistrement pour indiquer au système la position d’évènements saillants. Ces évènements peuvent être par exemple le début d’une note ou la position des pulsations.
- *Mapping/Captation gestuelle* : cette fonction récupère les données issues d’une interface gestuelle et les retranscrit en paramètres de contrôle temporel et spectral. Elle permet par exemple de retranscrire le mouvement d’un potentiomètre en valeur de transposition de Pitch.

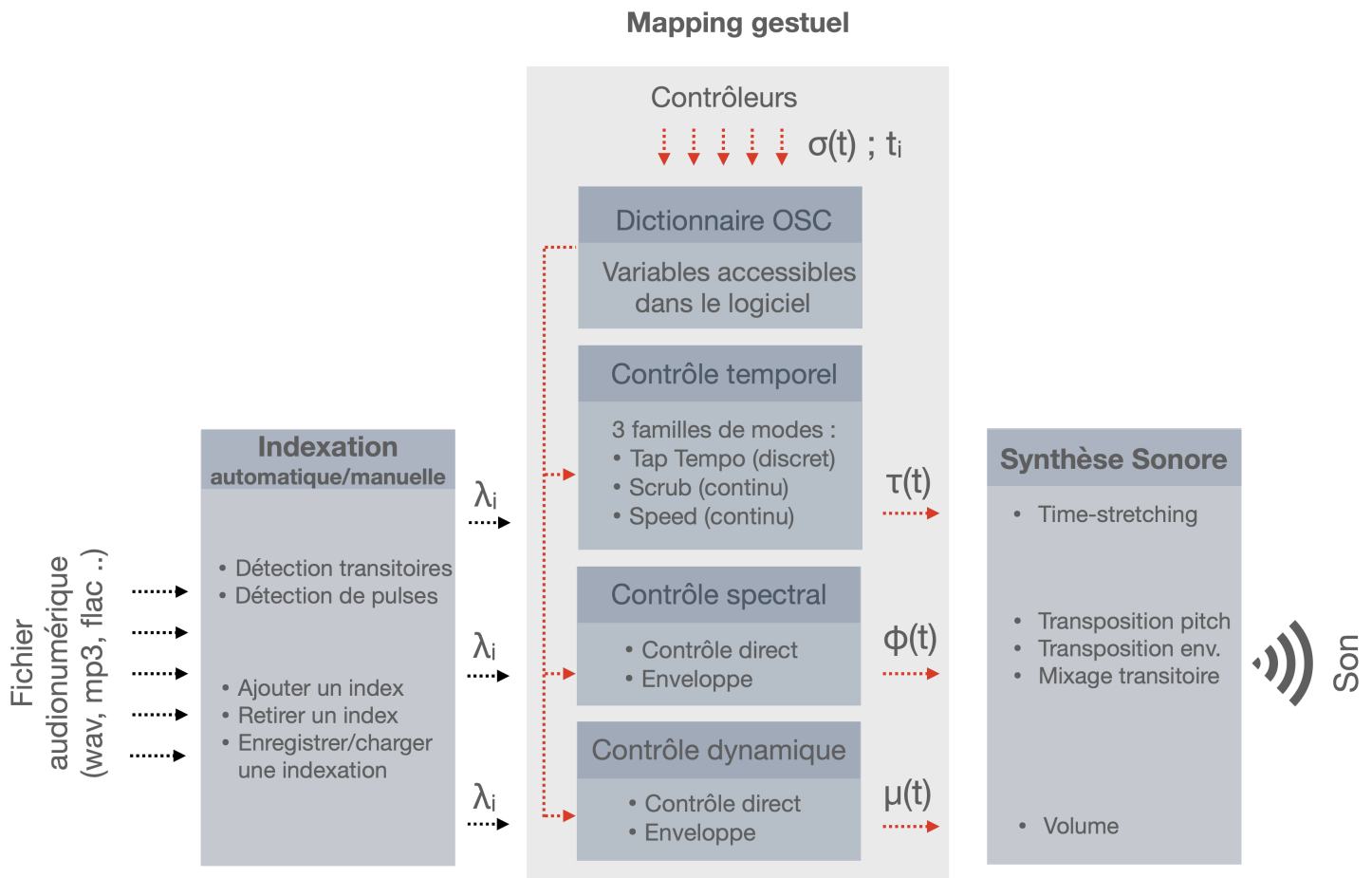


FIGURE 3.2 – Schéma fonctionnel du logiciel Mono-Replay. La fonction d’indexation sert à positionner des ancrs temporelles sur un enregistrement audio numérique, la fonction de mapping gestuel sert à relier les données d’un contrôleur aux paramètres de contrôle du logiciel, la fonction synthèse sonore sert à produire du son compte tenu du geste effectué par l’utilisateur.

- *Synthèse/analyse sonore* : cette fonction permet de générer en temps réel un rendu sonore qui correspond à la performance de l’utilisateur. Par exemple si l’utilisateur tape un tempo moins rapide une transformation sonore est effectuée pour que le fichier audio soit lu moins vite.

La figure 3.2 illustre cette architecture. On y aperçoit les entrées et les sorties de chaque fonction qui compose Mono-Replay ainsi que les liaisons synergiques qui les lient les unes aux autres. Pour bien comprendre les enjeux de cette synergie nous décrivons les entrées et les sorties de chacune de ces

fonctions ci-dessous :

*La fonction d'indexation* prend en entrée un enregistrement audionumérique ainsi que les commandes d'édition de l'utilisateur, afin de créer/corriger la collection d'ancres temporelles qu'il souhaite utiliser pour contrôler le déroulement rythmique de l'enregistrement. Cette fonction intègre toutes les dispositions décrites dans la section 2.2 et dispose ainsi d'un outil d'indexation manuelle et d'un outil d'indexation automatique. Elle produit en sortie un fichier texte stocké localement sur la machine dans lequel sont indiquées les positions temporelles de chaque ancre. Chaque ligne de ce fichier suit la syntaxe suivante : " $n;t$ " avec  $n$  le numéro de l'ancre et  $t$  sa position temporelle en *ms*. Une fois qu'un fichier de ce type est créé pour un enregistrement, il est automatiquement proposé à l'utilisateur si celui-ci sélectionne cet enregistrement.

*La fonction de captation gestuelle* prend en entrée le fichier d'indexation préalablement déterminé par la fonction d'indexation ainsi que les données captées par le geste d'un utilisateur. Comme nous l'avons vu dans la section 2.3.2 les données captées peuvent être continues ou discrètes en fonction de l'interface gestuelle utilisée. Grâce à cette fonction les données gestuelles issues de l'interface sont transcrites en paramètres de contrôle pour les différents modes de contrôle temporel. Certains modes de contrôle temporel par "battue de tempo" vont en plus analyser le comportement de l'utilisateur pendant sa performance pour modifier un autre paramètre de contrôle comme abordé dans la section 2.3.3.1. Typiquement, il s'agira de suivre l'espacement entre deux points de contrôle générés par le geste de l'utilisateur, qui appuie sur une touche du clavier, pour mesurer l'évolution de la durée d'une pulsation. En sortie de cette fonction on récupère à chaque instant  $t$  une valeur d'indice temporel  $\tau(t)$  que l'on définit dans la section 2.3.1.1. Pour rappel, cet indice temporel correspond à la position à synthétiser dans l'enregistrement compte tenu des transformations temporelles en cours.

*La fonction de synthèse/analyse* récupère en entrée l'indice temporel généré par la fonction de captation gestuelle et les collections d'ancres temporelles générées par la fonction d'indexation. L'indice temporel indique au système la position à synthétiser dans l'enregistrement. Le vocodeur de phase SuperVP

[50] permet d'effectuer les transformations sonores nécessaires pour synthétiser l'enregistrement compte tenu de l'évolution de cet indice temporel. Si il varie lentement le vocodeur de phase permettra de réduire la vitesse de synthèse du fichier audio sans avoir de transposition de hauteur ou de phénomène de discontinuité. Si l'indice temporel ne varie plus dans le temps, le vocodeur de phase synthétisera un son correspondant à la fenêtre d'analyse pointée par l'indice temporel. Le son sera comme "gelé".

## **3.2 Choix des langages de programmation**

Le choix des langages informatiques pour programmer des transformations audio en temps réel est relativement restreint. Dans la sphère des développeurs en informatique musicale, il existe une véritable dynamique de recherche pour trouver des langages qui permettraient plus de possibilités quant à la portée des applications développées. Nous détaillons ici les motivations pour utiliser les langages informatiques utilisés dans le cadre du développement de Mono-Replay.

### **3.2.1 PUCE MUSE et Max**

Max/MSP est un langage très utilisé pour la création musicale assistée par ordinateur. Son aspect graphique le rend intuitif dans son utilisation et la quantité d'objets disponibles en fait un outil complet pour développer des traitements sonores pointus ou des prototypes de logiciels. Il est néanmoins très limité pour faciliter la distribution sur plusieurs plateformes compatibles. Des langages comme FAUST sont plus appropriés pour convertir un script de transformation audio temps réel en plusieurs formats informatiques compatibles avec les plateformes utilisées pour la création musicale assisté par ordinateur. Ainsi un langage comme FAUST permet en un clic de convertir son script de traitement du son en d'autres langages comme un objet max, un script csound, un vst, un vsti ou un objet pure data tous compatibles avec une multitude de plateformes telles que iOS, macOS, Windows, Linux, Android... Le choix de Max pour développer les logiciels au sein de la structure PUCE MUSE repose donc plutôt sur la facilité d'utilisation de ce langage qui per-



met à la fois à des utilisateurs débutants de concevoir des prototypes qu'à des utilisateurs experts de concevoir des logiciels complets. Les patchs Max/MSP développés au sein de PUCE MUSE sont distribués sous forme d'applications Mac stand alone et d'exécutables Windows.

Il est également à noter que Max/MSP n'a pas été conçu pour le versionning. Un effort considérable a été porté par PUCE MUSE pour rendre cette étape de développement informatique le plus ergonomique possible dans Max/MSP. Un système de sous-modules stockés dans un répertoire Git permet de faire des mises à jour communes sur toutes les applications Mono. Ces sous-modules concernent l'affichage de l'interface utilisateur, la gestion des mémoires, la captation audio et vidéo et la configuration des paramètres systèmes.

### **3.2.2 Accéder aux algorithmes de MIR dans Max à l'aide de Node.js**

Il existe plusieurs librairies Max/MSP qui permettent de suivre l'évolution de paramètres audio directement depuis le signal. Néanmoins la plupart sont conçus pour un suivi en temps réel, soit en même temps que le rendu audio. Dans notre cas il est nécessaire que l'analyse se fasse de façon instantanée en amont de la performance. Compte tenu de cette nécessité, la récupération d'informations sur le contenu audio d'un enregistrement dans Max/MSP est beaucoup moins courante.

L'environnement Node.js a permis de grandement faciliter la gestion de scripts Javascript dans les logiciels informatiques. Ainsi de nombreux concepteurs d'algorithmes ont mis à disposition leur travail dans cet environnement pour bénéficier de ces facilités. Les dernières mises à jour de Max/MSP permettent d'accéder aux librairies JavaScript *via* cet environnement. Un objet disponible dans Max/MSP permet de configurer l'exécution d'un script faisant appel à des librairies disponibles dans Node.js. C'est notamment le cas de la librairie C++ Essentia qui nous donne accès à de nombreux algorithmes de récupération d'informations depuis le contenu audio.

Essentia [7] est un projet de recherche interuniversitaire qui met à disposition depuis 2013 plus d'une centaine de méthodes de traitement du signal pour aider à la conception de scripts visant à récupérer des informations sur le

contenu audio d'un enregistrement. Initialement conçue en C++, cette librairie a été portée en JavaScript très récemment et s'intègre complètement dans l'environnement Node.js qui est dorénavant accessible depuis Max/MSP. L'accès aux algorithmes de cette librairie directement depuis notre logiciel décuple le nombre de possibilités pour segmenter un enregistrement selon l'évolution d'un paramètre audio. Tout comme les modes de contrôle temporel basés sur la détection de transitoires ou la détection de pulsations que l'on présente dans la suite de notre travail on pourrait ainsi imaginer d'autres modes basés sur les autres algorithmes de la librairie Essentia comme par exemple la détection de pitch, l'évolution du centroïde spectral... De plus la librairie Essentia est très souvent mise à jour avec les derniers algorithmes de traitement du signal. Rendre compatible notre logiciel avec cette librairie, nous permet d'assurer un peu plus la pérennité de notre système qui bénéficiera aussi de ces mises à jour. L'effort fait pour rendre compatible cette librairie avec notre logiciel étant fait, il devient aussi facile d'ajouter d'autres algorithmes de cette librairie à notre éditeur de segment.

### 3.3 Système d'indexation assistée

Comme nous l'avons vu dans la section 2.2, dans tout système audio interactif permettant de manipuler un enregistrement, il convient d'intégrer un système d'indexation manuelle. Ce dernier peut-être assisté par des méthodes de détection automatiques. Effectivement la détection automatique permet de gagner du temps dans la création du fichier d'indexation et l'édition manuelle laisse la possibilité à l'utilisateur de corriger les résultats si des ancres déterminées par détection automatique ne lui conviennent pas. Il arrive aussi souvent de vouloir indexer des phénomènes qui ne sont pas détectables par des algorithmes informatiques comme par exemple les syllabes d'une voix enregistrée. Dans ce cas l'accès à des fonctionnalités d'indexation manuelle est essentiel. Le logiciel Mono-Replay intègre un tel système. L'éditeur de segment représenté figure 3.3 permet d'utiliser les algorithmes *SuperFlux Extractor* [6], *MultiFeature Beat Tracking* [59][58] et *Degara* [20] tous issus de la librairie Essentia [7] pour déterminer automatiquement la position des transitoires d'attaques, la

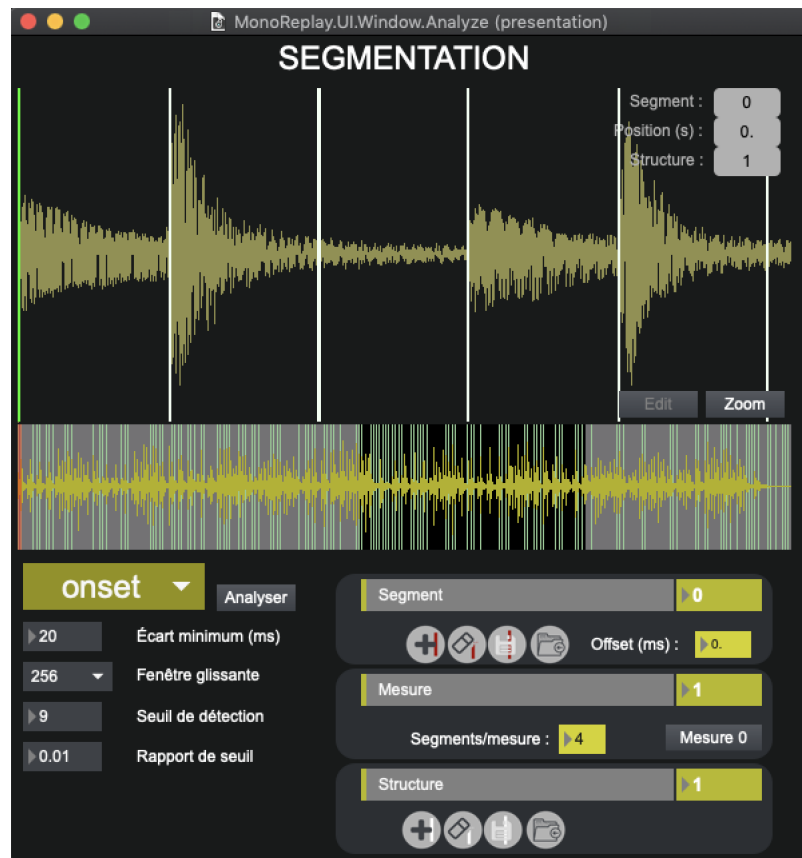


FIGURE 3.3 – Fenêtre "Éditeur de segment" du logiciel Mono-Replay

valeur de BPM et la position des pulsations du morceau.

### 3.3.1 Paramétrer les algorithmes de *essentia.js* dans Max/MSP

Mono-Replay permet de paramétrer et de lancer des fonctions de la librairie *essentia.js* directement depuis l'interface utilisateur. La fenêtre éditrice de segment présentée en figure 3.3, donne accès aux paramètres de la fonction sélectionnée.

#### 3.3.1.1 SuperFluxExtractor

La fonction *SuperFluxextractor* [6] est une méthode de détection de transitoire basée sur l'analyse de la courbe d'évolution du flux spectral. Le flux spectral est la somme des différences pour chaque coefficient d'une transformée

de Fourier d'une fenêtre à l'autre. Le calcul du flux spectral  $SF$  est effectué selon l'équation suivante  $SF$  :

$$SF(n) = \sum_{k=-N/2}^{N/2-1} H(|X(n, k)| - |X(n-1, k)|) \quad boeck \quad (3.1)$$

où  $k$  est la fréquence,  $n$  la trame pour laquelle on calcule le flux spectral et  $N$  la taille de la fenêtre d'analyse de la transformée de Fourier. La méthode intègre en plus une fonction de suivi de phases qui permet d'évaluer si les brusques modulations en amplitude et en fréquence sont des vibratos ou simplement la présence d'un nouveau transitoire. Effectivement, le début de ces deux phénomènes se caractérise par le même comportement en fréquence et en amplitude. L'accès au paramétrage de la fonction de détection permet d'adapter une utilisation pour un certain type d'enregistrement. Ces paramètres présentés dans la figure 3.4 sont les suivants :



FIGURE 3.4 – Onglet de paramétrage de la méthode de détection de transitoire *Super-Fluxextractor* dans le logiciel Mono-Replay.

- *Écart minimum (ms)* : Ce paramètre permet de définir l'écart minimum entre deux transitoires consécutifs. Dans le cas d'un enregistrement polyphonique, il est fréquent de détecter un léger décalage entre le déclenchement des différentes notes qui constituent un accord. Ces petits décalages sont détectés par notre algorithme comme le début d'une nouvelle note. Sur le principe c'est bien le cas, mais dans la pratique on souhaite souvent placer une seule ancre temporelle au début de l'accord et pas plusieurs ancres pour chaque note qui constitue cet accord. Le

réglage de ce paramètre permet d'éviter la détection de transitoires trop rapprochés et donc de pallier ce problème. Ce paramètre est aussi très utile pour détecter des transitoires selon une division rythmique. Par exemple, si on connaît la valeur de tempo du morceau qu'on souhaite indexer, on peut régler l'écart minimum entre deux transitoires sur une valeur de durée proche de celle qui sépare deux pulsations à la valeur de tempo connue, et ainsi obtenir une indexation selon les transitoires du morceau compte tenu de cette division rythmique.

- *Taille de fenêtre et taille de fenêtre glissante* : Ces paramètres permettent de régler la taille de la fenêtre d'analyse ainsi que la taille de la fenêtre glissante pour la transformée de Fourier que l'on calcule pour déterminer le flux spectral. Plus la fenêtre d'analyse et la fenêtre glissante sont petites, plus on a de points sur la courbe du flux spectral. On pourra alors mieux détecter les très courtes variations de cette courbe. Néanmoins, cela entraînera une résolution spectrale plus faible et donc une diminution de la dynamique de la courbe. Pour bien régler ce paramètre il faut trouver le juste milieu entre une bonne résolution temporelle (nombre de points sur la courbe d'évolution du flux spectral) et une résolution fréquentielle suffisante (nombre de coefficients spectraux pour notre transformée de Fourier) pour détecter suffisamment de transitoires.
- *Rapport de seuil* : Ce paramètre permet de régler le seuil de détection sur la courbe représentative du flux spectral, dite aussi courbe de nouveauté, à partir duquel on considère qu'un dépassement du seuil correspond à un nouveau transitoire. La valeur de ce seuil correspond au rapport suivant :  

$$\text{seuil} = (\text{courbe de nouveauté}) / (\text{moyenne de la courbe de nouveauté})$$
ce qui permet d'être plus sensible aux transitoires de faible énergie.
- *Seuil de détection* : Ce paramètre permet de régler le seuil de détection sur la courbe représentative du flux spectral (courbe de nouveauté) à partir duquel on considère qu'un dépassement du seuil correspond à

un nouveau transitoire. La valeur de ce seuil correspond à la différence suivante :

$seuil = \text{courbe de nouveauté} - \text{signal moyen}$

ce qui permet de tenir compte du bruit ambiant dans notre détection de transitoire.

Une fois la détection de transitoire effectuée, les positions temporelles de ceux-ci sont stockées en mémoire. Ces ancrs temporelles apparaissent dans l'interface graphique de l'application sous forme de barres verticales positionnées sur la forme d'onde de l'enregistrement sélectionnée comme présentée dans la figure 3.1. Une fois les corrections faites, on peut enregistrer les repères temporels dans un fichier texte en local sur la machine de l'utilisateur. Ainsi ce dernier peut charger à tout moment ce fichier d'indexation.

### 3.3.1.2 RhythmExtractor2013

Cette fonction permet de détecter la position des pulsations d'un enregistrement à l'aide de l'algorithme de détection de pulsation *multifeature beat-tracking* [59] ou de l'algorithme de détection de pulsation *degara* [20]. Les paramètres qui permettent de la régler, présentés dans la figure 3.5 sont les suivants :

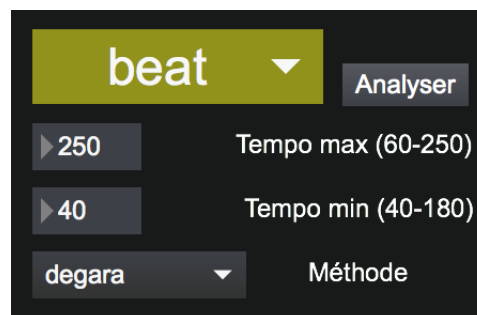


FIGURE 3.5 – Paramétrage de la méthode de détection de pulsation *multifeature beat-tracking*[59] et *degara*[20] dans le logiciel Mono-Replay

- *Tempo max* : Ce paramètre permet de régler la valeur maximale de tempo en battement par minute (BPM) détectable par l'algorithme. La version de l'algorithme disponible dans la librairie *essentia.js* permet de définir cette valeur de BPM par un entier compris entre 60 et 250.

Le réglage de ce paramètre est très utile dans le cas où l'algorithme détecte une pulsation trop grande par rapport à la valeur réelle. Basé sur la détection de périodicité dans un enregistrement, cet algorithme détermine une valeur de tempo en déterminant l'écart temporel le plus représenté entre les pics d'une fonction d'autocorrelation. Il arrive parfois que l'algorithme détecte un tempo qui soit multiple de la valeur réelle. Dans quel cas le paramètre Tempo max permet de corriger cela.

- *Tempo min* : Ce paramètre permet de régler la valeur minimale de tempo détectable dans l'enregistrement. La version de l'algorithme disponible dans la librairie *essentia.js* permet de définir cette valeur de BPM par un entier compris entre 40 et 180. Le réglage de ce paramètre est très utile dans le cas où l'algorithme détecte une pulsation trop petite par rapport à la valeur réelle.
- *Méthode* : Ce paramètre permet de choisir parmi les deux méthodes de détection de pulsation. La méthode *multifeature beat-tracking* détermine 8 fonctions de détection de transitoire, selon des méthodes spectrales ou de détermination d'amplitude puis applique la méthode *degara* pour estimer la position des pulsations pour chacune d'entre elle et ainsi obtenir 8 propositions de détection de pulsation différentes. Un coefficient de similarité est calculé pour chacune de ces propositions et la proposition qui a le coefficient le plus élevé est sélectionnée. Il est à noter que pour la méthode *degara* on estime la position de pulsation seulement sur une fonction de détection de transitoire.

### 3.3.2 Trois niveaux d'indexation

Lorsque l'on rejoue collectivement une musique non pulsée sans chef d'orchestre, il est très difficile pour les interprètes de se synchroniser les uns aux autres. Des logiciels de visualisation de partition comme *John* [28] sont des solutions efficaces pour informer les joueurs de la progression temporelle de chaque pupitre dans une partition. L'*Orchestre National Électroacoustique (ONE)* utilise ce système pour définir temporellement les différentes phases

d'improvisation pour chaque instrumentiste lors de sa performance.

L'expérience de PUCE MUSE au sein de cet orchestre nous a poussés à réfléchir à un système qui permettrait à plusieurs utilisateurs de Mono-Replay de rester synchronisés les uns autres lors du déroulement d'un même enregistrement. Pour cela, le système d'indexation intégré au logiciel a été conçu pour permettre à l'utilisateur de se positionner sur le début d'un segment selon trois niveaux hiérarchiques différents : le segment ; la mesure ou la structure. L'idée est de pouvoir se synchroniser par rapport aux autres utilisateurs en se référant au même découpage selon ces trois niveaux. Ainsi, si un joueur se décale par rapport à un autre, il peut se rendre au segment, à la mesure ou à la structure suivante pour attendre que l'autre joueur s'y rende aussi. L'utilisateur peut s'aider d'un retour visuel dans l'interface graphique du logiciel pour se positionner dans l'enregistrement compte tenu de cette hiérarchie comme suggéré sur la figure 3.6) :

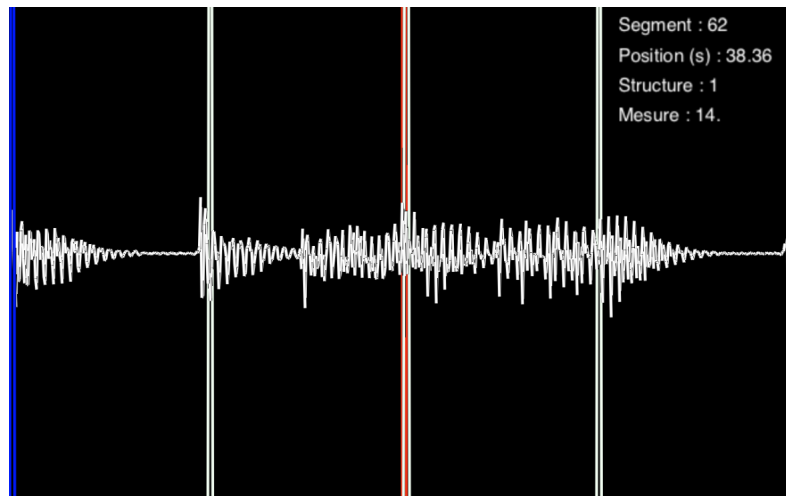


FIGURE 3.6 – Affichage de la forme d'onde dans Mono-Replay avec retour visuel de la position de lecture selon les trois niveaux d'indexation (segment, mesure, structure)

- *Le segment* : Il s'agit du plus petit découpage possible de l'enregistrement. Chaque mode de contrôle temporel intègre ce découpage pour calculer l'évolution de l'indice temporel  $\tau(t)$ . Pour obtenir ce découpage, l'application propose des méthodes de traitement du signal basées sur la détection de pulsation et de transitoire que nous avons détaillés plus haut. On peut aussi obtenir cette indexation grâce à des fonctions ba-



siques d'édition que nous détaillerons dans la suite de cette sous-section.

- *La mesure* : Il s'agit d'un découpage intermédiaire de l'enregistrement. Pour obtenir cette indexation, l'utilisateur renseigne un nombre de segments qui constituera une mesure. Par exemple, si l'utilisateur indique 4, l'application créera un découpage à la mesure tous les 4 segments.
- *La structure* : Il s'agit d'un découpage plus global de l'enregistrement. L'utilisateur renseigne les grandes structures musicales de l'enregistrement pour pouvoir s'y rendre à n'importe quel moment de la performance. Ce découpage peut indiquer par exemple le début d'un refrain, d'un couplet ou d'un solo.

En plus de proposer une solution au problème de synchronisation lorsque plusieurs utilisateurs de Mono-Replay jouent ensemble, le découpage en trois niveaux d'indexation permet aussi d'être plus rapide pour se déplacer dans l'enregistrement. En une commande, on peut se placer au début du refrain du morceau, revenir au segment précédent ou avancer mesure par mesure. La création de ces trois niveaux de découpage se fait toujours relativement au découpage de premier degré dit "au segment". Pour ajouter et corriger manuellement de nouveaux segments, il faut utiliser les commandes disponibles dans la partie droite de l'éditeur de segments. Elles sont organisées de la façon suivante :



FIGURE 3.7 – Bouton ajouter dans Mono-Replay

Comme présenté figure 3.7, le *bouton ajouter* permet d'ajouter une ancre temporelle pour marquer le début d'un segment. Ce bouton est disponible pour les trois niveaux d'indexations et permet d'ajouter une nouvelle mesure ou une nouvelle structure à la position souhaitée.

Comme présenté figure 3.8 le *bouton supprimer* permet d'effacer une ancre temporelle qui marque le début d'un segment. Ce bouton est aussi disponible



FIGURE 3.8 – Bouton supprimer dans Mono-Replay

pour les trois niveaux d'indexation du logiciel.



FIGURE 3.9 – Bouton enregistrer dans Mono-Replay

Comme présenté figure 3.9, le *bouton enregistrer* permet de stocker localement au format txt le découpage temporel en cours d'édition. Une fenêtre de dialogue s'ouvre et permet de spécifier le nom du fichier. Le fichier est automatiquement enregistré dans le répertoire qui s'appelle /index situé à la racine du projet comme présenté figure 3.18.

Dans la partie droite de l'éditeur de segment, comme indiqué sur la figure 3.3), nous pouvons aussi définir une durée pour reculer ou avancer la position de toutes les ancrs temporelles en même temps. Ce contrôle est très utile dans le cas d'une indexation assistée par détection de pulsation par exemple. En effet, il arrive souvent qu'une pulsation soit détectée au milieu d'un transitoire. Dans ce cas il est important de pouvoir appliquer facilement une correction de ce type sur l'ensemble des pulsations détectées pour que le début du segment soit positionné sur le début du transitoire.

C'est grâce à toutes ces fonctionnalités que l'éditeur de segment de Mono-Replay devient un outil complet pour l'indexation d'un enregistrement dont nous contrôlons le déroulement temporel. Il intègre le dispositif nécessaire pour intégrer des algorithmes de traitement du signal tout en gardant les fonctionnalités d'édition manuelle pour corriger ou ajouter des segments supplémentaires. La possibilité d'intégrer les algorithmes de traitement du signal de la librairie Essentia [7] pour assister l'indexation ouvre de très larges possibilités pour des améliorations futures du logiciel. Pour le moment seulement trois

algorithmes de cette librairie sont adaptés pour être directement intégrés aux modes de contrôle temporel de Mono-Replay. Il serait intéressant d'en ajouter d'autres pour imaginer de nouveaux modes de contrôle temporel. On pourrait imaginer des modes de contrôle temporel basés sur l'indexation d'éléments spectraux comme la fréquence fondamentale. L'ajout d'un tel mode nécessiterait un travail important pour intégrer les résultats de l'algorithme d'analyse dans les modes déjà existants. Néanmoins l'ajout de nouveaux algorithmes de la librairie Essentia dans Mono-Replay est une tâche relativement simple dans la mesure où l'architecture nécessaire pour accéder à cette librairie Javascript *via* Mono-Replay est déjà en place.

### 3.4 Gestion des modes de contrôle temporel

La motivation première de PUCE MUSE au travers de ce projet de recherche est la réalisation d'un système fonctionnel proposant différents scénarios d'interactions gestuelles permettant de contrôler de façon expressive le déroulement d'un enregistrement. Dans le cadre de la réalisation de ce système, des échanges hebdomadaires ont eu lieu avec le musicien et concepteur d'instrument numérique Serge de Laubier pour discuter de l'avancement des développements. Ces échanges ont permis de corriger et d'ajouter des fonctionnalités supplémentaires au système pour qu'il corresponde au mieux aux attentes de PUCE MUSE. L'expérience de Serge de Laubier dans la pratique de l'informatique musicale sur scène, en studio et dans le cadre de nombreux ateliers pédagogiques ont fait de ces échanges un véritable atout pour la conception de ce système.

Le terme contrôle temporel regroupe ici les modes de jeux de Mono-Replay qui sont axés sur le contrôle du déroulement temporel d'un enregistrement. Ces derniers ont été imaginés dans le cadre des échanges décrits ci-dessus. Leur implémentation s'est faite progressivement en même temps que leur phase de conception.

Cette méthode de travail nous a permis de traiter au cas par cas chaque mode de contrôle temporel et d'en créer des nouveaux en fonction de ce que

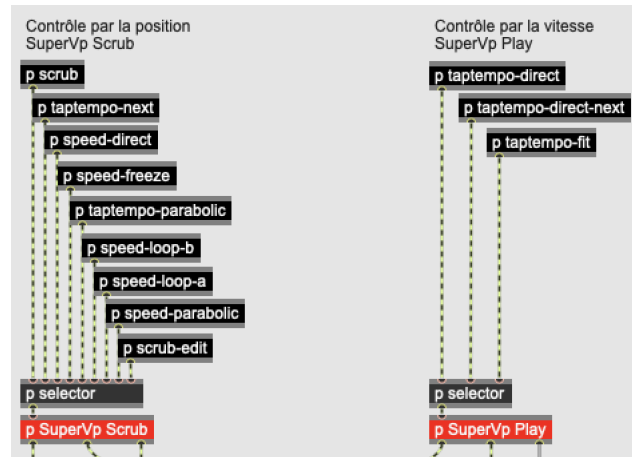


FIGURE 3.10 – Organisation des modes de contrôle temporel dans le code de Mono-Replay

les précédents ne permettaient pas de faire. Au fur et à mesure de leur implémentation, l'ajout des nouveaux modes venaient parfois compromettre le fonctionnement des anciens. Il devenait difficile de garder une logique dans le fonctionnement global du logiciel dans la mesure où l'organisation des fonctions de contrôle temporel n'était pas unifiée. C'est pour résoudre ce problème que nous avons pensé le formalisme décrit dans la section 2.3. Tous les modes de contrôle temporel présents dans Mono-Replay suivent donc ce formalisme. Ils prennent des données gestuelles discrètes ou continues en entrée ainsi qu'un fichier d'indexation et donnent en sortie une valeur d'indice temporel correspondant à la position synthétisée dans l'enregistrement. Cet indice temporel peut aussi être converti en vitesse de lecture pour certains modes de contrôle temporel discrets.

Mono-Replay compte pour le moment 11 modes de contrôle temporel que l'on peut sélectionner grâce à l'onglet présenté figure 3.10. Ils suivent tous le formalisme décrit ci-dessus. On peut en ajouter d'autres tant que l'on s'assure qu'ils prennent en entrée un fichier d'indexation ainsi que des contrôles gestuels et qu'ils génèrent bien en sortie un indice temporel. Intégrer ce formalisme dans Mono-Replay permet d'assurer la continuité du logiciel en garantissant la facilité d'intégration de prochains modes. Le mode de contrôle temporel intégré dans T-Voks [16] [34] [57] suit ce formalisme. Il a été ajouté très facilement dans Mono-Replay car il a déjà été implémenté dans Max/MSP. Dans Mono-

Replay ce mode s'appelle "parabolic".

### 3.4.1 Scrub, Speed et TapTempo

Les 11 modes de contrôle temporel de Mono-Replay sont accessibles par l'interface utilisateur en utilisant un menu déroulant que l'on peut voir figure 3.11. Ce dernier sert à filtrer les propositions de modes selon qu'ils permettent un contrôle de la position de lecture (Scrub), de la vitesse de lecture (Speed) ou du tempo auquel le fichier est lu (TapTempo). Chacun des modes, intégré dans Mono-Replay est compatible avec le découpage à trois niveaux décrit précédemment. Cela permet de positionner l'indice temporel  $\tau(t)$  au début du segment, de la mesure ou de la structure sélectionnée. Et ce, à n'importe quel moment de la performance. Cette fonction permet notamment de résoudre les problèmes de synchronisations que nous avons abordés dans la section 3.3.2.

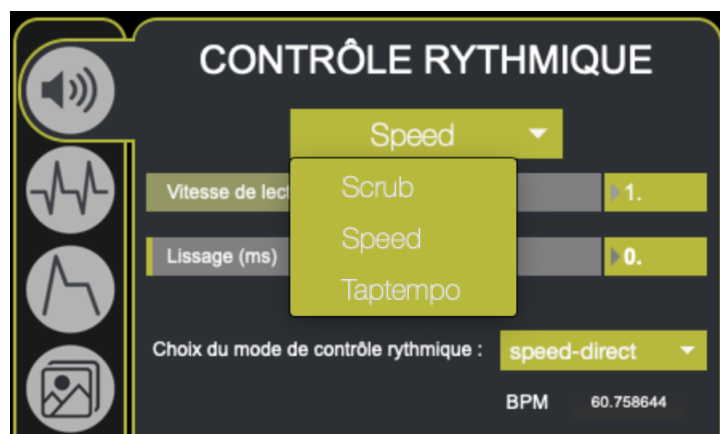


FIGURE 3.11 – Onglet pour sélectionner les grandes familles de contrôle temporel dans Mono-Replay. Il y en a trois : Speed qui regroupe les modes de contrôle temporel par paramétrage de la vitesse de lecture. Scrub qui permet de contrôler en direct la position de lecture d'un enregistrement. Taptempo qui regroupe tous les modes de contrôle basés sur le paramétrage de la vitesse de lecture par battue du tempo.

#### 3.4.1.1 Scrub

Le mode *Scrub* permet de contrôler directement la position de lecture de l'enregistrement à l'aide du vocodeur de phase *SuperVp*. Comme présenté dans la figure 3.12 L'utilisateur a accès à plusieurs sliders qui lui permettent de paramétrer :

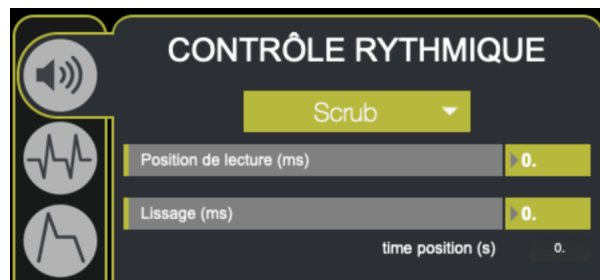


FIGURE 3.12 – Onglet pour contrôler la position de lecture de l’enregistrement à l’aide du mode *Scrub* dans Mono-Replay

- La *Position de lecture* à l’intérieur du segment sélectionné. Le slider peut prendre une valeur décimale, comprise entre 0 et 1 où 0 représente le début du segment et 1 la fin.
- La valeur de *lissage* en milli-seconde qui indique la durée que prend le vocodeur de phase pour passer d’une valeur de position à une autre. La transition suit une progression linéaire de la valeur initiale à la valeur finale avec un temps correspondant à la valeur de lissage.

Le mode *Scrub* s’inspire des contrôles temporels intégrés dans les logiciels de DJ qui permettent de reproduire des scratches en contrôlant de façon continue la position de la tête de lecture dans l’enregistrement. Néanmoins, le vocodeur de phase permet ici de ne pas avoir de transposition de pitch lorsque l’on étire temporellement l’enregistrement. Et ce, à la différence des platines analogiques qui modulent la hauteur lorsque la vitesse de lecture varie.

On verra dans la section suivante de ce travail une application de ce mode de contrôle pour le déroulement d’un enregistrement de *Accidents/Harmoniques* de Bernard Parmegiani. Dans ce scénario, on utilise un geste circulaire pour avancer segment par segment dans l’enregistrement.

### 3.4.1.2 Speed

L’onglet *Speed* regroupe tous les modes de contrôle temporel qui permettent de modifier directement la vitesse de lecture de l’enregistrement. Ces sous-modes sont au nombre de 5 et sont accessibles *via* un menu déroulant

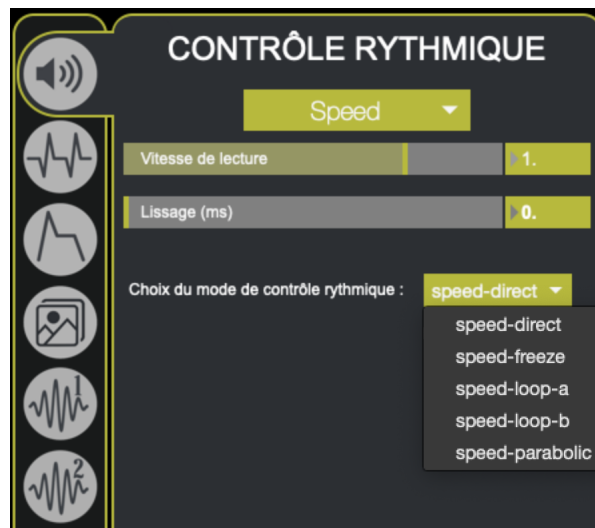


FIGURE 3.13 – Onglet pour contrôler la vitesse de lecture d'un enregistrement dans Mono-Replay. À l'aide du sous onglet en bas l'utilisateur peut sélectionner parmi les 5 modes de contrôle de la vitesse de lecture. À l'aide des sliders il peut ensuite régler la vitesse de lecture et le temps de lissage pour passer d'une valeur de vitesse à une autre.

figurant dans l'onglet *Speed*. Ces modes sont les suivants :

- *Speed-direct* : Ce mode est une application pratique du mode "Contrôle continu de la vitesse" décrit dans la sous-section 2.3.2. Il permet d'avoir un contrôle direct et continu de la vitesse de lecture d'un enregistrement. À cet effet, l'utilisateur peut configurer la vitesse de lecture à l'aide du slider présent sur la figure 3.13 ou en passant par le protocole OSC (décrit dans la section 3.8). Le slider permet de régler la vitesse de lecture entre -2 et 2. À 2 le fichier est lu avec une vitesse égale à deux fois la vitesse de lecture nominale et à -2 le fichier est lu avec une vitesse égale à deux fois la vitesse de lecture nominale en sens inverse. Ce mode s'inspire des contrôles disponibles dans les logiciels de DJ permettant de régler la vitesse de lecture d'un fichier compte tenu d'une valeur de tempo. Dans Mono-Replay la valeur de tempo est calculée par la méthode *Multi-feature Beat Tracking*[59] et est multipliée par la valeur de vitesse indiquée par le slider. On a ainsi un retour visuel indiquant la valeur de tempo correspondant à la vitesse de lecture sélectionnée.
- *Speed-freez* : Ce mode fonctionne comme le précédent, mis à part le

fait qu'en fin de segment, la vitesse de lecture passe automatiquement à zéro. Cela produit un effet de gel audio et permet à l'utilisateur de marquer un temps avant de passer au segment suivant. Le passage au segment suivant se fait obligatoirement par contrôle de l'utilisateur. Si on souhaite dérouler l'intégralité de l'enregistrement, une implication active de l'utilisateur est nécessaire pour passer au segment suivant. Par exemple, si l'indexation sélectionnée représente les mesures de l'enregistrement alors il sera nécessaire que l'utilisateur indique le passage à la mesure suivante pour continuer la lecture de l'enregistrement.

- *Speed-loopA* : Ce mode lit le fichier audio en boucle pour une vitesse sélectionnée. La lecture s'effectue selon trois repères temporels placés dans le segment. Le premier est placé au début du segment, le deuxième au milieu et le troisième en fin de segment. La lecture débute au premier repère puis une fois le troisième repère atteint la tête de lecture revient au deuxième repère. Dès lors la boucle audio s'effectue entre le deuxième repère et le troisième. La position du deuxième repère temporel est paramétrable directement depuis l'interface. Par exemple si le paramètre est réglé sur 0.5 la boucle débutera à la moitié du segment sélectionné. La boucle se terminera néanmoins toujours au troisième repère temporel. Comme pour les modes précédents, la vitesse de lecture est réglable par le même slider et permet de lire le fichier dans le sens normal pour les valeurs de vitesse positives et dans le sens inverse pour les valeurs de vitesses négatives.
- *Speed-loopB* : Ce mode joue aussi en boucle le segment sélectionné selon les trois positions temporelles décrites précédemment. La lecture débute au début du segment, puis une fois qu'elle arrive en fin de segment, la tête de lecture revient en arrière jusqu'au repère temporel situé au milieu du segment puis repart vers la fin du segment et ainsi de suite. Selon le même fonctionnement que le mode *Speed-loopA*, il est possible de régler la position du deuxième repère temporel à l'aide de l'interface utilisateur.



- *Speed-parabolic* : Il s'agit d'une intégration pratique du mode de contrôle temporel *synchrone convexe* décrit dans la section 2.3.3.1. Le réglage de la vitesse permet de régler le paramètre  $v$  de l'équation de la parabole décrite dans cette section.

Il est à noter que le mode *Speed-parabolic* a été implémenté après que le travail d'harmonisation des modes de contrôle temporel autour du formalisme décrit dans la section 2.3.2 a été réalisé. Cette intégration a été facilitée vu que les entrées et les sorties de T-Voks s'exprimaient déjà avec les variables décrites dans la partie 2.3.2. Puisque la majorité des modes de contrôle temporel déjà existants peuvent aussi s'exprimer selon ces variables, il devient facile d'intégrer de nouveaux modes à Mono-Replay.

#### 3.4.1.3 TapTempo



FIGURE 3.14 – Onglet pour contrôler la vitesse de lecture d'un enregistrement en tapant son Tempo dans Mono-Replay. À l'aide du sous-onglet à gauche l'utilisateur choisi un mode de TapTempo parmi les 5. Il peut ensuite battre la pulsation à l'aide de la touche K de son ordinateur pour régler la vitesse de lecture.

Les 5 modes de contrôle temporel disponibles dans l'onglet "TapTempo" sont accessibles grâce au sous-menu présenté dans la figure 3.14. Ces modes permettent de contrôler de façon discrète la vitesse et la position de lecture de l'enregistrement en battant un tempo. Le terme "Tap Tempo" fait référence aux algorithmes d'estimation de tempo par la battue de celui-ci. On retrouve ces algorithmes dans de nombreux logiciels de type Digital Audio Workstation (DAW) comme Ableton Live, Steinberg Cubase ou encore Reason. Cette fonc-

tion permet de régler le tempo d'une "session" en appuyant sur une touche du clavier de façon périodique. La pulsation "tapée" par l'utilisateur est alors déterminée et sert à configurer la valeur de tempo de la session. Aussi très utilisée dans les applications de Dj-ing, cette fonctionnalité permet de facilement régler la vitesse d'un morceau en tapant la pulsation à laquelle on souhaite le voir se dérouler. De nombreuses applications web intègrent aussi ce type d'algorithme<sup>2</sup>.

C'est dans notre volonté de créer un système intégrant des contrôles temporels expressifs que nous avons ajoutés dans Mono-Replay la possibilité de contrôler le déroulement temporel d'un enregistrement compte tenu d'une pulsation "jouée". À la manière d'un chef d'orchestre indiquant les évolutions de tempo à des instrumentistes, l'utilisateur peut ici grâce à ces modes de contrôle temporel contrôler la vitesse de lecture en battant la mesure. Contrairement aux modes décrits précédemment, le contrôle de la vitesse ne s'effectue pas de façon directe. En effet pour faire varier la vitesse de lecture, il est nécessaire de préalablement déterminer une valeur de tempo compte tenu de la performance de l'utilisateur. Cette différence est importante car elle nécessite l'ajout d'une fonction interne supplémentaire décrite dans la section 2.3.3. Cette fonction permet d'analyser la performance d'un utilisateur pour la transcrire en valeur de vitesse relative au tempo battu. Pour battre le tempo, l'utilisateur peut utiliser le bouton "Tape un tempo" présenté sur la figure 3.14 ou y accéder par une variable OSC.

Il est à noter que tous les modes présentés ici fonctionnent seulement pour les musiques pulsées dont l'indexation a été effectuée préalablement de manière à indiquer les positions temporelles de chaque pulsation. Il est aussi à noter que pour chaque méthode présentée ci-dessous, la fonction d'estimation de la pulsation s'effectue par moyenne glissante, selon la méthode présentée dans la section 2.3.3. Pour paramétrer la fenêtre glissante de cette moyenne, l'utilisateur peut régler la plage de valeurs sur laquelle elle est calculée grâce au slider "Moyenne glissante" disponible *via* l'interface utilisateur présenté figure 3.14. Les modes disponibles dans l'onglet "TapTempo" sont les suivants :

---

2. De nombreux outils d'estimation de tempo par battue de pulsation sont disponibles en ligne. Les modes TapTempo de Mono-Replay s'en inspirent : <http://www.all18.com/tools/bpm.htm>

- *TapTempo-direct* : Ce mode de contrôle temporel est décrit mathématiquement dans la section 2.3.3.1. Il est dit "asynchrone linéaire" car son utilisation n'induit pas un changement de phase et qu'il fait évoluer l'indice temporel  $\tau(t)$  de façon linéaire. Ce mode permet de changer la vitesse de lecture d'un enregistrement en calculant une moyenne glissante sur  $n$  intervalles temporels. Ces intervalles correspondent à l'écart temporel entre deux pulsations consécutives battue par l'utilisateur à l'aide du bouton "Tape un tempo" présenté sur la figure 3.14. Pour déterminer cette vitesse on fait le rapport suivant :  $v = \frac{\bar{T}_n}{T_{tempo}}$ . Lorsque  $v = 1$  le fichier est lu à la vitesse nominale. Lorsque  $v=2$  le fichier est lu à une vitesse deux fois supérieur à la vitesse nominale.  $T_{tempo}$  correspond à la période séparant deux pulsations pour une valeur de tempo déterminé à l'aide de l'algorithme MultiFeature Beat Tracking [59].  $\bar{T}_n$  correspond à la moyenne glissante sur  $n$  période séparant deux impulsions battues par un utilisateur.
- *TapTempo-directstep* : Ce mode fonctionne de la même manière que le mode décrit précédemment, à la différence près que la nouvelle valeur de vitesse est effective seulement lorsque la position de la tête de lecture atteint le début du segment suivant. Ce mode permet par exemple d'anticiper une transition de tempo au temps précédent.
- *TapTempo-next* : Ce mode de contrôle temporel est décrit mathématiquement dans la section 2.3.3.1. Il est dit "synchrone linéaire" car il permet de modifier l'évolution de l'indice temporel de façon linéaire et qu'il induit un changement de phase à son utilisation. Ce mode estime aussi un tempo calculé par moyenne glissante sur  $n$  battues. Néanmoins, à la fin de chaque segment, la vitesse de lecture passe à zéro, ce qui produit un "gel" audio. Pour passer au segment suivant l'utilisateur doit de nouveau battre une pulsation. Ainsi avec le même geste, l'utilisateur contrôle la variation de vitesse ainsi que la phase. Ce mode est très interactif et nécessite une participation plus active de l'utilisateur pour dérouler l'enregistrement dans sa globalité.

- *TapTempo-fit* : Ce mode de contrôle temporel est décrit mathématiquement dans la section 2.3.3.1. Il est dit "linéaire continu" car il permet de faire évoluer l'indice temporel de façon linéaire. Il tient aussi compte dans le calcul de l'indice temporel de la position de lecture de l'enregistrement par rapport au découpage en segment sans inclure de discontinuité, c'est pourquoi on dit qu'il est continu. L'estimation du tempo calculé par moyenne glissante permet de calculer la vitesse de lecture de l'enregistrement selon le rapport suivant :  $v = \frac{\bar{T}_n}{T_{segment}}$ .  $\bar{T}_n$  correspond à la moyenne glissante sur  $n$  période séparant deux impulsions battues par un utilisateur.  $T_{tempo}$  correspond à la période séparant deux pulsations pour une valeur de tempo déterminée à l'aide de l'algorithme MultiFeature Beat Tracking [59].  $T_{segment}$  correspond à la durée d'un segment séparant deux pulsations consécutives détectées à l'aide de l'algorithme MultiFeature Beat Tracking [59]. Cette méthode tient compte des courtes variations de tempo. Cette méthode ne tient pas compte du tempo global estimé sur l'ensemble de l'enregistrement, mais tient compte de la durée de chaque segment et le compare à la durée entre chaque pulsation effectuée par l'utilisateur.
- *TapTempo-parabolic* : Ce mode de contrôle temporel est décrit mathématiquement dans la section 2.3.3.1. Il est dit "synchrone parabolique" car il permet de faire varier l'indice temporel selon une parabole. Il permet de faire varier la phase de l'enregistrement et n'induit pas de discontinuité dans son calcul d'indice temporel. L'estimation du tempo calculé par moyenne glissante permet de calculer la vitesse de lecture de l'enregistrement par le rapport suivant :  $v = \frac{\bar{T}_n}{T_{tempo}}$ . Lorsque  $v=1$  le fichier est lu à la vitesse nominale. Lorsque  $v=2$  le fichier est lu à une vitesse deux fois supérieure à la vitesse nominale.  $T_{tempo}$  correspond à la période séparant deux pulsations pour une valeur de tempo déterminé à l'aide de l'algorithme MultiFeature Beat Tracking [59].  $\bar{T}_n$  correspond à la moyenne glissante sur  $n$  période séparant deux impulsions battues par un utilisateur. La vitesse d'articulation intervient dans le calcul de la parabole qui guide l'évolution de la vitesse.

En pensant ces modes selon les variables décrites dans la section 2.3.2, on s'assure de leurs compatibilités et de leurs facilités d'intégration dans Mono-Replay. L'estimation du tempo battu par l'utilisateur s'effectue systématiquement par calcul d'une moyenne glissante sur plusieurs intervalles relevés lors d'une performance. On pourrait imaginer intégrer d'autres méthodes d'estimation de tempo comme celles décrites dans la section 2.3.3. Il serait alors possible de combiner des modes de contrôle temporel par battue de pulsation et des modes d'estimation de tempo différents de celui déjà existant dans le logiciel.

### 3.4.2 Naviguer dans l'enregistrement selon trois niveaux d'indexation

Comme nous l'avons vu dans la section 3.3.2, pour manipuler de façon confortable le déroulement temporel d'un enregistrement, il faut pouvoir se déplacer instantanément à différents moments de celui-ci. Comme décrit précédemment, cela peut être une solution au problème de synchronisation lors de l'utilisation de plusieurs Mono-Replay de façon simultanée. L'organisation de ce découpage est expliquée dans les parties précédentes. Ici nous détaillons comment l'utilisateur a accès au contrôle permettant d'utiliser ce découpage au sein du logiciel.

L'utilisateur peut positionner le début de la tête de lecture selon l'indexation à trois niveaux à l'aide des boutons de l'interface présentés sur la figure 3.15. Cette indexation est organisée de la façon suivante :

- *Segment* : Le segment représente le plus petit découpage dans Mono-Replay. Si on utilise la méthode de détection de transitoire SuperFlux développée par [6] disponible dans l'éditeur de Mono-Replay [59], les segments représenteront les portions de l'enregistrement séparées par deux transitoires consécutifs. Si on utilise la méthode MultiFeature Beat Tracking, les segments représenteront les portions du signal séparées par deux pulsations consécutives. On peut aussi créer manuellement ses propres collections de segments.

- *Mesure* : La mesure est le niveau intermédiaire de découpage dans Mono-Replay. Il correspond à un découpage selon des groupes de segments. Pour créer ce découpage, l'utilisateur doit renseigner un nombre entier qui correspond au nombre de segments inclus dans une mesure. Tous les  $n$  segments, une nouvelle mesure est créée. Si le premier niveau d'indexation est obtenu à l'aide de la méthode MultiFeature Beat Tracking, alors la mesure dans Mono-Replay se rapproche de la mesure dans la théorie musicale, à savoir un certain nombre de pulsations.
- *Structure* : La structure est le dernier niveau de découpage dans Mono-Replay. Elle permet de placer des ancrs temporelles manuellement à une échelle temporelle plus élevée que celle du segment ou de la mesure. Il est d'usage de placer ces ancrs sur le début des structures musicales de l'enregistrement pour se déplacer dans celui-ci selon les refrains couplet etc.

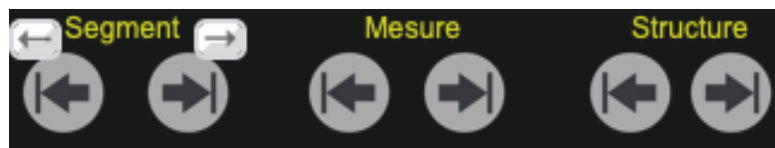


FIGURE 3.15 – Mono-Replay permet de naviguer dans un enregistrement selon trois niveaux d'indexation : le segment, la mesure et la structure

### 3.5 Gestion des transformations spectrales

Mono-Replay intègre plusieurs fonctionnalités issues du moteur audio SuperVP [50]. Ce vocoder de phase permet d'effectuer des transformations spectrales de haute définition pour étirer ou compresser temporellement un enregistrement. Il permet aussi de configurer d'autres paramètres de la synthèse de manière à transposer la hauteur, l'enveloppe spectrale ou encore mélanger les composantes harmoniques et inharmoniques de l'enregistrement.

Les possibilités de contrôle temporel offertes par Mono-Replay nous permettent d'imaginer des façons originales de contrôler ces transformations. Un

système complexe d'enveloppe, appliqué à chaque segment, permet de piloter les variations des paramètres de transformations spectrales selon le découpage effectué dans Mono-Replay. Par exemple, on peut dessiner une enveloppe pour contrôler la variation de Pitch et suivre son évolution, selon la position de lecture dans un segment. L'enveloppe se réinitialisera au début du prochain segment et ainsi de suite.

### 3.5.1 Contrôles des paramètres SuperVP

SuperVP a été conçu pour produire des effets d'étirements temporels, tout en minimisant les artefacts dus à la présence de transitoire. L'algorithme intègre une méthode de détection des parties transitoires d'un son. En relevant les positions temporelles de ceux ci dans l'enregistrement il applique un traitement différent au moment de la re-synthèse du son. Ainsi on peut étirer ou compresser le son tout en conservant le caractère bref des transitoires.

SuperVp est distribué sous la forme d'un objet compatible avec l'environnement Max/MSP. On peut le paramétrer dans Mono-Replay à l'aide de l'interface présentée dans la figure 3.16. Chaque slider de cette interface permet de régler une fonction de l'objet SuperVP intégré dans le logiciel.

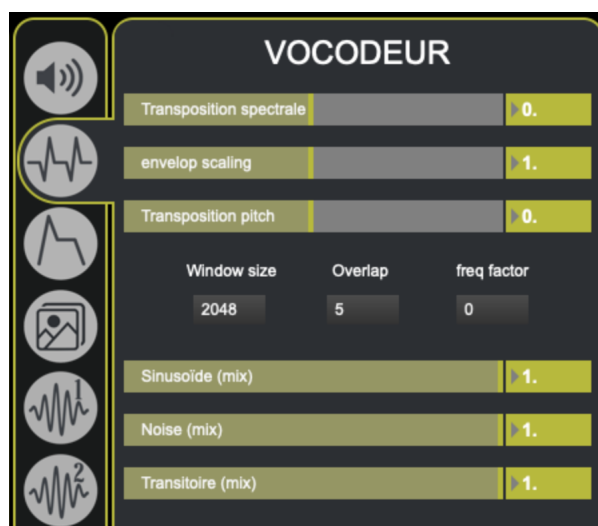


FIGURE 3.16 – Interface de contrôle des paramètres de transformations spectrales de SuperVP dans Mono-Replay

On peut à l'aide de ces sliders contrôler les transformations suivantes :

- *Transposition de pitch* : Le vocodeur de phase SuperVp est équipé d'un système complexe permettant de transposer la hauteur en conservant l'enveloppe spectrale [51]. Pour régler la valeur de cette transposition, l'interface utilisateur de Mono-Replay dispose d'un slider qui configure la valeur de transposition selon une valeur d'intervalle musicale exprimée en *cents*. Les valeurs que peut prendre ce paramètre varient entre  $-3600cents$  et  $3600cents$ .
- *Transposition de l'enveloppe spectrale* : Il est aussi possible de transposer l'enveloppe grâce à SuperVP. Pour configurer cette transformation on utilise un slider dont les valeurs varient aussi entre  $-3600cents$  et  $3600cents$ .
- *Timbre* : Ce paramètre permet de changer le timbre du son synthétisé en faisant varier la répartition harmonique du signal.
- *Mixage des transitoires, des sinusoïdes et du bruit* : SuperVp possède des fonctionnalités grâce auxquelles il est possible de répartir le mixage des composantes harmoniques, inharmoniques et transitoires du son. Grâce à ces fonctionnalités, on peut par exemple laisser seulement les composantes harmoniques lors de la re-synthèse de l'enregistrement et ainsi entendre seulement l'attaque de chaque son. Pour configurer ces paramètres, l'utilisateur a accès à des sliders qui contrôlent le mélange de ces composantes sur une échelle de valeur allant de 0 à 1.

### 3.5.2 Éditeur d'enveloppe de transformation

Programmer l'évolution des paramètres de transformation audio selon le déroulement temporel d'un enregistrement est une fonction habituelle des DAW connue sous le nom d'automation. Pendant le développement de Mono-Replay, on a réfléchi à des moyens pour effectuer ce type d'opération en tirant profit de l'indexation de l'enregistrement effectuée en amont. Le système que nous décrivons ici permet de telles opérations. Il s'agit d'un générateur d'enve-



loppe que l'on peut paramétrer, puis relier aux paramètres de transformations, listés ci-dessus, de manière à contrôler leur évolution selon les valeurs prises par l'enveloppe. La valeur de l'enveloppe est multipliée par la valeur du paramètre à contrôler pour donner une nouvelle valeur.

Cette enveloppe est générée par modulation de l'amplitude et de la fréquence d'une fonction sinusoïdale. Pour contrôler ces modulations, on utilise deux enveloppes intermédiaires  $E_1$  et  $E_2$ , déjà présentées figure 2.20, qui vont faire varier l'amplitude et la fréquence selon la forme qu'elles dessinent. La durée de l'enveloppe varie d'un segment à l'autre et correspond à la durée du segment lu. L'équation de cette enveloppe est présentée section 2.4.4.

## 3.6 Gestion des mémoires

La gestion des mémoires de Mono-Replay est commune à tous les logiciels de la suite Mono développés par PUCE MUSE. Cette fonction permet de sauvegarder un état des paramètres de contrôle du logiciel ainsi que de le rappeler. Pour les rappeler on peut se servir d'un menu déroulant mais aussi d'un système complexe d'interpolation que nous décrivons dans la suite de cette partie.

### 3.6.1 Pattern storage

La gestion de mémoire dans les Mono se base sur les fonctions de l'objet pattern storage disponible dans Max. Cet objet permet de sauvegarder plusieurs états d'un même paramètre et de les rappeler à tout moment. Dans les Mono un bouton permet de sauvegarder l'état de ces paramètres. Les valeurs de la variable sont ensuite enregistrées dans un fichier texte accessible dans le répertoire projet de l'ordinateur sur lequel est installé le logiciel. Ainsi, à chaque utilisation du logiciel, l'utilisateur peut retrouver les précédentes mémoires enregistrées.

### 3.6.2 Interpolation de mémoires

La sauvegarde d'état de chaque variable du logiciel dans une même mémoire est très pratique dans l'utilisation des Mono. On peut grâce à elles,

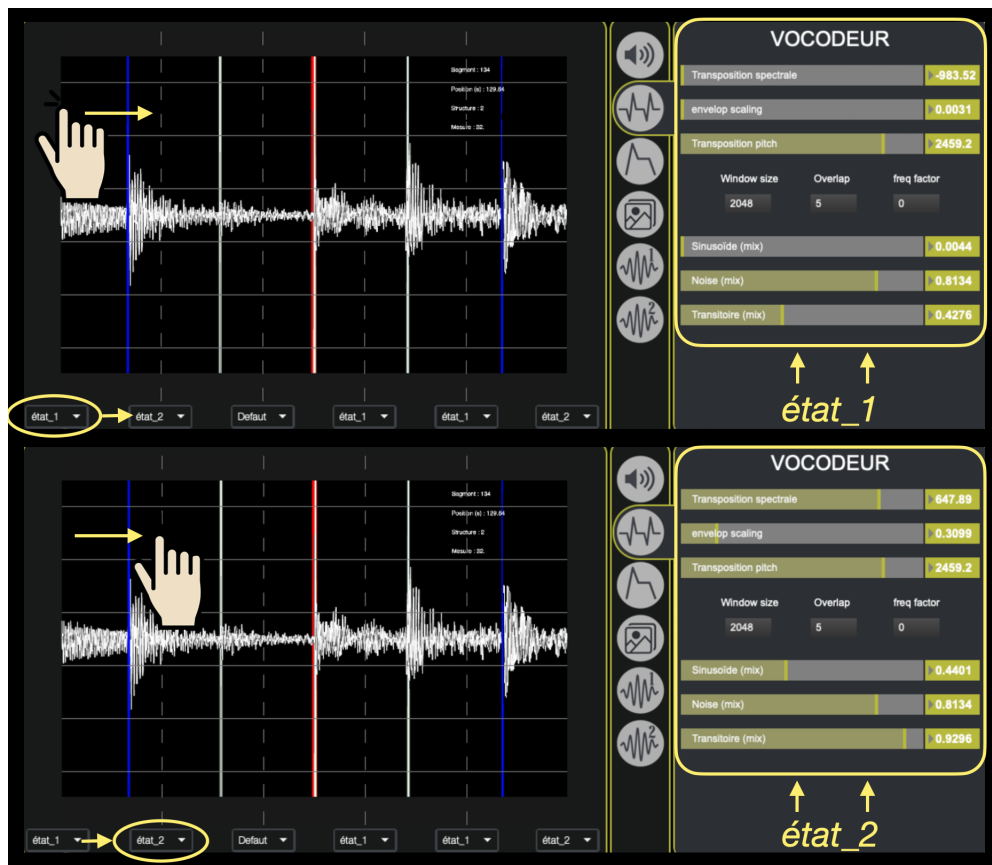


FIGURE 3.17 – Représentation schématique du système d’interpolation présent dans les logiciels Mono. En glissant le curseur de la souris sur la ligne d’un paramètre on passe d’un état 1 (figure du dessus) des paramètres du vocodeur de phase à un état 2 (figure du dessous) par interpolation linéaire dans Mono-Replay.

passer d’un réglage du logiciel à un autre en une seule action. Dans le cas de Mono-Replay les mémoires intègrent des variables, comme par exemple la sélection d’un mode de contrôle temporel ou la sélection d’un enregistrement. Ainsi on peut changer de morceau et de mode de contrôle temporel durant une performance simplement en sélectionnant un état de cette mémoire.

PUCE MUSE a développé un système original pour rappeler ces états de mémoires de façon plus expressive. Ce système prend la forme d’une grille d’interpolation qui permet de passer d’une mémoire à une autre en déplaçant verticalement le curseur de la souris sur la fenêtre de rendu de l’interface utilisateur comme indiquée sur la figure 3.17. La grille d’interpolation est organisée de telle sorte que les lignes verticales représentent des états de mémoire et les

lignes horizontales des familles de paramètres. Ainsi, en glissant horizontalement le curseur de la souris, on peut changer un état de mémoire pour une même famille de paramètres. Si on souhaite changer de famille de paramètres, il faut alors changer de ligne verticale.

Pour éviter d'avoir des discontinuités lors du changement d'une mémoire à l'autre, une fonction linéaire permet de changer progressivement l'état d'un paramètre d'une mémoire à l'autre. Les barres verticales de la grille d'interpolation représentent la valeur du paramètre à une mémoire donnée. En bougeant la souris de façon horizontale entre deux barres verticales de cette grille, on génère une interpolation linéaire entre deux valeurs d'un paramètre.

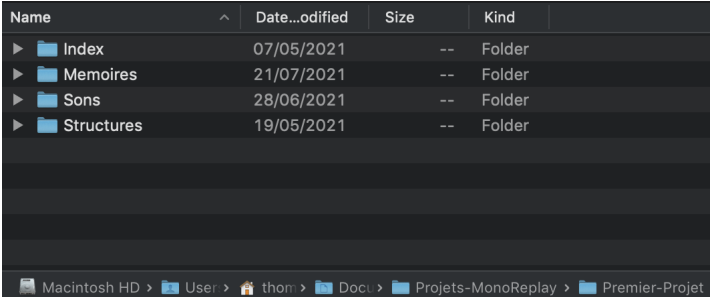
### 3.7 Création de projet

Dans les *Mono* on appelle "Projet" le répertoire stockant les ressources externes au logiciel, nécessaires pour son utilisation. Il est accessible directement depuis l'interface utilisateur et permet d'enregistrer un ensemble de données propre à une utilisation du logiciel. Chaque *Mono* utilise des types de médias différents et l'organisation à l'intérieur du projet est propre à chaque logiciel. Dans Mono-Replay, le répertoire projet permet de stocker les sons, les fichiers d'analyses, les fichiers d'indexations et les mémoires nécessaires pour le bon fonctionnement du logiciel. Le répertoire est organisé comme sur la figure 3.18 et possède les dossiers suivants :

- *Index* : il s'agit du répertoire stockant les fichiers d'indexation créés dans Mono-Replay. Ce répertoire contient aussi un dossier BPM stockant la valeur de BPM dans le cas où la méthode *Multifeature BeatTracking* est lancée. Comme présenté dans la section (section 3.3), lorsque l'on enregistre un fichier d'indexation dans Mono-Replay un fichier texte se crée automatiquement à l'intérieur du dossier "Index" et le fichier d'analyse est accessible *via* l'interface graphique du logiciel à l'aide du menu déroulant en bas au centre comme dans la figure 3.1.
- *Mémoires* : il s'agit du répertoire contenant les mémoires d'état de chaque paramètre. Si on regarde par exemple la figure 3.17, une mé-

moire d'état correspond à l'état 1 et à l'état 2 des paramètres du vocodeur de phase.

- *Sons* : il s'agit du répertoire contenant tous les enregistrements du projet au format wav. On peut glisser/déposer un fichier audio sur l'interface audio pour qu'il se range automatiquement dans le répertoire sons. Lorsqu'un projet est sélectionné, les fichiers audio présents dans ce répertoire sont accessibles *via* le menu déroulant en bas à gauche de l'interface utilisateur présentée figure 3.1.
- *Structures* : il s'agit du répertoire stockant les fichiers d'indexation pour une indexation à la structure.



Name	Date...odified	Size	Kind
▶ Index	07/05/2021	--	Folder
▶ Memoires	21/07/2021	--	Folder
▶ Sons	28/06/2021	--	Folder
▶ Structures	19/05/2021	--	Folder

Macintosh HD > User > thom > Docu > Projets-MonoReplay > Premier-Projet

FIGURE 3.18 – Dans le logiciel Mono-Replay, un projet est organisé en quatre sous-dossiers. Le dossier "Index" contient les mémoires d'indexation de segments sous la forme de fichier texte ; Le dossier "Mémoires" contient les différentes mémoires d'état décrit dans la section 3.6 sous forme de fichier json ; Le dossier "Sons" contient les enregistrements manipulés au format WAV ; le dossier "Structures" contient les mémoires d'indexation de structures sous la forme de fichier texte.

Pour créer un nouveau projet, il faut partir d'un répertoire vide avec l'architecture déjà existante comme sur la figure 3.18. Une fois ce répertoire créé il faut le placer à la racine du répertoire "Documents" sur l'ordinateur de l'utilisateur pour qu'il soit visible dans Mono-Replay. Toutes les autres manipulations pour concevoir les ressources nécessaires à la création d'une interaction sont accessibles directement dans Mono-Replay.

### 3.8 Dictionnaire OSC et mapping

Le protocole "Open Sound Control" permet d'envoyer et recevoir des informations de contrôle en temps réel, depuis un ou plusieurs systèmes compatibles. Plus rapide et facile à utiliser que le MIDI, ce protocole équipe de plus en plus les synthétiseurs, logiciels de création ou robots. Les logiciels Mono intègrent les fonctionnalités nécessaires pour utiliser ce protocole de communication. Cela permet entre autre d'accéder aux variables du logiciel *via* des logiciels et des interfaces gestuelles externes.

Chaque commande disponible sur l'interface utilisateur de Mono-Replay est accessible *via* une adresse OSC. Ces adresses sont listées et stockées dans un dictionnaire qui fait état de toutes les variables pilotables par ce procédé. Une fois le dictionnaire élaboré, il n'y a plus qu'à établir un port de communication entre Mono-Replay et le logiciel qui le contrôlera pour pouvoir paramétrer les variables souhaitées.

Mono-Map est un logiciel spécialement conçu par PUCE MUSE pour contrôler en OSC les paramètres de chaque Mono. Il prend la forme d'un tableau représenté figure 3.19 ou chaque ligne permet de :

- Définir une interface d'entrée
- Sélectionner un fichier de mapping gestuel. Dans ce fichier est contenu le routage nécessaire pour que les données de contrôle soient liées aux variables du logiciel que l'on souhaite contrôler.
- Accéder à une représentation visuelle du mapping

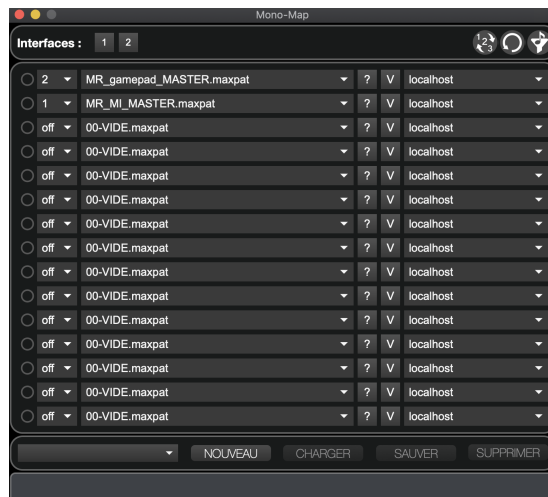


FIGURE 3.19 – Mono-Map est un outil qui permet de lancer plusieurs mappings gestuels simultanément afin de contrôler les logiciels édités par PUCE MUSE. Ce logiciel communique avec les logiciels de la suite Mono à l’aide du protocole OSC

## Chapitre 4

# Cas d'usage de Mono-Replay

### Sommaire

---

<b>4.1</b>	<b>Contrôler Mono-Replay à l'aide d'une manette de jeu vidéo</b>	<b>120</b>
4.1.1	Ateliers Mono-Replay . . . . .	123
4.1.2	Animer un enregistrement multipiste du morceau <i>Superstition</i> par Stevie Wonder . . . . .	125
4.1.3	Création du morceau <i>Bar Mono-Replay</i> par Pehoz . . . . .	127
<b>4.2</b>	<b>Contrôler Mono-Replay à l'aide du Méta-Instrument 4 . .</b>	<b>129</b>
4.2.1	Le Méta-Instrument 4 . . . . .	129
4.2.2	Animer un enregistrement du <i>Trio n°2</i> de Franz Schubert par le Beaux Arts Trio en battant une pulsation . . . . .	131
4.2.3	Animer <i>Accidents/Harmoniques</i> de Bernard Parmegiani en contrôlant de façon continue la position de lecture . . . . .	136
<b>4.3</b>	<b>Animer deux pièces de musiques électroniques composées par Nicolas Rivière . . . . .</b>	<b>138</b>
4.3.1	<i>SUPERNOME DIVA 2 - Contemplation du vide</i> . . . . .	139
4.3.2	<i>SUPERNOME DIVA 2.5 - Anatomie du vide</i> . . . . .	139
4.3.3	Dispositif et mappings . . . . .	141

---

Comme nous venons de le voir dans la partie précédente, Mono-Replay intègre l'architecture et les fonctionnalités que nous jugeons nécessaires pour concevoir un nombre important d'interactions musicales expressives basées sur le contrôle temporel d'enregistrement. Pour vérifier cela, nous avons conçu plusieurs scénarios d'interaction. Ces interactions font à chaque fois intervenir

une interface gestuelle externe et le logiciel. Nous proposons de les détailler dans ce chapitre. Pour plus de clarté, nous groupons ces interactions en fonction de l'interface gestuelle qu'elles utilisent. La plupart d'entre elles sont des interfaces habituelles de l'environnement PUCE MUSE. Nous expliquerons pourquoi elles ont été choisies et comment elles ont été utilisées pour contrôler Mono-Replay.

#### 4.1 Contrôler Mono-Replay à l'aide d'une manette de jeu vidéo

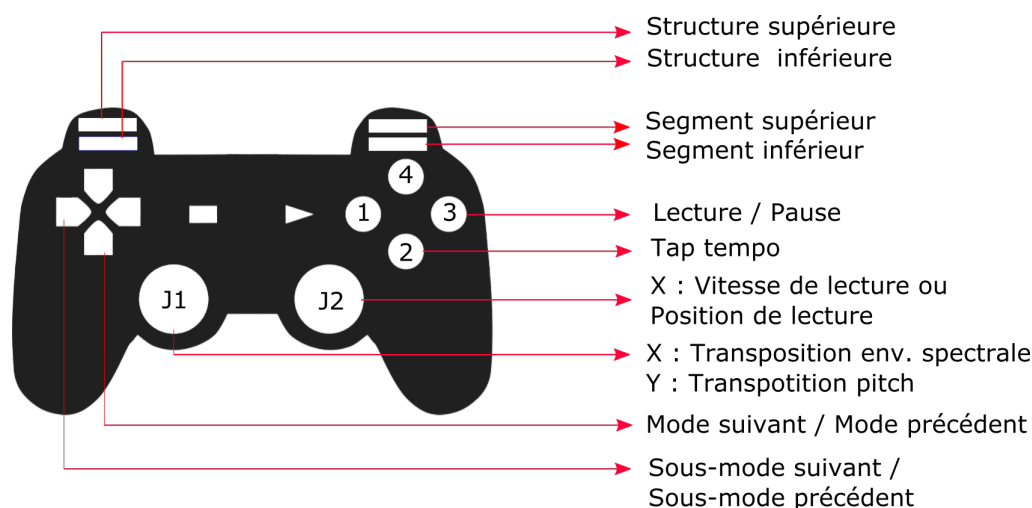


FIGURE 4.1 – Schéma d'un mapping permettant de contrôler le logiciel Mono-Replay à l'aide d'une manette DualAnalog.

Depuis plusieurs années, certains musiciens détournent les manettes de jeu vidéo pour contrôler des algorithmes de transformation sonore. Au début des années 1990, un objet Max intitulé "glove" a permis à de nombreux musiciens d'utiliser la manette Nintendo "Mattel Power Glove" comme interface gestuelle de contrôle. En les couplant avec des logiciels de transformation sonore, les manettes de jeu vidéo deviennent de véritables instruments de musique électronique. Dans cet article [2] Ariza relie les contrôles d'une manette DualAnalog à différents paramètres de transformation sonore. Ainsi la manette devient un échantillonneur polyphonique, un générateur de boucle ou encore un générateur polyphonique de bruit. L'accessibilité de l'interface en



fait un outil facile à prendre en main. C'est d'ailleurs la motivation première de l'auteur. La plupart des nouvelles interfaces de contrôle développées ces dernières années sont coûteuses et demandent beaucoup de temps d'apprentissage. Alors que la manette est très bon marché et demeure un objet familier pour la plupart des personnes.



FIGURE 4.2 – L'installation Org'n One permet à un musicien placé sous un écran de contrôler le son et l'image d'une séquence d'orgue préenregistrée à l'aide d'une manette de jeu. Accompagnée par l'orgue et les musiciens de l'ensemble ONE, cette installation a été mise en œuvre dans l'église Sainte-Élisabeth de Hongrie située à Paris durant Nuit Blanche 2021.

L'utilisation des manettes de jeu pour contrôler des transformations sonores est une pratique très courante dans les activités de PUCE MUSE [43]. Comme pour l'exemple précédent, la familiarité de l'interface permet à des utilisateurs novices de facilement prendre en main les fonctionnalités d'un logiciel qu'ils viennent de découvrir. Dans l'installation Org'n One par exemple, dont une photo est visible figure 4.2, 8 musiciens contrôlent, à l'aide d'une manette, la position et la vitesse de lecture d'une séquence d'orgue préenregistrée. En trois répétitions seulement, les musiciens ont pu découvrir le logiciel, prendre en main l'interface et créer un discours musical cohérent d'une vingtaine de minutes qu'ils ont ensuite rejouée publiquement lors de Nuit Blanche 2021 à Paris dans l'église Sainte-Élisabeth de Hongrie.

Dans chaque atelier mis en place par PUCE MUSE, l'utilisation de la ma-

nette offre un gain de temps considérable dans la prise en main des outils. En quelques minutes, on comprend les possibilités qui sont offertes par l'algorithme que l'on contrôle. C'est d'ailleurs pour ces raisons que le DualAnalog est une interface par défaut pour tous les logiciels développés par PUCE MUSE. Pour explorer les possibilités offertes par ce type d'interface pour contrôler la lecture d'un enregistrement, nous avons réalisé plusieurs mappings gestuels, dont celui par défaut est présenté figure 4.1. Ce mapping permet de contrôler le logiciel Mono-Replay à l'aide d'une manette DualAnalog. Il relie les contrôles de l'interface aux fonctionnalités du logiciel de la façon suivante :

<i>Nom de la variable</i>	<i>Contrôle associé</i>
Gâchette droite supérieure	Passe au segment supérieur
Gâchette droite inférieure	Passe au segment inférieur
Gâchette gauche supérieure	Passe à la structure supérieure
Gâchette gauche inférieure	Passe à la structure inférieure
Gâchette gauche supérieure	Passe à la structure supérieure
Bouton 2	Règle la vitesse de lecture en tapant un tempo si le mode "Tap Tempo" est sélectionné.
Bouton 3	Lance ou arrête la lecture d'un enregistrement
Joystick droit X	Contrôle la transposition de l'enveloppe spectrale sur l'intervalle [-1200 cents ; 1200 cents]
Joystick droit Y	Contrôle la transposition du pitch sur l'intervalle [-1200 cents ; 1200 cents]
Joystick gauche X	Contrôle la vitesse de lecture dans le cas où le mode "speed" est sélectionné/ permet de contrôler la position de lecture dans le cas où le mode "scrub" est sélectionné

Les gâchettes et les boutons envoient une information discrète à chaque fois que l'utilisateur appuie dessus. Chaque joystick envoie un contrôle continu que l'on récupère pour contrôler les paramètres de transformation de SuperVp de la façon suivante :

$$\forall t \in \mathbb{N},$$

$$\sigma(t) = \Delta \cdot \alpha(t) + S_{min} \quad (4.1)$$

Avec  $\Delta = S_{max} - S_{min}$ ,  $S_{max}$  la valeur maximale que peut prendre le para-

mètre de transformation spectrale,  $S_{min}$  la valeur minimale et  $\alpha(t)$  la valeur de contrôle du joystick, comprise dans l'intervalle  $[0; 1]$ .

#### 4.1.1 Ateliers Mono-Replay

La conception de Mono-Replay fait partie des attentes de ce projet. Pour explorer les possibilités offertes par le logiciel, plusieurs ateliers pratiques ont été organisés avec 8 participants dont deux chercheurs (Christophe d'Alessandro et Gabriela Patino Lakatos), un doctorant (Grégoire Locqueville), un développeur informatique (Jérémy Gaston Raoul), un ingénieur de recherche (Hugues Genevois), le dirigeant de la société Eowave (Marc Sirguy) et le directeur artistique de l'association PUCE MUSE. Chaque utilisateur disposait d'une version compilée du logiciel sur son ordinateur personnel.

*Atelier 1* : Exploration de Mono-Replay au clavier et à la souris (deux séances). Ces séances ont permis de présenter chaque fonction du logiciel. Les participants disposaient d'une version compilée du logiciel qu'il manipulait au clavier et à la souris. L'idée était de créer un échange pour souligner les difficultés rencontrées lors de l'utilisation de l'outil. Ainsi ces échanges ont permis d'ajouter les trois fonctionnalités suivantes à Mono-Replay :

- 1 - Le niveau intermédiaire de segmentation dit "à la mesure" qui permet de naviguer selon un pas plus grand que le "segment" mais plus petit que la "structure". Ces niveaux de segmentation sont détaillés dans la section 3.3.2. Durant les ateliers plusieurs utilisateurs ont rencontré des problèmes pour se déplacer rapidement à une position précise. La segmentation "à la structure" permet de se positionner rapidement à des points clefs de l'enregistrement, comme le début d'un refrain ou le début d'une outro. Le découpage "au segment", quant à lui, permet de se déplacer événement par événement dans l'enregistrement. Il n'existait pas de niveau de segmentation intermédiaire. Lors des ateliers, il arrivait souvent que les participants veuillent tous se synchroniser sur le même segment. C'est pour cette raison que cette étape de segmentation a été ajoutée. Ainsi en sélectionnant un nombre  $N$ , l'utilisateur peut créer une nouvelle "mesure" tous les  $N$  segments. Cette fonctionnalité

est décrite dans la section 3.3.2.

- 2 - La possibilité de décaler tous les index d'une même durée. Lorsque les participants de l'atelier utilisaient l'algorithme [59] pour segmenter l'enregistrement à la pulsation, ils souhaitaient décaler légèrement la position des pulses pour qu'elle tombe juste avant le début d'une note. Effectivement, si on n'effectue pas ce décalage la position des pulses détectées par l'algorithme [59] tombe souvent au milieu d'une note. En décalant cette position de quelques millisecondes avant la pulse on récupère l'attaque de la note.
- 3 - La disposition des éléments de l'interface graphique. Durant les ateliers plusieurs retours portant sur les onglets de Mono-Replay ont été faits. Suite à cela, un système de menu déroulant a été mis en place dans chaque onglet pour remplacer le système de pages déjà existant.

*Atelier 2* : Utilisation de Mono-Replay avec un gamepad DualAnalog (une séance). Ce deuxième atelier portait sur la prise en main du logiciel avec l'interface DualAnalog décrite en introduction de cette sous-section. Le but était de vérifier la bonne prise en main du logiciel avec le mapping présenté sur la figure 4.1. Durant cette séance, chaque mode de contrôle rythmique a été exploré pour déterminer celui qui semblait le plus facile à jouer collectivement de façon synchronisée. Le mode synchrone linéaire "Speed-freez" décrit dans la section 3.4.1.2 a été le plus facile à prendre en main en utilisant cette interface. Dans ce mode, le passage à l'index suivant est possible seulement si l'utilisateur actionne le bouton prévu pour cet effet. Si tout le monde presse le bouton au même moment alors tous les utilisateurs passent au segment suivant en même temps. La facilité de prise en main de cette interface a permis à tous les joueurs de jouer le même enregistrement de façon synchronisée, en une seule séance. *Atelier 3* : Création d'un projet Mono-Replay (une séance).

Durant le dernier atelier chaque participant devait créer son propre projet avec Mono-Replay. Il fallait suivre les instructions du document disponible annexe G pour importer un enregistrement, le découper et contrôler son déroulement

temporel à l'aide du logiciel. La majeure partie des retours faits durant cet atelier ont permis d'améliorer l'étape d'étiquetage. Les ateliers SMAC furent réalisés pendant le développement du logiciel. Les retours qui ont été faits ont donc été pris en compte dans sa dernière version. En parallèle de ces ateliers se sont déroulés des échanges plus approfondis avec la chercheuse Gabriela Patino Lakatos. Le but de ces échanges était d'élaborer plusieurs questionnaires visant à évaluer l'expérience des utilisateurs des logiciels Mono dont Mono-Replay fait partie. Ces questionnaires ont permis par la suite de conserver les expériences d'utilisateurs avec le logiciel.

#### 4.1.2 Animer un enregistrement multipiste du morceau *Superstition* par Stevie Wonder

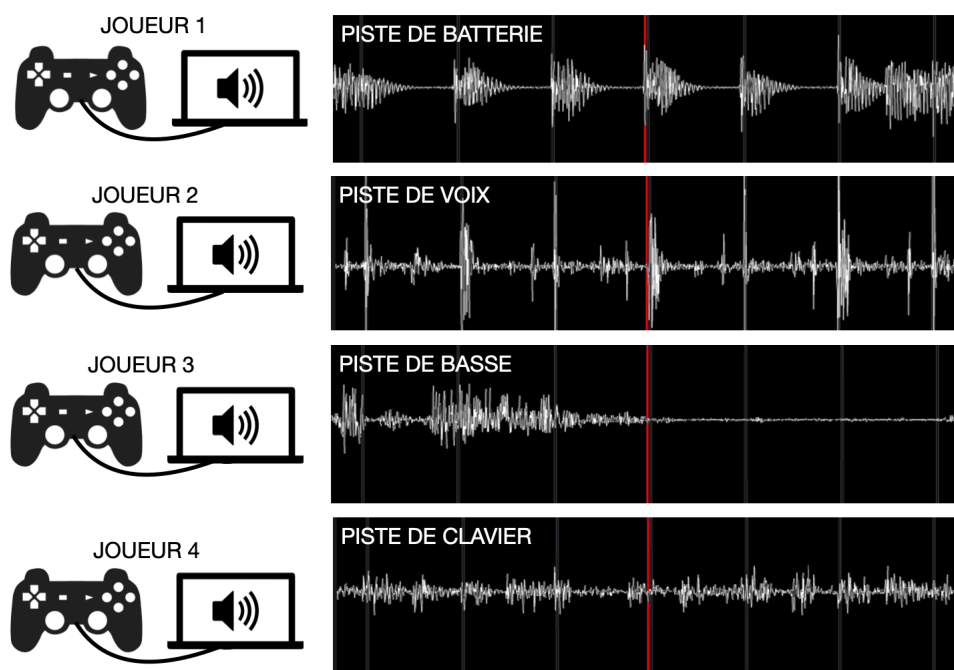


FIGURE 4.3 – 4 étudiants de Sorbonne Université ont utilisé Mono-Replay pour rejouer ensemble un enregistrement multipiste de *Superstition* de Stevie Wonder. Ils disposaient tous d'une manette de jeu pour contrôler la position et la vitesse de lecture d'une des pistes de l'enregistrement. Ainsi le joueur 1 contrôle la piste de batterie, le joueur 2 contrôle la piste de voix, le joueur 3 contrôle la piste de basse et le joueur 4 contrôle la piste de clavier.

Un groupe constitué de 4 étudiants de Sorbonne Université, dont moi-même, a pris en main Mono-Replay pour explorer ses possibilités dans le cas d'un contrôle avec la manette de jeu. Nous travaillons tous sur des sujets d'informatique musicale. Nous sommes donc habitués à manipuler des logiciels de transformation sonore ainsi que des interfaces gestuelles expressives. Chacun d'entre nous disposait d'une version compilée de l'application. Nous avons pour consigne de trouver plusieurs façons de rejouer ensemble un enregistrement multipiste de *Superstition* de Stevie Wonder, en le déroulant temps par temps selon une segmentation à la pulsation obtenue par la méthode [59].

La première séance a été une présentation du logiciel et de ses différentes fonctionnalités. La deuxième et la troisième séance furent dédiées à une exploration libre de tous les modes de contrôle rythmique pour déterminer lesquels semblaient les plus adaptés pour rejouer un enregistrement multipiste de façon synchronisée. La dernière séance a consisté en l'enregistrement de plusieurs performances compte tenu des observations faites dans les précédentes séances.

À la suite de ces ateliers, chaque participant a rempli un questionnaire conçu par Gabriela Patino-Lakatos visant à évaluer l'expérience des logiciels Mono dans le cadre d'un projet de création. Ce questionnaire est disponible en annexe I. Sur les 7 personnes ayant répondu à ce questionnaire, seules 3 ont utilisé Mono-Replay. Il est donc difficile d'extraire de ces résultats des conclusions valables.

Durant cet atelier nous avons observé qu'il était plus facile de synchroniser les participants au même rythme de déroulement temporel en suivant un stimulus sonore externe comme un métronome. On parvenait aussi facilement à les synchroniser en définissant un participant comme "leader" dont il fallait suivre le rythme de déroulement. L'interface graphique et l'affichage de la forme d'onde avaient plutôt tendance à perturber le jeu collectif. En faisant abstraction de l'affichage et en se concentrant uniquement sur le son produit, il était plus facile de se synchroniser.

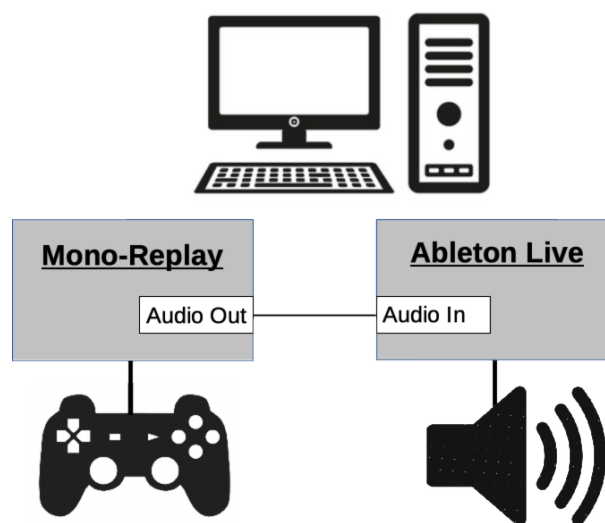


FIGURE 4.4 – Schéma du dispositif utilisé par l’artiste Pehoz pour contrôler Mono-Replay à l’aide d’une manette de jeu et intégrer la performance dans un arrangement préconçu

#### 4.1.3 Création du morceau *Bar Mono-Replay* par Pehoz

Une évaluation qualitative est présentée dans le chapitre 5 de ce manuscrit. Pour la mener nous avons demandé à trois musiciens de prendre en main Mono-Replay pour réaliser une création. Nous avons ensuite mené des entretiens semi-directifs pour récupérer suffisamment d’informations afin de conclure sur l’apport créatif de l’outil.

Parmi ces musiciens figure le Beatmaker Pehoz. Compositeur de hip-hop au sein du label Jeune à Jamais. Passionné par le sampling, discipline décrite dans la section 1.2.1, il manipule des algorithmes de contrôle temporel d’enregistrement depuis plus de 10 ans pour créer ses compositions. Il est habitué des environnements logiciels pour la création sonore.

Pehoz a utilisé Mono-Replay selon le dispositif schématisé figure 4.4 pour découper, étirer et manipuler un enregistrement de *Vocal soul train* par The Three Degrees et de *My baby just cares for me* par Nina Simone. Ces manipulations sont enregistrées dans Ableton Live où se joue un arrangement déjà préconçu.

Le dispositif représenté figure 4.4 est constitué d’une manette de jeu DualAnalog Thrustmaster branchée à un ordinateur qui fonctionne sous Windows 8.



Cette manette permet de contrôler Mono-Replay selon le mapping représenté figure 4.1. À l'aide du logiciel Jack Audio Connection Kit, la sortie audio de Mono-Replay est reliée à l'entrée audio du logiciel Ableton Live. Des pistes d'arrangements sont jouées dans Ableton Live et le son généré par Mono-Replay est directement enregistré dans le logiciel. Un lien vers le morceau *Bar Mono-Replay* créé à l'aide de ce dispositif est disponible en bas de page suivante<sup>1</sup>. Un entretien détaillé avec l'artiste est disponible annexe E. Cet entretien a eu lieu après la création du morceau *BAR MONOREPLAY* et porte sur l'expérience de l'artiste avec ce dispositif. Son analyse est disponible dans le chapitre 5 de ce manuscrit.



FIGURE 4.5 – Le musicien Pehoz dans son home studio utilisant Mono-Replay pour découper, étirer et manipuler un enregistrement de Nina Simone et de The Three Degrees.



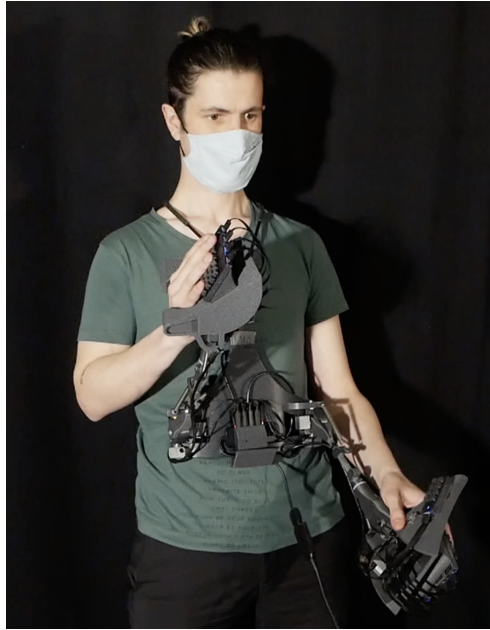


FIGURE 4.6 – Prise en main du *Méta-instrument 4* pour animer le *Trio n°2, op. 100* de Franz Schubert en battant la pulsation

## 4.2 Contrôler Mono-Replay à l'aide du Méta-Instrument 4

### 4.2.1 Le Méta-Instrument 4

Le Méta-Instrument représenté figure 4.6 est un instrument numérique conçu et développé par Serge de Laubier depuis plus de trente ans. Né du double objectif de contrôler le déplacement et le positionnement des sons dans l'espace et de permettre l'interprétation de la musique électroacoustique en direct [14], l'instrument offre la possibilité de développer un très grand nombre de mappings grâce à la quantité importante de données mesurées. Combinant treize interfaces, pilotant 92 capteurs et faisant l'acquisition de 184 variables, le Méta-Instrument 4 est un outil complexe à prendre en main. Cette complexité en fait un outil adapté pour aborder la problématique du contrôle gestuel pour les musiques enregistrées. Si l'on se rapporte à la théorie

---

1. Le morceau réalisé par le musicien Pehoz à l'aide de Mono-Replay est disponible par le lien suivant : [https://drive.google.com/file/d/1hZl27uR\\_eitd\\_tWzsvT\\_G6kILBwk\\_VAo/view?usp=sharing](https://drive.google.com/file/d/1hZl27uR_eitd_tWzsvT_G6kILBwk_VAo/view?usp=sharing)

de Denis Smalley [54] sur le substitut gestuel, abordé dans la section 1.2, il existerait autant de gestes représentés par l'esprit que de sons existants. Si on considère que ces gestes "imaginés" sont les plus adaptés pour contrôler le déroulement temporel d'un enregistrement, le Méta-Instrument permettrait d'en réaliser une grande quantité.

Chaque bras du MI4 (Méta-Instrument 4), est équipé de 40 capteurs de

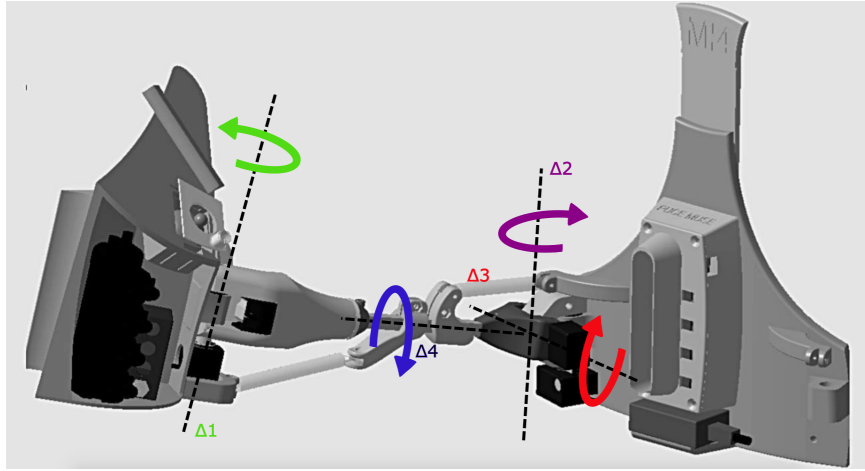


FIGURE 4.7 – Les différents axes de rotation d'un des bras du Méta-Instrument 4

pression, un joystick et 4 potentiomètres. Ces potentiomètres permettent de récupérer le mouvement de chaque bras articulé selon les axes de rotation présentés figure 4.7. Dans les mappings que nous présentons à la suite de ce paragraphe, le mouvement des potentiomètres sert à contrôler l'évolution des paramètres spectraux de SuperVp. Le signal émis par ces potentiomètres est numérisé par des convertisseurs embarqués sur l'instrument. Ces données sont ensuite converties en signal MIDI codé sur 14 bits. Les valeurs minimum et maximum de butée du potentiomètre deviennent des données MIDI variant de 0 à 1683. On normalise ensuite ces valeurs pour qu'elles ne varient qu'entre 0 et 1. On peut alors contrôler de façon continue nos paramètres spectraux selon l'équation suivante :

$$\forall t \in \mathbb{N}, \quad \phi(t) = \Delta \cdot \alpha(t) + S_{min} \quad (4.2)$$

Avec  $\Delta = S_{max} - S_{min}$ ,  $S_{max}$  la valeur maximale que peut prendre le paramètre de transformation spectrale,  $S_{min}$  la valeur minimale et  $\alpha(t)$  la valeur

de contrôle du potentiomètre comprise dans l'intervalle  $[0;1]$ .

On peut, par exemple, contrôler l'évolution de la transposition de hauteur selon l'axe de rotation  $\delta_2$  représenté sur la figure 4.7. Si on choisit  $S_{min}=-1200$  cents et  $S_{max}=1200$  cents, la position haute du bras selon cet axe règle le paramètre à 1200 cents et la position basse du bras règle le paramètre à -1200 cents. Pour obtenir de plus petites variations et recréer l'effet de vibrato, on peut choisir  $S_{min}=-100$  cents et  $S_{max}=100$  cents.

Évidemment tous les paramètres de contrôle spectral de SuperVp sont contrôlables selon ce principe. La sensibilité du Méta-Instrument 4 est très utile pour régler précisément les valeurs de transformation spectrale. Néanmoins, il arrive souvent que l'on veuille réduire la sensibilité du capteur autour d'une valeur précise  $\Theta$  afin de pouvoir l'atteindre précisément. On peut paramétrer l'évolution de nos paramètres de transformation spectrale de la façon suivante :  $\forall t \in \mathbb{N}$ ,

$$\phi(t) = \begin{cases} (S(t) - \Theta_{min}) \cdot \frac{S_{min} - \Theta}{S_{min} - \Theta_{min}} + \Theta & \text{si } S(t) \in (S_{min}, \Theta_{min}) \\ \Theta & \text{si } S(t) \in (\Theta_{min}, \Theta_{max}) \\ (S(t) - \Theta_{max}) \cdot \frac{S_{max} - \Theta}{S_{max} - \Theta_{max}} + \Theta & \text{si } S(t) \in (\Theta_{max}, S_{max}) \end{cases} \quad (4.3)$$

avec  $S_{min}$  et  $S_{max}$  les valeurs minimum et maximum que peut prendre le paramètre spectral que l'on souhaite contrôler,  $\Theta_{min}$  et  $\Theta_{max}$  les valeurs seuils qui encadrent la valeur  $\Theta$  que l'on souhaite "viser",  $S(t)$  la fonction affine décrite dans le mode précédent de telle sorte que :  $S(t) = \Delta \cdot \alpha(t) + S_{min}$ , avec  $\Delta = S_{max} - S_{min}$  et  $\alpha(t)$  les données gestuelles captées en entrée du système, variant sur l'intervalle  $[0;1]$ . Dans la suite de ce paragraphe nous présentons plusieurs scénarios faisant intervenir le MI4 et Mono-Replay. Le but de chaque scénario est de rejouer l'enregistrement dans sa globalité mais de façon expressive.

#### **4.2.2 Animer un enregistrement du *Trio n°2* de Franz Schubert par le Beaux Arts Trio en battant une pulsation**

Au début de ce travail, de nombreuses pistes concernant les modes de contrôles qui seront intégrés à Mono-Replay ont été évoquées. De nombreuses

vidéos de démonstration pour des systèmes déjà existants ont été visionnées et certains gestes de contrôle ont été plus marquants que d'autres. C'est le cas de la vidéo de présentation des interfaces modulaires MO développé par l'IRCAM<sup>2</sup>. Dans cette vidéo, l'effet du geste sur le système est parfaitement compris du point de vue du spectateur : en effectuant un mouvement périodique, on contrôle l'avancement de la lecture d'un enregistrement. L'expressivité du geste évoque beaucoup celle du chef d'orchestre ce qui semble rendre l'expérience encore plus intuitive.

Réaliser ce geste avec le MI4 et Mono-Replay nous a semblé être une bonne démonstration des possibilités de notre système. En couplant le MI4 aux différents contrôles rythmiques de Mono-Replay, on obtient un système expressif capable de réaliser de véritables gestes musicaux. Si on observe ce système selon les fonctions définissant un geste instrumental selon Claude Cadoz [9] on peut le décomposer de la façon suivante :

- *La fonction ergotique* (énergie du geste) serait occupée par la vitesse d'exécution du geste de battue de tempo.
- *La fonction épistémique* qui définit le rapport haptique et sensoriel du musicien avec le corps de l'instrument serait ici exercée par la main de l'utilisateur tenant le MI4.
- *La fonction sémiotique* par laquelle le musicien transmet une intention au spectateur serait ici comparable à l'intention transcrite par un chef d'orchestre lors d'une performance.

Pour réaliser ce scénario nous avons choisi un enregistrement de musique classique pulsée dont le tempo n'est pas trop rapide : Le *Trio n°2, op. 100* de Franz Schubert. Un lien vers une vidéo montrant ce scénario est disponible en bas de page<sup>3</sup>. La première étape consiste à étiqueter cet enregistrement avec Mono-Replay. C'est ce que nous décrivons ci-dessous sous la forme d'un petit tutoriel.

---

2. La vidéo de démonstration du capteur MO a beaucoup inspiré les scénarios d'utilisation décrits dans ce paragraphe : [https://www.youtube.com/watch?v=Uhps\\_U2E90M&t=1s](https://www.youtube.com/watch?v=Uhps_U2E90M&t=1s)

3. Cette vidéo montre un scénario d'interaction dans lequel un utilisateur contrôle la position de lecture d'un enregistrement du *Trio pour corde et piano* de Franz Schubert interprété par le Beaux Arts Trio, en battant la pulsation : <https://youtu.be/lyIwBcB1M3I>

#### 4.2.2.1 Étiquetage

Pour pouvoir contrôler le déroulement d'un enregistrement en battant une pulsation avec Mono-Replay, il faut, dans un premier temps, réaliser l'étiquetage. Pour cela on glisse le fichier audio dans le logiciel et on ouvre l'éditeur de segment qui est représenté figure 4.8.

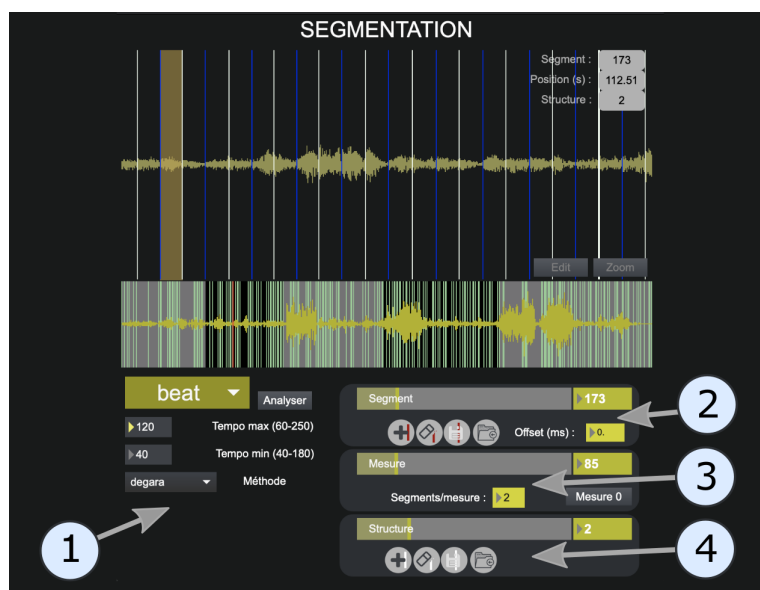


FIGURE 4.8 – Grâce à l'éditeur de segment de Mono-Replay on peut : 1) sélectionner un algorithme d'analyse; 2) effacer - ajouter - décaler un segment; 3) définir un nombre de segments par mesure et définir la position de la première mesure; 4) effacer - ajouter - enregistrer une structure. On peut aussi naviguer dans l'enregistrement selon les trois niveaux de segmentation (segment, mesure, structure).

Une fois l'éditeur de segment ouvert, on peut suivre les étapes représentées dans la figure 4.8 :

- *Étape 1* : On sélectionne *beat* dans le menu déroulant pour choisir la méthode de segmentation par suivi de pulsation. Il arrive souvent que les méthodes de suivi de pulsation proposent des découpages selon des valeurs de tempo multiples de la valeur réelle, si des subdivisions rythmiques sont sur-représentées. En réglant une valeur maximum à ne pas dépasser, on s'assure que la détection ne se fera pas pour une battue à la croche, par exemple. Dans notre cas, le tempo à la noire étant environ 80, on règle le paramètre *tempo max* à 120 pour éviter qu'il

ne détecte un tempo à la croche. La nature des transitoires contenus dans cet enregistrement étant bien détectée par la méthode de flux spectral, on choisit donc la méthode Degara [20] pour estimer la position des pulsations. Si les transitoires étaient de nature composites, nous aurions choisi la méthode "Multifeature" [59] qui, elle, détermine plusieurs fonctions de détection pulse selon 8 algorithmes de détection de transitoire. Une estimation de la similarité des résultats de chaque fonction permet de sélectionner le meilleur candidat pour la détection de transitoire. Une fois la fonction paramétrée, on clique sur "Analyser" pour lancer l'algorithme.

- *Étape 2* : On utilise les deux boutons d'édition de segments visibles sur la figure 4.8 pour ajouter ou supprimer les segments résultant de l'analyse. Il arrive qu'il faille faire ce type de correction pour les brusques évolutions de tempo d'un enregistrement. L'algorithme [59] ne détecte pas bien ces phénomènes. Une fois la détection de pulse effectuée, on peut décaler la position de toutes les ancrs temporelles du fichier d'étiquetage à l'aide du bouton "offset (ms)". Il arrive souvent que les pulsations soient placées au milieu d'une note. Dans ce cas on peut les décaler de quelques millisecondes pour avoir l'intégralité de la note. Une fois que cette étape de correction est effectuée, il suffit de cliquer sur le bouton "disquette" pour enregistrer notre fichier d'étiquetage dans le répertoire du projet. Ainsi le fichier d'étiquetage sera directement accessible via l'interface de Mono-Replay.
- *Étape 3* : Une fois notre premier niveau de segmentation effectué, on peut réaliser un deuxième type de segmentation dit "à la mesure". Le bouton "Segments/mesures" permet de définir un nombre de segments par mesure. Pour l'enregistrement du *Trio n°2, op. 100* de Franz Schubert on choisit deux segments par mesure. Ainsi on se conforme à la signature rythmique du morceau qui est inscrite sur la partition. Le bouton "Mesure 0" permet de définir la position du début de la première mesure du morceau. Il arrive parfois que des morceaux commencent au milieu d'une mesure, auquel cas il faut indiquer au système que la

première mesure débute après les premières notes du morceau.

- *Étape 4* : La dernière étape de segmentation est l'étiquetage dit "à la structure". Les boutons d'édition permettent exactement les mêmes interactions que pour l'édition de segment. Dans le cas de l'enregistrement du *Trio n°2* de Franz Schubert on place des ancrs temporelles au début de chaque thème.

Cette étape d'étiquetage nous permet de préparer toutes les données nécessaires pour utiliser l'ensemble des modes de contrôle rythmique disponibles dans Mono-Replay.

#### 4.2.2.2 Mapping

Une fois l'étiquetage de l'enregistrement effectué, on relie les valeurs des potentiomètres du MI4 aux contrôles de Mono-Replay de la façon suivante :

- *Potentiomètre 1* : Lorsque le sens de variation du potentiomètre placé sur l'axe de rotation  $\Delta 3$  représenté figure 4.7 change, un point de contrôle est envoyé à tous les modes "Tap-Tempo" décrits dans la section 3. La durée entre deux points de contrôle augmente ou diminue la vitesse de lecture. Plus le tempo battu sera rapide, plus la vitesse augmentera rapidement. Dans la vidéo dont le lien est disponible en bas de page<sup>4</sup> le mode "Tap-tempo" utilisé est le mode "parabolic" décrit dans la section 3. À chaque nouvelle pulsation battue le système calcule une fonction parabolique qui définit le profil d'évolution de la vitesse.
- *Potentiomètre 2* : Sur un autre bras du MI4, le potentiomètre placé sur l'axe de rotation  $\Delta 3$  représenté figure 4.7 définit la valeur de transposition de hauteur selon l'équation 4.3 présentée plus haut.
- *Potentiomètre 3* : Le potentiomètre placé sur l'axe de rotation  $\Delta 4$  permet de faire varier la valeur de transposition spectrale : le mouvement du poignet fait varier la transposition de l'enveloppe spectrale selon

---

4. Le début de cette vidéo montre un scénario d'interaction dans lequel un utilisateur contrôle la position de lecture d'un enregistrement du *Trio pour corde et piano* de Franz Schubert en battant la pulsation : <https://youtu.be/lyIwBcB1M3I>

l'équation 4.3.

#### 4.2.3 Animer *Accidents/Harmoniques* de Bernard Parmegiani en contrôlant de façon continue la position de lecture

Suite à un entretien plusieurs années auparavant avec Bernard Parmegiani, Serge de Laubier a voulu explorer de nouvelles possibilités pour contrôler rythmiquement les pièces de ce compositeur. Lors de cet entretien Serge de Laubier a montré au compositeur différents mappings gestuels réalisés avec un joystick pour contrôler des transformations sonores. Le compositeur s'est alors montré enthousiaste pour utiliser ces gestes dans sa création. C'est pour cette raison que nous avons réalisé un scénario permettant de contrôler de façon continue la position de lecture de la pièce enregistrée *Accidents/Harmoniques* à l'aide d'un joystick.

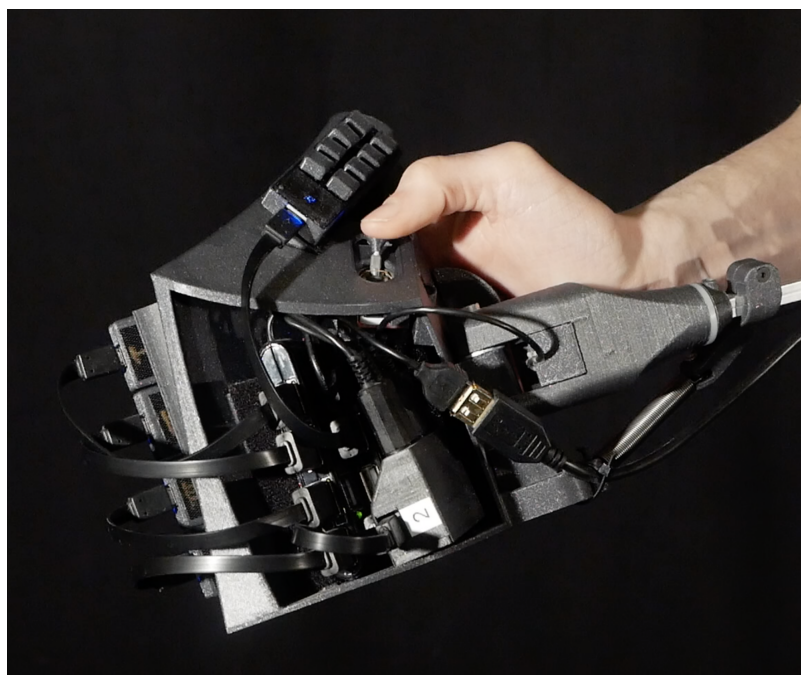


FIGURE 4.9 – Le joystick droit du MI4 (le pouce de l'utilisateur est posé dessus) est ici utilisé pour contrôler de façon continue la position de lecture de la pièce de musique électroacoustique *Accidents/Harmoniques* composée par Bernard Parmegiani.

Le joystick que nous utilisons est celui présent sur le bras droit du MI4.



Sur la figure 4.9 le pouce de l'utilisateur est placé sur ce joystick. Le signal envoyé par le joystick est ensuite numérisé par un convertisseur Méta-Digit présent sur l'instrument.

#### 4.2.3.1 Étiquetage

Pour étiqueter cet enregistrement, nous utilisons l'algorithme de détection de transitoire [6]. Depuis l'éditeur de segment représenté figure 3.3, on règle l'écart minimum entre deux transitoires consécutifs à 150 ms, le seuil de détection à 16 et le rapport de seuil à 0,01. Ces réglages nous permettent de limiter le nombre de transitoires détectés par erreur. Une fois les résultats de l'analyse obtenus, on supprime, corrige et ajoute les index manquants selon la partition du compositeur présente dans [39]. Dans la figure 4.10 on voit un extrait de cette partition en haut avec les transitoires que l'on souhaite détecter en rouge et la forme d'onde avec les transitoires correspondants en bas.

#### 4.2.3.2 Mapping

Le mode de contrôle temporel utilisé ici est celui par contrôle continu de position décrit dans la section 2.3.2. L'évolution de l'indice temporel  $\tau(t)$  selon ce mode s'exprime mathématiquement de la façon suivante :  $\forall i \in \mathbb{N}, \forall t \in \mathbb{R}$ ,

$$\tau(t) = (\lambda_{i+1} - \lambda_i)p(t) \quad (4.4)$$

Avec  $\tau(t)$  l'indice temporel qui correspond à la position à jouer dans l'enregistrement,  $\lambda_i$  le marqueur temporel à l'indice  $i$  et  $p(t)$  l'angle d'ouverture du joystick. La liste des marqueurs temporels  $\lambda$  est déterminée selon la méthode décrite dans la section 4.2.3.1. L'angle d'ouverture du joystick  $p(t)$  est converti en *tour* et ses valeurs varient dans l'intervalle  $[0;1]$ . Dans le cas où l'angle d'ouverture évolue de façon positive, le mapping est conçu pour passer automatiquement au segment suivant lorsque  $p(t) = 1$ . Dans le cas où l'angle d'ouverture évolue de façon négative, le mapping est conçu pour passer automatiquement au segment suivant lorsque  $p(t) = 1$ .

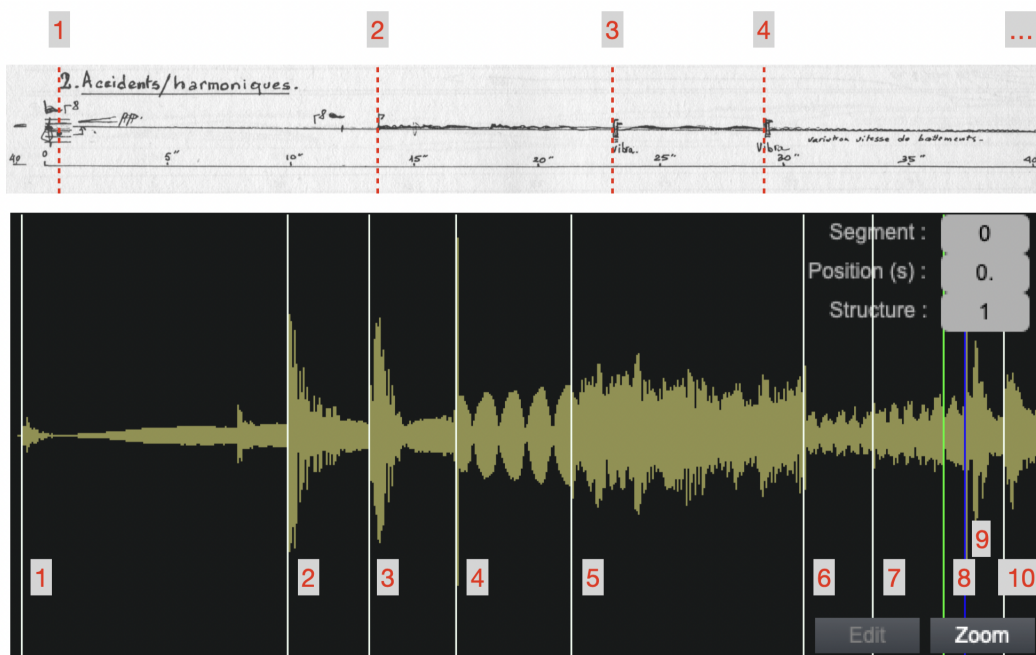


FIGURE 4.10 – En haut le début de la partition de *Accident/Harmoniques* avec les index, en rouge, que l'on souhaite récupérer dans Mono-Replay. En bas la forme d'onde du début de l'enregistrement de *Accidents/Harmoniques* avec les index correspondant en blanc. Les index ont été trouvés grâce à l'algorithme de détection de transitoire d'attaque Superflux [6]

Une vidéo dont le lien est en bas de page<sup>5</sup> montre un extrait de la relecture de *Accidents/Harmoniques* selon ce scénario. Dans cette vidéo chaque tour de joystick permet de parcourir un segment de l'enregistrement obtenu selon la méthode décrite précédemment.

### 4.3 Animer deux pièces de musiques électroniques composées par Nicolas Rivière

Le logiciel Mono-Replay a été utilisé par le compositeur Nicolas Rivière pour créer et rejouer deux de ses créations. Tout comme Pehoz, un entretien semi-directif a été mené avec Nicolas suite à ce travail afin de récupérer l'ex-

5. à 1 min 55 l'utilisateur contrôle la position de lecture de la pièce *Accidents/Harmoniques* de Bernard Parmegiani à l'aide du joystick droit du Méta-Instrument 4 : <https://youtu.be/1yIwBcB1M3I>

périence subjective de l'artiste avec le système et ainsi évaluer l'apport créatif du logiciel. Une analyse de cet entretien est disponible dans la section 5 de ce manuscrit.

#### **4.3.1 *SUPERNOME DIVA 2 - Contemplation du vide***

*SUPERNOME DIVA 2 - Contemplation du vide* est le nom du premier morceau composé par Nicolas Rivière et spécialement conçu pour une utilisation dans Mono-Replay. Il s'agit d'une pièce en 3 mouvements mélangeant plusieurs sons électroacoustiques préenregistrés qu'il manipule à l'aide Mono-Replay.

Le premier et le dernier mouvement sont joués à l'aide du mode "Scrub" disponible dans le logiciel. Ces mouvements n'ont pas de pulsations et sont composés essentiellement de nappes de synthétiseurs. Un index indique le début et la fin de ces deux mouvements. Le mouvement central est une valse. Plusieurs index indiquent la position de chaque pulsation de ce mouvement. Les pulsations de ce mouvement ont été détectées à l'aide de l'algorithme Multi-feature Beat Tracking [59] intégré dans Mono-Replay. Il se sert ensuite de cette indexation pour modifier la position de lecture de l'enregistrement pendant la performance. Des séquences préenregistrées à la fin et au début de la pièce sont lues parallèlement avec le logiciel Méta-DJ. Le dispositif qu'il utilise pour manipuler Mono-Replay est décrit dans la sous-section 4.3.3.

Une restitution publique de cette pièce a eu lieu à Paris au sein de l'Institut Jean Le Rond d'Alembert le 20 novembre 2021. Une vidéo de cette restitution est disponible en bas de page<sup>6</sup>

#### **4.3.2 *SUPERNOME DIVA 2.5 - Anatomie du vide***

*SUPERNOME DIVA 2.5 - Anatomie du vide* est le nom du deuxième morceau conçu pour être rejoué avec le logiciel Mono-Replay. Ce morceau a été composé cette fois-ci par Nicolas Rivière et Elias Cheyroux. Pour rejouer la pièce Nicolas contrôle le logiciel Mono-Replay et Elias joue du synthéti-

---

6. Le compositeur Nicolas Rivière a joué sa pièce *SUPERNOME DIVA 2 - Contemplation du vide* à Paris au sein de l'Institut Jean Le Rond d'Alembert le 20 novembre 2021. Une vidéo montrant ce concert est disponible via le lien suivant : <https://youtu.be/15k080rCriI>



FIGURE 4.11 – Niolas Rivière manipulant le logiciel Mono-Replay lors du concert *SUPER-NOME DIVA 2 : Contemplation du vide* ayant eu lieu à Paris au sein de l'Institut Jean Le Rond d'Alembert le 10 novembre 2021

seur analogique prophet 5 et du sampler Akai MPC. La pièce est composée



FIGURE 4.12 – Niolas Rivière et Elias Cheyroux lors du concert *SUPER-NOME DIVA 2.5 : Anatomie du vide* ayant eu lieu à Paris dans la tour Zamanksy

de 7 mouvements durant lesquels les deux musiciens improvisent autour de thèmes préalablement préparés. La partie jouée dans Mono-Replay est un en-

enregistrement d'une vingtaine de minutes qui est transformé temporellement et spectralement à l'aide du logiciel. Dans cette partie une indexation est faite sur l'enregistrement pour indiquer le début de chaque mouvement. Ainsi Nicolas peut se positionner au début de ceux-ci à n'importe quel moment de la performance. Les mouvements 4 et 7 sont composés de 4 sous-segments indiquant le début et la fin de boucles rythmiques. Ces différentes indexations ont été conçues dans le logiciel à l'aide des fonctions d'indexations manuelles.

Pendant la performance Nicolas Rivière contrôle la vitesse et la position de lecture de son enregistrement à l'aide des modes "Speed-direct" et "Scrub" de Mono-Replay. Le dispositif utilisé pour contrôler le logiciel est le même que celui utilisé pour la pièce *SUPERNOME DIVA 2 - Contemplation du vide*. Ce dispositif est présenté dans la sous-section 4.3.3.

Une représentation publique a eu lieu au 24<sup>ème</sup> étage de la tour Zamansky le 4 février 2022. Une photo prise lors de cette représentation est disponible figure 4.12

### 4.3.3 Dispositif et mappings

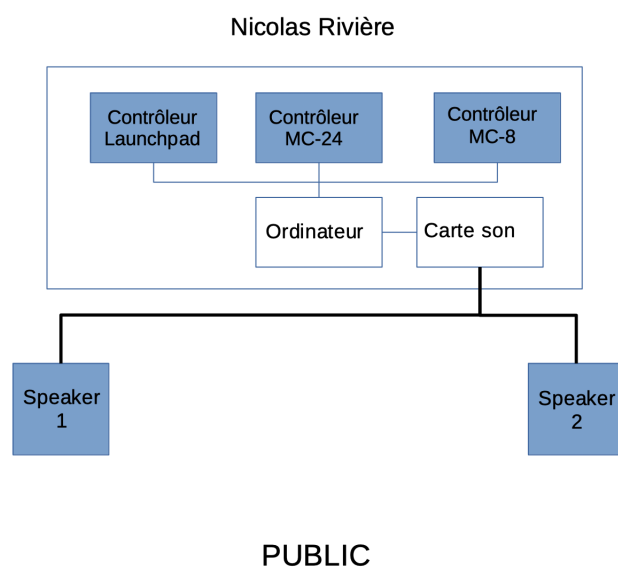


FIGURE 4.13 – Dispositif utilisé par Nicolas Rivière pour jouer sa pièce *SUPERNOME DIVA 2 - Contemplation du vide* à l'aide de Mono-Replay

Le dispositif utilisé pour rejouer ces deux pièces avec Mono-Replay est

visible figure 4.13. Il est composé de trois interfaces gestuelles :

- Un contrôleur MIDI MC-8 composé de 8 faders.
- Un contrôleur MIDI MC-24 composé de 24 potentiomètres.
- Un contrôleur MIDI Novation LaunchPad Mini composé de 64 boutons.

Pour contrôler la vitesse de lecture Nicolas pilote le mode "Speed-direct" à l'aide du potentiomètre 1 visible sur la figure 4.15. Pour contrôler la position de lecture il utilise le mode "Scrub" qu'il pilote à l'aide du potentiomètre 2 visible sur la figure 4.15. Sur cette même figure les potentiomètres 4 et 5 sont utilisés pour contrôler la fréquence de modulation de l'enveloppe d'effet 1 et 2 intégré dans Mono-Replay.

Les transformations spectrales sont pilotées à l'aide des interfaces gestuelles visibles figure 4.14 et 4.16. Les mappings utilisés sont détaillés dans la légende de ces figures.

Les trois interfaces de ce dispositif permettent de contrôler le logiciel Mono-Replay via le protocole OSC. Les contrôles de ces interfaces sont reliés aux contrôles du logiciel de la manière suivante. Les noms des variables visibles dans les figures 4.14, 4.15 et 4.16 font référence aux variables du dictionnaire OSC disponible en annexe H :

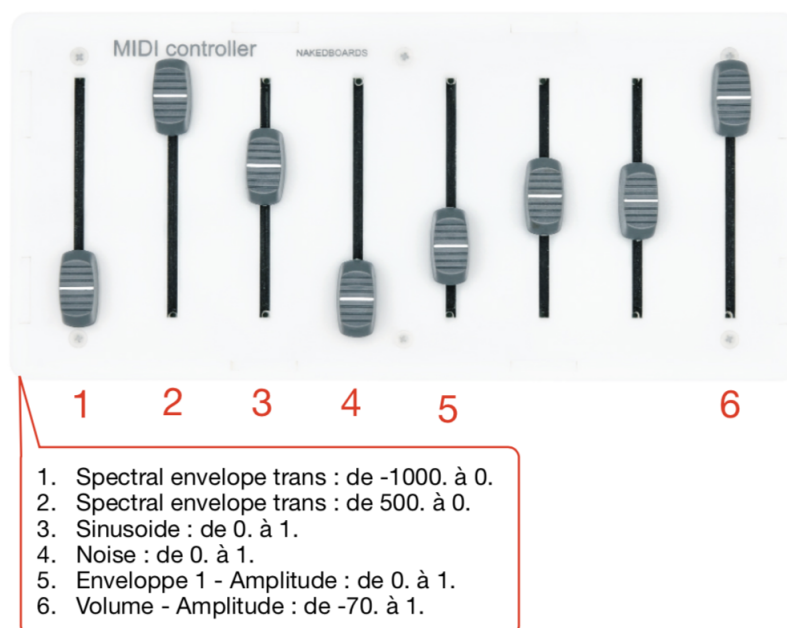


FIGURE 4.14 – Mapping des faders de l'interface MC-8 pour contrôler les différentes fonctions de Mono-Replay pour jouer la pièce *SUPERNOME DIVA 2 - Contemplation du vide*. 1- transposition de l'enveloppe spectrale de -1000 cents à 0 cent ; 2- transposition de l'enveloppe spectrale de 500 cents à 0 cent ; 3- mixage de la partie sinusoïdale du son de 0 à 1 ; 4- mixage de la partie inharmonique du son de 0 à 1 ; 5 - réglage de l'amplitude de l'enveloppe de 0 à 1 ; 6- réglage du volume de -70 db à 1 db.



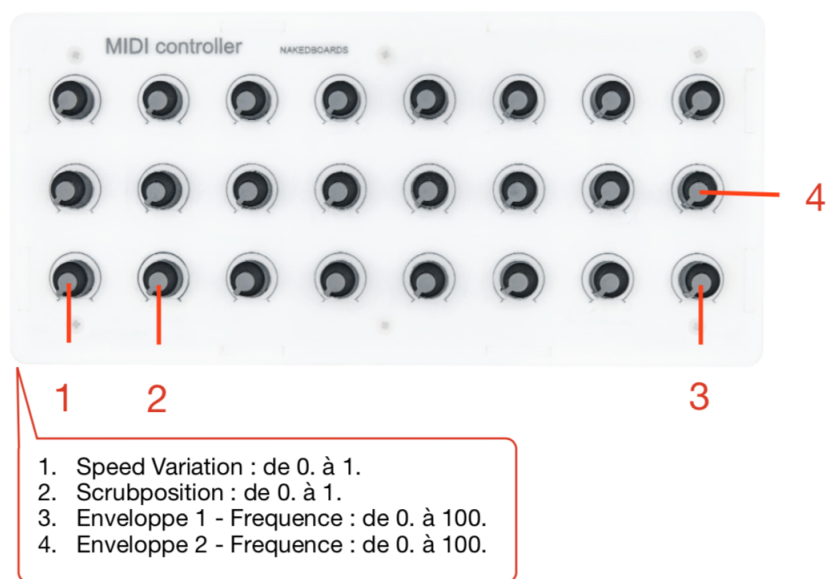


FIGURE 4.15 – Mapping de l'interface MC-24 composée de potentiomètres pour contrôler les différentes fonctions de Mono-Replay pour jouer la pièce *SUPERNOME DIVA 2 - Contemplation du vide*. 1- contrôle de la vitesse de lecture du mode "speed-direct" entre 0 et 1 ; 2- contrôle de la position de lecture du mode "scrub" entre 0 et 1 ; 3 - contrôle de la fréquence de l'enveloppe 1 entre 0 et 100 Hz ; 4 - contrôle de la fréquence de l'enveloppe 2 entre 0 et 100 Hz.



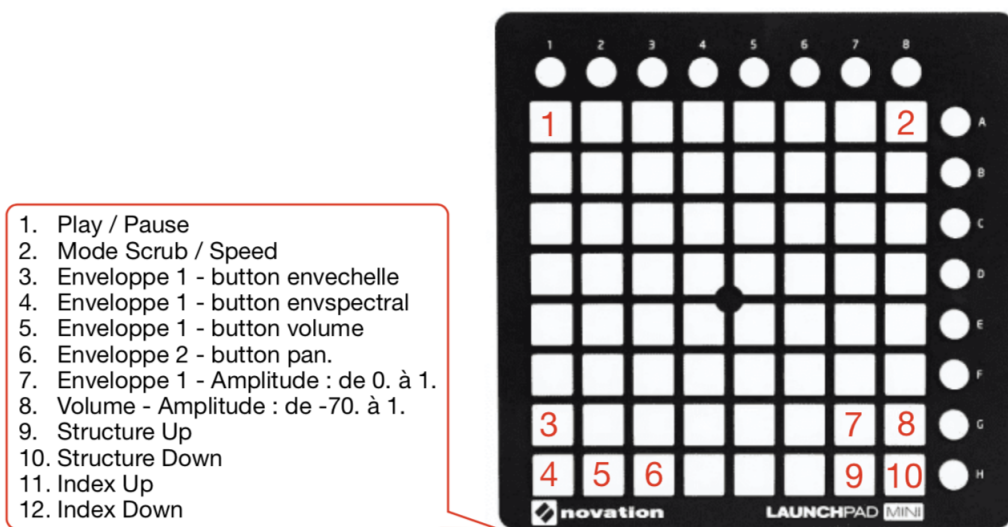


FIGURE 4.16 – Mapping de l'interface Launchpad Mini composée de boutons pour contrôler les différentes fonctions de Mono-Replay. 1 - déclenchement de la lecture et de l'arrêt de l'enregistrement ; 2- passage du mode "scrub" au mode "speed-direct" ; de 3 à 6, activation des paramètres contrôlés par l'enveloppe ; 7- permet de régler la valeur d'amplitude de l'enveloppe 1 entre 0 et 1 ; 8- permet de régler la valeur de volume entre - 70 et 0 db ; 9- permet de passer à la structure supérieure ; 10- permet de passer à la structure inférieure ; 11- permet de passer au segment supérieur ; 12- permet de passer au segment inférieur



## Chapitre 5

# Élaboration et analyse d'entretiens semi-directifs visant à évaluer qualitativement le logiciel Mono-Replay

### Sommaire

---

<b>5.1</b>	<b>Présentation des sujets . . . . .</b>	<b>149</b>
5.1.1	Nicolas Rivière . . . . .	149
5.1.2	Paul Denis Kingue-Manjole (Pehoz) . . . . .	150
5.1.3	Basile Chassaing . . . . .	151
<b>5.2</b>	<b>L'entretien exploratoire . . . . .</b>	<b>151</b>
<b>5.3</b>	<b>Élaboration du guide d'entretien . . . . .</b>	<b>155</b>
<b>5.4</b>	<b>À propos du contrôle temporel . . . . .</b>	<b>158</b>
<b>5.5</b>	<b>Au sujet du ressenti global de l'expérience . . . . .</b>	<b>159</b>
<b>5.6</b>	<b>À propos des interfaces gestuelles . . . . .</b>	<b>164</b>
<b>5.7</b>	<b>À propos de l'interface graphique . . . . .</b>	<b>166</b>
<b>5.8</b>	<b>Les perspectives de Mono-Replay selon les musiciens . . . .</b>	<b>168</b>
<b>5.9</b>	<b>Synthèse . . . . .</b>	<b>170</b>

---

Pour évaluer les possibilités créatives du système présenté ici, trois entretiens semi-directifs ont été menés. Ces entretiens prennent la forme d'échanges



sur des questions ouvertes entre un interviewer, moi-même, et trois musiciens ayant utilisé le logiciel pour leur création. Les questions abordent plusieurs aspects du logiciel et sont présentées dans le tableau à la fin de la sous-section 5.3. Les réponses de chaque musicien aux questions figurant sur le guide d'entretien permettent d'accéder à l'expérience subjective de chacun d'entre eux avec le système. Ces réponses sont analysées et servent à conclure sur l'apport créatif de l'outil comparé aux outils habituellement utilisés par ces musiciens. L'évaluation ici est qualitative. En opposition à une évaluation statistique il n'est pas nécessaire d'avoir un nombre important de sujets pour conclure sur les résultats obtenus.

Dans le domaine des sciences humaines, la méthode de l'entretien est fréquemment utilisée. Elle permet de récupérer plusieurs points de vue sur une expérience, un sujet ou un fait pour comprendre comment celui-ci est perçu. Dans le domaine des interactions homme machine les entretiens sont des méthodes courantes pour évaluer certains aspects de ceux-ci. Ils ont lieu après l'utilisation du logiciel et ont pour but de récupérer suffisamment d'informations pour comprendre comment l'interaction a été vécue par l'utilisateur.

Dans notre cas nous souhaitons accéder à l'expérience subjective d'artistes utilisant Mono-Replay pour une création musicale. Le but principal ici est de comprendre la place de cet outil dans le résultat final qu'ils ont produit. La nature ouverte des questions permet au musicien d'apporter plus de détails sur la réponse qu'il donne. On peut ainsi mieux comprendre comment la singularité de chaque artiste a joué un rôle dans l'utilisation du logiciel. En récupérant des informations sur leurs rapports au contrôle temporel, leurs

choix artistiques, leurs influences, leurs expériences en tant qu'utilisateurs de logiciel pour la création, nous pourrions évaluer le potentiel créatif de Mono-Replay.

Ce travail d'évaluation qualitative a été réalisé avec l'aide de Gabriela Patino Lakatos qui est spécialisée dans les sciences de l'éducation et leur observation par des outils numériques pour la création [45] [46]. Elle contribue grandement depuis 2015 à l'évaluation des différents projets pédagogiques de PUCE MUSE [44] [43] [47]. Son aide a été précieuse dans l'élaboration de ce protocole d'évaluation.

## **5.1 Présentation des sujets**

Une commande a été demandée à chaque musicien interrogé dans le cadre de cette évaluation. Ils devaient tous créer une oeuvre en utilisant le logiciel Mono-Replay. Le choix des musiciens s'est fait compte tenu de la singularité de leur pratique. Bien qu'ils se servent tous d'outils informatiques pour manipuler des échantillons pré-enregistrés ils n'en ont pas le même usage. Nous détaillons ces usages dans la suite du document.

### **5.1.1 Nicolas Rivière**

Nicolas Rivière se définit comme compositeur de musique électronique dans le style IDM, pour Intelligent Dance Music. Âgé de 25 ans c'est le plus jeune des trois musiciens. Il crée une musique mélangeant sons concrets, traitements temps réels et musique électronique. Il se sert de l'outil informatique pour la création musicale depuis 4 ans. Il a suivi une formation en sound design et montage à l'École Supérieure de Réalisation Audiovisuelle. Il a une bonne connaissance des outils informatiques pour la création ainsi que de leurs usages.

Selon le tableau récapitulatif des informations recueillies lors de l'entretien exploratoire présenté à la fin de cette section il se sert essentiellement du logiciel Pro-Tools dont il se définit comme utilisateur expert. Il est aussi habitué à utiliser le GRM player qui de par son format stand alone (qui fonctionne en dehors de toute application tierce) s'approche beaucoup de Mono-Replay.

Il utilise parfois Max/MSP pour créer certains effets et réaliser des mappings gestuels. Il est habitué à réaliser du contrôle temporel d'enregistrement et a souvent du mal à trouver des gestes expressifs pour les réaliser. Mono-Replay l'intéresse particulièrement pour cette tâche.

Nicolas Rivière a déjà travaillé plusieurs fois pour PUCE MUSE en tant que régisseur, animateur d'atelier ou artiste. Il est donc très habitué à l'environnement logiciel de l'association et a déjà réalisé une commande de la sorte pour le logiciel Méta-DJ.

### **5.1.2 Paul Denis Kingue-Manjole (Pehoz)**

Paul Denis Kingue-Manjole se définit comme producteur de musique électronique et beatmaker. Il signe la plupart de ses compositions sous le nom de Pehoz depuis 10 ans. Il travaille au sein du label de hip-hop Jeune à Jamais. Ses productions sonores sont la plupart du temps instrumentales mais il arrive aussi qu'il collabore avec des rappeurs ou rappeuses et des chanteurs ou chanteuses.

Âgé de 31 ans il pratique la musique assistée par ordinateur depuis plus de 10 ans. Il est autodidacte dans les métiers du son et de la musique. Il utilise essentiellement le logiciel Ableton Live pour sa création. Il se définit comme utilisateur expert et intervient parfois dans des ateliers pédagogiques autour de ce logiciel.

Paul Denis pratique le Finger Drumming lors de ses prestations live. Cette pratique consiste à déclencher en rythme la lecture d'un segment enregistré en appuyant sur des pads. Cette pratique le distingue des autres musiciens qui n'interagissent pas de manière aussi rythmée avec la lecture d'un enregistrement. Ses attentes envers Mono-Replay se portent essentiellement sur l'utilisation d'une manette de jeu pour contrôler le logiciel. Étant joueur fréquent de jeux vidéo il a formulé cette envie lors des échanges préliminaires à l'expérience.

Contrairement aux autres musiciens Paul Denis n'a jamais utilisé les outils développés par l'association PUCE MUSE.

### 5.1.3 Basile Chassaing

Basile Chassaing se définit comme compositeur de musique électro-acoustique, mixte, temps réel et concrète. Il est âgé de 35 ans et utilise l'informatique dans sa création depuis 6 ans. Il a été formé aux métiers du son lors d'un BTS audiovisuel option son. Il a aussi suivi une formation de la trompette en conservatoire jusqu'au niveau DEM. Ses créations font intervenir des instruments acoustiques qu'il transforme à l'aide de traitements numériques temps-réels.

Il utilise essentiellement le logiciel Max/MSP pour réaliser ces traitements numériques aussi bien dans la phase de création que dans la phase de restitution scénique. Il utilise aussi beaucoup le logiciel Reaper pour concevoir ses montages sons en amont.

Au moment de l'entretien Basile Chassaing est employé au sein de PUCE MUSE en tant que chargé de production depuis 4 ans. Il est donc très habitué à l'environnement logiciel PUCE MUSE bien qu'il ne l'ait jamais utilisé pour sa création comme il nous l'a confié lors de l'entretien exploratoire dont le tableau récapitulatif est disponible à la fin de cette section. Selon lui le format application des logiciels de PUCE MUSE rend difficile l'intégration de Mono-Replay dans son environnement de travail. Il souhaite néanmoins essayer d'utiliser le logiciel pour sa création.

## 5.2 L'entretien exploratoire

Les entretiens exploratoires ont pour fonction de "mettre en lumière les aspects du phénomène auxquels le chercheur ne peut penser spontanément et de compléter les pistes de travail suggérées par ses lectures". [1] Ils permettent notamment de récupérer des informations utiles pour la conception du guide d'entretien. Ces informations peuvent porter sur la pratique informatique de chaque musicien mais aussi sur sa conception du contrôle temporel.

Ces entretiens préliminaires sont essentiels pour cibler au mieux les points que l'on souhaitera développer durant l'entretien final. Les questions posées durant ces entretiens figurent sur la figure 5.1. Elles ont été organisées autour de trois thèmes : présentation de l'artiste ; place de l'outil informatique dans

le travail de l'artiste ; place du contrôle temporel dans le travail de l'artiste. Le choix de ces thèmes a été discuté en amont avec Gabriela Patino Lakatos qui est habituée à mener ce type d'entretien. Le but est souligner la singularité de chaque musicien dans sa pratique de l'outil informatique. Cette singularité s'explique de par le parcours de chaque musicien, des outils qu'il a l'habitude d'utiliser et de son expérience dans le contrôle temporel d'enregistrement.

Thèmes	Questions
<i>Présentation de l'artiste</i>	<ul style="list-style-type: none"> <li>• Comment l'artiste définit son activité ?</li> <li>• A t-il un statut administratif ?</li> <li>• Comment répartit-il son temps entre le travail de studio et celui de concert ?</li> </ul>
<i>À propos de l'outil informatique</i>	<ul style="list-style-type: none"> <li>• Depuis combien de temps pratique t-il l'informatique musicale ?</li> <li>• Comment s'est t-il formé à cette discipline ?</li> <li>• Sous quelles formes utilise t-il les outils numériques de créations (pour le studio et pour le live) ?</li> <li>• Utilise t-il des interfaces de contrôle gestuelle ?</li> <li>• Quelle est son installation scénique ?</li> </ul>
<i>À propos du contrôle temporel</i>	<ul style="list-style-type: none"> <li>• A-t-il déjà contrôlé la vitesse et la position de lecture d'un enregistrement pour de la création ?</li> <li>• A-t-il déjà utilisé des échantillonneurs ?</li> <li>• Que lui évoque la notion "contrôle temporel" ?</li> </ul>

FIGURE 5.1 – Les questions posées durant l'entretien exploratoire ont été organisé selon trois thèmes. Le but de ces questions est de dresser un profil des musiciens interrogés pour mieux préparer les entretiens finaux

Une grille résumée de ces entretiens exploratoires est disponible à la fin de cette section. Grâce à elle on peut souligner les similarités suivantes entre chaque musicien :

- Chaque musicien interrogé pratique l'informatique musicale depuis plus de 4 ans.
- Ils sont tous des utilisateurs experts d'outils de type Digital Audio Workstation (DAW).
- La composition en studio occupe la majeure partie de leur temps comparativement à la pratique scénique.
- Ils ont abordé plusieurs fois la question du contrôle temporel d'enregis-



trements dans leurs créations.

Ces similarités nous permettent de concevoir un guide d'entretien plus pointu. Savoir qu'ils sont tous experts dans la manipulation d'au moins un logiciel pour la création musicale nous permet de penser des questions plus techniques.

En dehors des similarités, les entretiens exploratoires nous permettent aussi de définir la singularité de chacun des musiciens interrogés. La manière d'appréhender le contrôle temporel est différente d'un musicien à l'autre et il est important d'en tenir compte pendant l'entretien final. Ces singularités sont résumées dans le tableau récapitulatif disponible à la page suivante. Parmi ces singularités on peut citer celle du musicien Paul-Denis qui pratique souvent le "Finger Drumming" contrairement aux autres. Cette pratique consiste à déclencher la lecture d'un échantillon sonore en tapant sur la surface d'un pad. Ainsi les sons sont lus selon le rythme joué par les doigts du musicien. Les autres musiciens n'ont pas l'habitude de contrôler la position de lecture d'un échantillon d'une manière aussi rythmée, ce qui marque déjà une différence quant aux attentes vis à vis de Mono-Replay. Les entretiens exploratoires nous permettent aussi de connaître l'état d'esprit du musicien avant l'utilisation le logiciel. On pourra ainsi mieux comprendre si son expérience avec le logiciel a été concluante par rapport aux attentes qu'il avait en amont.

**Tableau résumé des entretiens exploratoires**

<b><u>Nom</u></b>	<b><u>Âge/description</u></b>	<b><u>Répartition temps de travail</u></b>	<b><u>Formation liée à la création assistée par ordinateur</u></b>	<b><u>Pratiques instrumentales autres</u></b>	<b><u>Outils utilisés en Studio</u></b>	<b><u>Outils utilisés en Live</u></b>	<b><u>Autour du contrôle rythmique</u></b>
Nicolas Rivière	- 25 ans - Compositeur musique électronique (Intelligent Dance Music) - Intermittent	- Concert : 3 % - Installation régie & formation avec PUCE MUSE (32%) - Composition (65%)	- Il pratique la musique assistée par ordinateur (MAO) depuis 4 ans. - Formation en Sound Design et montage à L'ESRA.	- Chanteur dans des groupes de rock - Participation à des ateliers pédagogiques avec PUCE MUSE	- Pro Tools : création édition et mixage ; utilisateur expert - Rizoiope RX - GRM Player - Utilise très peu de logiciels au format vst (plugin)  <i>Notes : ces logiciels servent essentiellement à faire du sampling et de l'édition.</i>	- Max/MSP : utilisateur moyen - GRM Player - Table de mixage pour contrôler des effets comme l'équalisation ou la compression de plusieurs bandes.  <i>Notes : Pro-Tools sert à faire un rendu sonore de la pièce dans son intégralité. MaxMSP sert à trouver des moyens moyen pour que le travail en studio sonne en concert.</i>	- Il rejoint un échantillon déjà existant à sa façon. - Utilisation de l'Octatrack pour séquencer la lecture d'échantillons. - Il n'effectue pas beaucoup de contrôle temps réel sur la vitesse de lecture. - Il aimerait avoir plus de gestes expressifs pour contrôler le déroulement temporel d'un enregistrement.
Basile Chassaing	- 35 ans - Compositeur musique électro-acoustique/ temps réel/ mixte/ concrète. - Artiste/Auteur	- 90 % composition : écriture de partition, montage son - 10 % répétition, rencontre avec les musiciens et régie informatique pendant les concerts	- BTS audiovisuelle « option son » - 6 ans qu'il utilise l'informatique et l'électronique pour la composition	- Pratique de la trompette en école de musique/ conservatoire (niveau post DEM)	- Max/MSP : création d'effet dans Max ; - Reaper : création d'automations ; simulation de la pièce ; montage mixage ;écriture de partition ;	- MaxMSP pour contrôler les effets audio et lumière - Sur scène les musiciens sont autonomes pour paramétrer les effets élaborés par Basile en amont.	- Il contrôle la vitesse et la position de lecture des échantillons grâce à l'objet groove disponible dans MaxMSP. - Il est difficile de trouver des façons expressives pour contrôler la position et la vitesse de lecture d'un échantillon. - Dans certaines de ses pièces des gestes chorégraphiés servent à mimer des transformations sonores pré-enregistrés
Paul-Denis Kingue – Manjole (nom d'artiste :Pehoz)	- 31 ans - Producteur musique électronique (Beat maker Hip/Hop) - Auto-entrepreneur - Artiste du label Jeune à Jamais	- 50 % Studio (25 % Home Studio ; 25 % Studio extérieur) - 50 % Scénique (25 % Répétition ; 25 % Scène)	- Pratique la MAO depuis 10 ans avec Ableton Live - Autoïdacte dans l'apprentissage de la musique - À suivi une formation de 3 ans dans la vidéo	- Aucune pratique instrumentale autre que la MAO. - Intervient dans des ateliers pédagogiques sur Ableton	- Ableton Live : échantillonnage ; utilisation de plugin externe à Ableton Live ; utilisateur expert	- Ableton Live : Utilise la partie « séquenceur » - Maschine : interface gestuel pour contrôler Live ; Maschine sert à jouer certaines partie de ses morceaux ; - Ableton Push	- Il étire et compresse les échantillon dans le sampler de Ableton Live. - Il contrôle la position de déclenchement du début de l'échantillon. - Il joue des parties très rythmées à l'aide des pads de l'interface Maschine ; Il s'agit souvent des parties de batterie.

### 5.3 Élaboration du guide d’entretien

Le guide d’entretien est le tableau visible à la fin de cette section et qui permet de diriger l’entretien semi-directif. Ce document est commun aux trois entretiens et se constitue d’une liste de questions principales et d’une liste de questions complémentaires sur différents thèmes. Les questions principales sont conçues pour laisser suffisamment de liberté dans la réponse de l’artiste. Elles permettent d’ouvrir un espace de dialogue dans lequel le musicien peut développer sa propre vision du problème. Les questions complémentaires sont quant à elles conçues pour aborder plus en détail certains aspects de la problématique générale.

Comme visible dans le tableau en fin de section, le guide d’entretien a été conçu autour de cinq thèmes principaux : le contrôle temporel ; l’expérience globale du musicien avec le logiciel ; l’expérience du musicien avec l’interface graphique ; l’expérience du musicien avec l’interface gestuelle et les perspectives d’améliorations du logiciel. Le choix de ces thèmes a été réalisé avec Gabriela Patino Lakatos qui a déjà travaillé sur l’évaluation des logiciels Mono.

Les questions portant sur le thème du contrôle temporel ont pour but d’interroger les musiciens sur leur pratique musicale. Contrôler la temporalité d’un enregistrement peut s’effectuer de plusieurs manières différentes (scratch, finger drumming, sampling). Questionner les sujets des entretiens sur leurs habitudes concernant cet aspect de la création nous permet de mieux souligner les attentes de chacun vis-à-vis de Mono-Replay.

Les questions portant sur l’expérience globale de l’artiste avec le logiciel ont pour but d’ouvrir une discussion assez générale sur la philosophie du logiciel. Chaque musicien se sert de Mono-Replay pour une tâche particulière. Selon ce prisme ils n’auront pas la même expérience avec l’outil. Le but ici est de comprendre pourquoi ils ont choisi un usage du logiciel plutôt qu’un autre et comment cela leur a permis d’aboutir au résultat souhaité.

Les questions portant sur l’interface graphique ont pour but de comprendre quels éléments de celles-ci n’ont pas été clairs pour le musicien. L’interface graphique représente une part importante de l’expérience d’un utilisateur avec un logiciel. Pour évaluer l’intégralité du logiciel il est important de questionner

aussi le ressenti de chaque musicien avec cet aspect de Mono-Replay. De plus l'interface graphique de Mono-Replay reprend des éléments communs à tous les Mono. Les remarques faites sur celle-ci pourrait aussi servir à l'amélioration générale des Mono.

Chaque musicien a utilisé une interface gestuelle différente pour contrôler le logiciel. Nous avons préparé plusieurs questions dans le guide d'entretien pour comprendre ce choix. Le but de ces questions est aussi de comprendre l'impact qu'a eu l'interface gestuelle dans l'expérience de l'utilisateur avec le logiciel.

Chaque musicien interrogé a sa propre vision du contrôle temporel voilà pourquoi nous avons décidé de les interroger aussi sur les perspectives du logiciel. Le but est de comprendre comment selon eux Mono-Replay peut évoluer pour permettre plus d'expressivité et plus de facilité d'utilisation.

Pour chaque entretien les questions posées suivent ce guide d'entretien. La durée des entretiens varie entre 40 minutes et 1 heure. Pour Paul-Denis et Nicolas Rivière ils ont été réalisés à leur domicile et Basile au sein de l'institut Jean Le Rond d'Alembert.

Dans les sections suivantes nous présentons les résultats de ces entretiens sous la forme de tableaux synthétiques. Dans ces tableaux figurent les points clefs de chaque réponse des musiciens aux questions du guide d'entretien. L'intégralité des réponses est disponible dans les transcriptions des entretiens annexe [F](#), annexe [D](#) et annexe [E](#).

Pour chaque tableau nous faisons un bilan des réponses. Nous chercherons les aspects de Mono-Replay qui ont servi ou desservi la démarche artistique du compositeur.

### Présentation :

Cet entretien s'intègre dans un travail de recherche portant sur la conception d'outils expressifs pour le contrôle temporel d'un enregistrement. Dans ce travail, un modèle permettant d'ajouter, de paramétrer et de piloter plusieurs modes de contrôle temporel a été implémenté dans un logiciel. Ce logiciel a ensuite été utilisé par des artistes/compositeurs pour leurs créations. Pour évaluer le potentiel expressif de ce système, un entretien a été mené à la suite de cette expérience. En récupérant l'expérience subjective de chaque musicien avec cet outil nous pourrions conclure sur l'apport créatif de ce système par rapport aux autres logiciels de contrôle temporel.

Avec l'accord du participant, le compte rendu sera publié en annexe du manuscrit de thèse et disponible par tous.

Thèmes	Questions principales	Questions complémentaires
Contrôle temporel	- Comment utilisez-vous le contrôle temporel dans vos créations ?	- Quels outils logiciels utilisez-vous pour cette tâche ? - Comment pensez-vous qu'il soit possible d'améliorer ces outils pour permettre plus d'expressivité musicale ?
À propos de l'expérience globale de l'artiste avec le logiciel	- Comment s'est passée votre expérience avec Mono-Replay ?	- Comment avez-vous choisi les enregistrements que vous manipulez ? - Qu'est-ce que Mono-Replay a apporté de plus à vos outils habituels ? - Seriez-vous arrivé à un résultat équivalent en utilisant un autre système ? - Quels ont été les motivations concernant le paramétrage du logiciel ? - Quel mode de contrôle temporel avez-vous utilisé ? - Pourquoi n'avez-vous pas utilisé les autres modes ? - Quelles ont été les difficultés pour intégrer Mono-Replay dans votre environnement habituel de travail ?
À propos de l'interface graphique	- Comment l'interface graphique vous a aidé dans votre création ?	- Quel est votre ressenti général sur l'interface graphique du logiciel ? - Comment l'interface vous a aidé dans votre création ? - Quels aspects de l'interface vous ont paru servir l'expressivité musicale ? - Quelles informations manquait-il sur l'interface ?
À propos de l'interface gestuelle	- Comment l'interface gestuelle vous a aidé dans votre création ?	- Pourquoi avez-vous choisi cette interface plutôt qu'une autre ? - Quel geste de contrôle vous évoque l'enregistrement que vous avez choisi ? - Comment avez-vous choisi la segmentation de l'enregistrement ? - Quels aspects de l'interface gestuelle vous ont paru servir l'expressivité musicale ? - Quelles furent les facilités et les difficultés d'utilisation de cette interface ?
Perspectives	- Quelles sont les perspectives d'améliorations de Mono-Replay ?	- Comment peut-on améliorer l'expressivité du logiciel selon vous ? - Comment peut-on améliorer sa facilité d'utilisation selon vous ?

## 5.4 À propos du contrôle temporel

Questions	Nicolas R.	Pehoz	Basile C.
<i>Comment utilisez vous le contrôle temporel dans vos créations ?</i>	<ul style="list-style-type: none"> <li>• Pour de l'édition via des logiciels de montage audio.</li> <li>• Compresse ou étire la durée d'un échantillon</li> </ul>	<ul style="list-style-type: none"> <li>• Place des marqueurs temporels au début de l'attaque d'un son puis déclenche sa lecture de façon rythmé</li> </ul>	<ul style="list-style-type: none"> <li>• Fait varier la position de déclenchement d'un échantillon et change sa fenêtre de lecture.</li> </ul>
<i>Quels outils logiciels utilisez-vous pour cette tâche ?</i>	<ul style="list-style-type: none"> <li>• GRM player pour l'expressivité</li> <li>• Max/MSP pour développer ses propres outils</li> <li>• Pro-Tools pour l'édition et le montage audio</li> </ul>	<ul style="list-style-type: none"> <li>• Ableton Live pour l'édition et les outils de visualisation</li> <li>• Akai MPC pour le contrôle gestuel</li> </ul>	<ul style="list-style-type: none"> <li>• Max/MSP et l'objet groove en particulier</li> </ul>
<i>Comment pensez-vous qu'il soit possible d'améliorer ces outils pour permettre plus d'expressivité musicale ?</i>	<ul style="list-style-type: none"> <li>• En développant un outil hybride qui intègre l'expressivité du GRM player et la précision de Pro-Tools</li> </ul>	<ul style="list-style-type: none"> <li>• En améliorant l'accessibilité des outils d'étiquetage de Ableton Live et de la MPC</li> </ul>	<ul style="list-style-type: none"> <li>• Réduire les artefacts sonores dus à l'étirement temporel des enregistrements</li> </ul>

FIGURE 5.2 – Tableau récapitulatif des réponses données par chaque musicien aux questions abordant le thème du contrôle temporel. Les transcriptions complètes de chaque entretien sont disponibles en annexe

Dans cette sous-section nous analysons les réponses données par chaque musicien aux questions présentées dans la figure 5.2 et portant sur le contrôle temporel. L'intégralité des réponses est disponible dans les transcriptions d'entretien annexe F, annexe D et annexe E.

### Réponses de Nicolas Rivière :

Nicolas compresse et étire la durée des enregistrements qu'il manipule dans le logiciel Pro-Tools. Pour chercher la matière sonore dont il a besoin il utilise souvent le GRM player [33]. Ce logiciel développé par Adrien lefèvre permet de contrôler la lecture de plusieurs enregistrements via des lecteurs numériques paramétrables. Très proche de Mono-Replay dans sa forme et son esprit le GRM Player est utilisé par Nicolas avant tout pour ses possibilités expressives. Les fonctionnalités de cet outil sont peu nombreuses ce qui permet de

se focaliser rapidement sur des paramètres de jeux plutôt que des paramètres de réglage. Son expérience antérieure avec le GRM player est sûrement une des raisons pour lequel Nicolas a pris Mono-Replay en main aussi facilement. Les deux logiciels sont au format application, permettent de contrôler des enregistrements de manière expressive et intègrent un enregistreur intégré pour exporter les séquences produites. Selon Nicolas il serait possible d'améliorer de façon générale les outils de contrôle temporel déjà existants en créant des logiciels hybrides intégrant certaines particularités de chaque logiciel. Ainsi on obtiendrait la précision de Pro-Tools pour le montage sonore et l'expressivité du GRM Player pour la recherche de texture.

#### **Réponses de Paul-Denis Kingue-Manjole :**

Ce musicien a une approche beaucoup plus performative du contrôle temporel. Effectivement il utilise des interfaces gestuelles pour déclencher la lecture de segments qu'il a préalablement préparés. Le déclenchement de ces segments se fait la plupart du temps de façon rythmée en tapant sur la surface de l'interface utilisée. Selon lui, les outils de contrôle temporel qu'il utilise pourraient être améliorés en rendant l'étape d'étiquetage plus accessible.

#### **Réponses de Basile Chassaing :**

Basile utilise le contrôle temporel pour faire varier la position de déclenchement et la fenêtre de lecture d'un échantillon. Le plus gros problème qu'il rencontre, lorsqu'il utilise des outils pour étirer temporellement des enregistrements, est la présence d'artefacts. Ces artefacts, dûs au changement d'échelle temporelle de l'enregistrement, ont tendance à dénaturer le rendu audio. Il serait selon lui possible d'améliorer les outils de contrôle temporel déjà existants en réduisant ces artefacts.

## **5.5 Au sujet du ressenti global de l'expérience**

Dans cette sous-section nous analysons les réponses données par chaque musicien aux questions présentées dans la figure 5.3 et 5.3. L'intégralité des réponses est disponible dans les transcriptions d'entretien en annexe F, annexe

Questions	Nicolas R.	Pehoz	Basile C.
<i>Comment s'est passé votre expérience avec Mono-Replay ?</i>	<ul style="list-style-type: none"> <li>• Bien</li> <li>• Les nombreux échanges ont permis de faciliter l'installation et la prise en main</li> </ul>	<ul style="list-style-type: none"> <li>• Très bien</li> <li>• Le choix de la manette pour contrôler le logiciel a été un vrai plus</li> </ul>	<ul style="list-style-type: none"> <li>• Utilisation compliquée car le système d'exploitation de l'ordinateur n'était pas compatible avec le logiciel</li> <li>• Interface de segmentation difficile à comprendre</li> </ul>
<i>Comment avez vous choisi les enregistrements que vous manipulez ?</i>	<ul style="list-style-type: none"> <li>• Le premier projet disponible a donné envie de choisir un enregistrement de piano rythmé. Le souhait était de pouvoir le jouer rubato</li> <li>• Les sons de piano sont synthétiques</li> </ul>	<ul style="list-style-type: none"> <li>• Envie de manipuler des sons "vintage"</li> <li>• Intéressé pour manipuler un enregistrement de la voix de Nina Simone et de la batterie du groupe Three Degrees</li> </ul>	<ul style="list-style-type: none"> <li>• Trouver un geste pour contrôler un enregistrement de voix.</li> </ul>
<i>Qu'est ce que Mono-Replay a apporté de plus à vos outils habituels ?</i>	<ul style="list-style-type: none"> <li>• Il amène beaucoup plus d'expressivité musicale.</li> <li>• Le logiciel permet de montrer le geste</li> </ul>	<ul style="list-style-type: none"> <li>• La possibilité de pouvoir gérer en temps réel à la fois le contrôle temporel et les effets</li> </ul>	<ul style="list-style-type: none"> <li>• Le format application pose un problème d'intégration. MR aurait apporté un plus si il avait été sous la forme d'une librairie Max/MSP.</li> <li>• Le fait que le moteur audio soit intégré à MR est un frein</li> </ul>

FIGURE 5.3 – Première partie du tableau récapitulatif des réponses données par chaque musicien aux questions portant sur le ressenti global de l'expérience avec Mono-Replay. Les transcriptions complètes de chaque entretien sont disponibles en annexe

D et annexe E.

### Réponses de Nicolas Rivière :

Nicolas a été le musicien le plus à l'aise dans l'utilisation de Mono-Replay. En plus de l'utiliser pour créer des sons, il a conçu un dispositif composé de trois interfaces gestuelles pour le contrôler. C'est aussi le seul musicien qui a l'habitude d'utiliser des logiciels de type application dédié au contrôle temporel comme le GRM player. L'intégration du logiciel dans son environnement de travail s'est donc faite sans problème. De plus, de nombreux échanges ont



eu lieu avec lui durant la phase de composition. Ces échanges portaient sur des propositions d'améliorations du logiciel pour permettre plus de possibilités expressives mais aussi sur des difficultés d'installation. Comme précisé dans la section 4.3 la création de Nicolas est constituée en partie d'un enregistrement qu'il a composé et qu'il manipule à l'aide du logiciel. Le choix de composer cet enregistrement est survenu à la suite de son utilisation du "premier projet" qui s'ouvre par défaut après l'installation de Mono-Replay. Dans celui ci, un enregistrement de la *Gymnopédie n°1* de Érik Satie peut être manipulé compte tenu de plusieurs types de segmentation. En manipulant cet enregistrement Nicolas a eu pour souhait de concevoir un enregistrement intégrant des sons de piano et une rythmique prononcée tout comme la *Gymnopédie n°1*. Il avait pour souhait de pouvoir contrôler cet enregistrement de manière rubato en ralentissant le tempo à la fin de certaines phrases. Pour sa création il utilise le mode "Speed direct" et "Srcub" afin de contrôler la vitesse et la position de lecture de son enregistrement. Selon lui, le logiciel amène beaucoup plus d'expressivité que les autres outils permettant de contrôler la position et la vitesse de lecture d'un enregistrement. Les modes de contrôles rythmiques permettent selon lui de montrer le geste qui est à l'origine de la transformation temporelle en cours.

#### **Réponses de Paul-Denis Kingue-Manjole :**

L'expérience générale avec le logiciel s'est aussi bien passée pour Paul-Denis. De façon générale, ce musicien aborde la musique comme un jeu. La possibilité d'utiliser la manette de jeu pour contrôler le logiciel a donc été un vrai plus pour lui. Paul-Denis est l'artiste qui a utilisé le plus de modes de contrôle temporel différents. Pour réaliser l'enregistrement *BAR MONO-REPLAY* il a utilisé les modes "Speed-direct", "Speed-freez" et "Loop-b". L'idée était de pouvoir adapter la vitesse de lecture d'un enregistrement de *My baby just cares for me* de Nina Simone et de *Vocal soul train* par The Three Degrees à un arrangement déjà existant. Il a ensuite appliqué des effets et fait varier la vitesse sur certains passages. Selon lui, gérer en temps réel le contrôle temporel et les effets de transformation audio est très difficile avec ses outils habituels que sont Ableton Live et Akai MPC. C'est cet aspect de Mono-Replay qui

l'intéresse particulièrement. Du point de vue de l'intégration du logiciel dans son espace de travail habituel, la difficulté a résidé dans la configuration des entrées audio de Ableton Live de manière à pouvoir récupérer directement le son produit par Mono-Replay. Le logiciel Jack a dû être utilisé à cet effet.

### **Réponses de Basile Chassaing :**

Pour Basile l'expérience avec Mono-Replay n'a pas été concluante. L'idée principale était d'utiliser le logiciel pour trouver des gestes afin de contrôler un enregistrement de voix. Le premier frein a été les problèmes d'incompatibilité entre son système d'exploitation et le format de l'application. Effectivement la version du logiciel utilisée pour développer Mono-Replay n'est pas compatible avec des systèmes d'exploitation mac de plus de 6 ans. Le logiciel dans sa version compilée en application ne peut donc être ouvert sur l'ordinateur de Basile sauf si celui-ci décide de mettre à jour son système d'exploitation. De part cette incompatibilité, Basile a dû utiliser Mono-Replay en intermittence sur un autre ordinateur qui n'était pas le sien. La prise en main a donc été ralentie par cet aspect. Par ailleurs, le format application de Mono-Replay a posé un réel problème pour Basile. Habitué à composer tout dans un même logiciel, il était difficile pour lui de trouver un moyen d'intégrer Mono-Replay sous cette forme dans l'environnement qu'il utilise habituellement (Max7). Selon lui, le choix du format application n'est pas le plus adapté pour Mono-Replay. Il aurait fallu le décliner en librairie d'objets directement accessible dans Max7 pour qu'il soit plus facile à prendre en main. Par ailleurs l'intégration d'un moteur audio au sein de l'application est selon lui un frein pour développer l'expressivité avec le logiciel. Selon cette vision Mono-Replay serait un ensemble de fonctions calculant un indice temporel, correspondant à la position de la tête de lecture dans l'enregistrement, que l'on pourrait "brancher" en entrée du moteur audio de notre choix. Ainsi, on gagnerait en flexibilité et il serait possible de choisir le moteur audio indépendamment de la méthode de calcul de l'indice temporel. Certains aspects ergonomiques du logiciel ont aussi freiné son utilisation. Parmi ceux-ci figurent la complexité de l'éditeur de segments qui a aussi été soulignée par Paul-Denis. Les modes de contrôle temporel proposés n'ont pas permis à Basile de trouver une façon satisfaisante

de contrôler l'enregistrement de voix.

Questions	Nicolas R.	Pehoz	Basile C.
<i>Seriez vous arrivé à un résultat équivalent en utilisant un autre système ?</i>	<ul style="list-style-type: none"> <li>• Possibilité de refaire certaines choses avec d'autres outils mais avec beaucoup de difficultés</li> </ul>	<ul style="list-style-type: none"> <li>• Impossibilité de contrôler la position de lecture tout en contrôlant les effets avec Ableton Live et Akai MPC.</li> </ul>	<ul style="list-style-type: none"> <li>• N'a pas abouti à un résultat fini</li> </ul>
<i>Quel mode de contrôle temporel avez-vous utilisé ?</i>	<ul style="list-style-type: none"> <li>• Speed direct : pour rejouer l'enregistrement en conservant son rythme tout en jouant sur les effets et la vitesse de lecture</li> <li>• Scrub : pour chercher des textures plus expérimentales et pouvoir faire des "gel" audio</li> </ul>	<ul style="list-style-type: none"> <li>• Speed direct</li> <li>• Speed freez</li> <li>• Loop-b</li> </ul>	<ul style="list-style-type: none"> <li>• A parcouru tous les modes mais n'a pas trouvé ceux qui pourraient l'intéresser</li> </ul>
<i>Quelles ont été les difficultés pour intégrer Mono-Replay dans votre environnement habituel de travail ?</i>	<ul style="list-style-type: none"> <li>• Pas de difficultés car a l'habitude de travailler avec des applications et de ne pas utiliser le même logiciel pour tout faire</li> </ul>	<ul style="list-style-type: none"> <li>• Pouvoir récupérer le son de Mono-Replay dans Ableton Live. Il a fallu utiliser un logiciel tierce</li> </ul>	<ul style="list-style-type: none"> <li>• Système d'exploitation trop vieux et pas compatible avec Max/MSP</li> <li>• Trop de mise à jour du logiciel. Difficulté de suivre le versioning</li> </ul>

FIGURE 5.4 – Deuxième partie du tableau récapitulatif des réponses données par chaque musicien aux questions portant sur le ressenti global de l'expérience avec Mono-Replay. Les transcriptions complètes de chaque entretien sont disponibles en annexe

## 5.6 À propos des interfaces gestuelles

Questions	Nicolas R.	Pehoz	Basile C.
<i>Comment l'interface gestuelle vous a aidé dans votre création ?</i>	<ul style="list-style-type: none"> <li>• Les potentiomètres ont permis de contrôler les modes de contrôle temporel et la fréquence des LFO</li> <li>• Les faders ont permis de contrôler le mixage des effets</li> <li>• Les boutons ont permis de lire/arrêter l'enregistrement, de naviguer dans l'enregistrement segments par segments ou mesures par mesures ou structures par structures</li> </ul>	<ul style="list-style-type: none"> <li>• La manette a été un vrai plus. A permis de rendre la manipulation du logiciel beaucoup plus intuitive car Pehoz utilise souvent ce genre d'interface gestuelle pour jouer aux jeux vidéo.</li> </ul>	<ul style="list-style-type: none"> <li>• N'a pas utilisé d'interface autre que le clavier et la souris</li> </ul>
<i>Pourquoi avez vous choisi cette interface plutôt qu'une autre ?</i>	<ul style="list-style-type: none"> <li>• Pouvoir faire des gestes violents en gardant une certaine précision</li> </ul>	<ul style="list-style-type: none"> <li>• La familiarité de la manette de jeux vidéo</li> </ul>	<ul style="list-style-type: none"> <li>• N'a pas utilisé d'interface autre que le clavier et la souris</li> </ul>
<i>Quel geste de contrôle vous évoque l'enregistrement que vous avez choisi ?</i>	<ul style="list-style-type: none"> <li>• Ma composition m'évoque des gestes qui "tapent fort" comme la rythmique du morceau</li> <li>• Souhaitait pouvoir rejouer l'enregistrement de manière violente et être décisif sur les choix d'effets</li> <li>• Avoir des gestes musicaux qui permettent de prendre des risques</li> </ul>	<ul style="list-style-type: none"> <li>• Le geste du pianiste pour contrôler l'enregistrement de Nina Simone</li> </ul>	<ul style="list-style-type: none"> <li>• Le rythme corporel de danseur devait faire varier la vitesse de lecture de l'enregistrement</li> <li>• Souhaitait contrôler des sons avec une certaine régularité</li> <li>• Si le geste s'arrête le son doit aussi s'arrêter</li> </ul>
<i>Quelles furent les facilités et les difficultés d'utilisation de cette interface ?</i>	<ul style="list-style-type: none"> <li>• Interfaces déjà familières</li> <li>• Pas de difficultés</li> </ul>	<ul style="list-style-type: none"> <li>• Interfaces déjà familières</li> <li>• Appréhende la musique comme un jeu donc la manette de jeux-vidéo l'a conforté dans cette idée</li> </ul>	<ul style="list-style-type: none"> <li>• N'a pas utilisé d'interfaces</li> </ul>

FIGURE 5.5 – Tableau récapitulatif des réponses données par chaque musicien aux questions portant sur les interfaces gestuelles utilisées pour contrôler Mono-Replay. Les transcriptions complètes de chaque entretien sont disponibles en annexe

**Réponses de Nicolas Rivière :**

Nicolas est le seul musicien à avoir conçu son propre dispositif pour contrôler Mono-Replay. Les trois interfaces qu'il a choisies lui ont permis d'avoir accès à trois types de geste qu'il souhaitait inclure dans la création. Le geste de potentiomètre lui permet d'avoir un contrôle précis sur la vitesse et la position de lecture de l'enregistrement ainsi que la fréquence des oscillateurs. Les faders lui permettent d'avoir un contrôle sur le mixage des effets. La robustesse de ces derniers lui permette de faire des gestes "violents" en gardant une certaine précision. Ces gestes "violents" ont été choisis compte tenu de la rythmique marquée du morceau. Il souhaitait réaliser ces gestes pour configurer les différents effets de Mono-Replay tout en gardant une certaine précision. La précision de ces gestes a été conçus pour garder une certaine "prise de risque" lors de leur exécution à la manière d'un instrumentiste.

**Réponses de Paul-Denis Kingue-Manjole :**

Paul-Denis a choisi spontanément la manette de jeux-vidéo comme interface gestuelle lorsque nous lui avons présenté les différents usages du logiciel. Il a toujours appréhendé la création musicale comme un jeu. Le choix de cette interface s'est donc fait très naturellement. Il avait par le passé imaginé plusieurs fois utiliser une manette de jeu comme contrôleur pour ses outils habituels. Le fait de pouvoir expérimenter cela avec Mono-Replay a été pour lui très motivant. La familiarité de l'interface lui a permis de prendre en main très rapidement le logiciel et de réaliser sans problème les effets qu'il avait en tête. S'il n'avait pas eu de propositions d'interfaces et avait dû imaginer un geste pour contrôler l'enregistrement choisi, Paul-Denis aurait essayé de reproduire le geste du pianiste. Ce geste lui est venu à l'esprit suite à l'écoute de la très célèbre phrase de piano contenu dans l'enregistrement de *My baby care for me* de Nina Simone. Ce geste évoqué vient souligner l'importance du substitut gestuel dans la conception d'une interaction. Le substitut gestuel fait référence aux travaux de Denis Smalley que nous présentons dans la section 1.2.

**Réponses de Basile Chassaing :**

La volonté première de Basile était d'utiliser Mono-Replay pour contrôler la

lecture d'enregistrement de voix parlée avec le rythme corporel de danseurs. Il souhaitait contrôler ces sons avec une certaine régularité tout en gardant un contrôle précis sur ceux-ci. Si le danseur s'arrête, la lecture de l'enregistrement de voix devait aussi s'arrêter. Il n'a pas utilisé d'autres interfaces que le clavier et la souris pour explorer le logiciel. Cela vient du fait qu'il a eu de grosses difficultés pour intégrer Mono-Replay à son environnement de travail. Son système d'exploitation n'étant pas compatible avec Max8, il ne pouvait ouvrir la version compilée ni la version projet du logiciel sur son ordinateur personnel.

## **5.7 À propos de l'interface graphique**

Dans cette série de questions nous souhaitons récupérer des informations précises sur l'expérience de l'utilisateur avec l'interface graphique. Bien que cet aspect ne soit pas directement lié à l'expressivité musicale, nous souhaitions savoir dans quelle mesure il a aidé le compositeur dans sa création.

### **Réponses de Nicolas Rivière :**

L'interface graphique a aidé Nicolas à comprendre la philosophie du logiciel. Son expérience avec celle-ci a été assez bonne. La manière dont elle était agencée et les sigles utilisés lui ont permis de comprendre les fonctionnalités en détail. Une fois qu'il a compris son fonctionnement il a cessé de prêter attention à l'interface graphique et s'est focalisé sur le son produit par le logiciel pour la création. L'interface graphique n'a donc pas directement agi sur un aspect créatif de la pièce.

### **Réponses de Paul-Denis Kingue-Manjole :**

Pour ce musicien, l'expérience avec l'interface graphique a été relativement bonne. Les différents sigles présents sont pour la plupart utilisés dans les autres logiciels de ce type, ce qui a été apprécié par Paul-Denis . La visualisation des segments sur la fenêtre principale n'a pas vraiment aidé Paul-Denis dans sa tâche et l'aspect global de cette fenêtre lui a plutôt évoqué un outil dédié à l'enregistrement plutôt qu'au contrôle temporel. Selon lui l'outil de segmentation intégré dans Mono-Replay est trop complexe. Dans cette sous-fenêtre,

Questions	Nicolas R.	Pehoz	Basile C.
<i>Comment l'interface graphique vous a aidé dans votre création ?</i>	<ul style="list-style-type: none"> <li>• Permet de comprendre la philosophie du logiciel et ses fonctionnalités.</li> <li>• Une fois que le logiciel est compris le visuel n'est plus pris en compte pour la création</li> </ul>	<ul style="list-style-type: none"> <li>• La présence de sigles communs aux autres logiciels aide à mieux le comprendre.</li> <li>• La visualisation des segments n'a pas vraiment aidé à comprendre le logiciel.</li> </ul>	<ul style="list-style-type: none"> <li>• L'aide contextuel n'est pas apparu dans la fenêtre de segmentation</li> <li>• Il manque la possibilité de modifier le port OSC via l'interface graphique</li> </ul>
<i>Quel est votre ressenti général sur l'interface graphique du logiciel ?</i>	<ul style="list-style-type: none"> <li>• Assez claire</li> </ul>	<ul style="list-style-type: none"> <li>• L'interface graphique évoque un outil dédié à l'enregistrement audio plus qu'au contrôle temporel</li> <li>• L'éditeur de segment est trop complexe. Il y'a trop d'informations et trop de segments. Il faut peut être changer les couleurs de ce menu.</li> <li>• Difficile de comprendre les différents effets</li> </ul>	<ul style="list-style-type: none"> <li>• Il faudrait plus de raccourcis au clavier</li> <li>• Il faudrait retrouver certaines fonctionnalités d'autres logiciel comme open music</li> </ul>
<i>Quels aspects de l'interface vous ont paru servir l'expressivité musicale ?</i>	<ul style="list-style-type: none"> <li>• L'interface graphique sert à comprendre les fonctionnalités du logiciel mais pas l'expressivité. L'expressivité réside dans les possibilités sonores offertes par celui-ci</li> </ul>	<ul style="list-style-type: none"> <li>• Les icônes m'ont bien aidé a comprendre le logiciel</li> </ul>	<ul style="list-style-type: none"> <li>• L'interface est bien pour un usage pédagogique mais pour un usage artistique c'est trop limité.</li> </ul>
<i>Quelles informations manquait-il sur l'interface ?</i>	<ul style="list-style-type: none"> <li>• Il ne manque pas d'informations</li> </ul>	<ul style="list-style-type: none"> <li>• Un système d'arborescence pour accéder aux fichiers de l'ordinateur. Cela faciliterai l'importation d'échantillons audio.</li> </ul>	

FIGURE 5.6 – Tableau récapitulatif des réponses données par chaque musicien aux questions abordant le thème de l'interface graphique. Les transcriptions complètes de chaque entretien sont disponibles en annexe

trop de segments sont affichés et cela a tendance à rendre la compréhension de cette fonction trop compliquée. Selon lui l’interface graphique de Mono-Replay pourrait être améliorée. Il faudrait pour cela intégrer un système d’arborescence grâce auquel il serait possible à l’utilisateur de naviguer dans les répertoires de l’ordinateur. Ainsi, on pourrait facilement importer de nouveaux fichiers audio et explorer leur rendu sonore une fois qu’ils sont transformés.

#### Réponses de Basile Chassaing :

Selon Basile l’interface de Mono-Replay est plus dédiée à un usage pédagogique qu’à un usage artistique. Il manque selon lui la possibilité de paramétrer le port OSC en direct pour faciliter la configuration de l’interface gestuelle que l’on utilisera. Les raccourcis claviers pourraient faciliter la prise en main et il serait bien d’intégrer des fonctions déjà existantes sur des logiciels de composition comme Open Music.

## 5.8 Les perspectives de Mono-Replay selon les musiciens

Questions	Nicolas R.	Pehoz	Basile C.
<i>Quelles sont les perspectives d'améliorations de Mono-Replay ?</i>	<ul style="list-style-type: none"> <li>• Explorer le concept du beat-repeat pour avoir possibilité de modifier certains aspects du contrôle temporel et revenir à des réglages précédents.</li> <li>• Ajouter une fonction pour modifier de façon aléatoire les paramètres du logiciel selon le découpage des segments.</li> </ul>	<ul style="list-style-type: none"> <li>• Donner plus d'accessibilité pour importer les audio depuis l'ordinateur avec un menu en arborescence</li> <li>• Ajouter un mode pour faire des “sidechains” et pouvoir faire osciller des paramètres depuis l'audio</li> <li>• Allez plus loin dans la gestion des boucles</li> </ul>	<ul style="list-style-type: none"> <li>• Améliorer le son du time stretch.</li> <li>• Donner plus d'exemples déjà indexés</li> </ul>

FIGURE 5.7 – Tableau récapitulatif des perspectives d’améliorations du logiciel selon chaque musicien. Les transcriptions complètes de chaque entretien sont disponibles en annexe

À la fin de chaque entretien nous avons abordé la question des perspectives du logiciel qui seraient, selon eux, intéressantes d’aborder pour la suite du développement de Mono-Replay. Les réponses à cette question ont été syn-



thétisées dans la figure 5.7.

#### **Réponses de Nicolas Rivière :**

Selon ce musicien, il serait intéressant d'intégrer deux nouvelles fonctions dans Mono-Replay. La première permettrait d'explorer le concept de "Beat-Repeat" au sein de ce logiciel. L'idée serait de pouvoir changer de mode de contrôle temporel en gardant l'évolution de l'indice temporel du précédent mode. On pourrait ainsi garder en mémoire l'évolution de l'indice temporel du précédent mode alors qu'un autre mode est utilisé. Selon Nicolas, il serait aussi intéressant d'intégrer une fonction paramétrant de façon aléatoire les effets de transformation spectrale de Mono-Replay. On pourrait ainsi obtenir de nouveaux réglages de ces effets à chaque début de segment.

#### **Réponses de Paul-Denis Kingue-Manjole :**

Selon Paul-Denis, une des perspectives qu'il serait intéressant d'aborder dans la suite du développement de Mono-Replay concernerait un aspect visuel du logiciel. Selon lui il serait en effet utile de donner plus d'accessibilité à la fonction d'importation des sons dans le logiciel. Ainsi, il imagine un système d'arborescence, équivalent à celui disponible dans Ableton Live, avec lequel on pourrait facilement importer de nouveaux sons et chercher de nouvelles textures sonores. Selon lui, il serait aussi intéressant de rajouter la possibilité de contrôler des effets directement depuis le son. Ainsi on pourrait par exemple créer des effets de type sidechain pour faire varier certains effets directement depuis le niveau sonore. Il serait aussi intéressant selon lui de développer davantage la gestion des boucles en ajoutant plus de possibilités pour les paramétrer.

#### **Réponses de Basile Chassaing :**

Selon ce musicien, le principal axe de perspective concernant Mono-Replay serait l'amélioration de son moteur audio. Les artefacts dus à l'étirement temporel des enregistrements sont selon lui trop présents à l'écoute. Il serait, selon lui, aussi intéressant de proposer plus d'exemples de segmentations dans le premier projet du logiciel.

## 5.9 Synthèse

Durant cette expérience, chaque musicien a été sollicité pour utiliser Mono-Replay dans des contextes différents. On a demandé au musicien Paul-Denis d'utiliser le logiciel comme un outil de studio afin de créer de la matière sonore et de l'intégrer dans un accompagnement plus riche. Nicolas Rivière et Basile Chassaing ont quant à eux été sollicités pour créer une pièce avec concert dans lequel Mono-Replay servirait d'instruments sur scène. Les attentes quant à l'utilisation du logiciel sont donc différentes. D'un côté il fallait créer de la matière sonore avec l'outil et de l'autre l'intégrer en plus dans un dispositif scénique. Le caractère performatif du geste utilisé pour contrôler le dispositif a donc été moins central pour Paul-Denis, même si l'utilisation de la manette de jeu comme interface a été un des aspects décisifs de sa création. Pour Basile et Nicolas le côté performatif a été au cœur de leur démarche artistique. Bien que Basile n'ait pas abouti au résultat souhaité, l'idée de base était de trouver un geste qui puisse être suffisamment expressif pour évoquer le contrôle syllabique d'une voix parlée.

Si on observe les réponses synthétisées dans la figure 5.2 chaque musicien interrogé a sa propre vision du contrôle temporel. Paul-Denis et Basile ont plus pour habitude d'appréhender cette question directement par des gestes performatifs. Nicolas, lui, est plus habitué à paramétrer de façon non-temps réel des fonctions d'étirement ou de compression temporelle directement depuis ses outils de montage. Dans la pratique, la conception d'un geste pour contrôler la position et la vitesse de lecture de l'enregistrement est un domaine nouveau pour Nicolas. Il a néanmoins trouvé rapidement les solutions nécessaires pour développer son dispositif. L'intégration de gestes expressifs pour contrôler des transformations temporelles est selon lui une piste d'amélioration pour les outils de contrôle temporel qu'il est habitué d'utiliser. Selon Paul-Denis et Basile, les outils de contrôle temporel déjà existants devraient être améliorés sur des aspects plus techniques comme la correction des artefacts des algorithmes d'étirement temporel et l'amélioration des interfaces pour l'étiquetage. Leurs attentes vis-à-vis du logiciel ont donc plus porté sur des aspects techniques que sur la philosophie générale de l'outil.

De façon générale, l'expérience avec le logiciel a été positive pour Paul-Denis et Nicolas. De nombreux échanges ont eu lieu avec ces deux musiciens tout au long de la création, ce qui a fluidifié la compréhension de l'outil et réglé les soucis techniques. Le bon déroulement de l'expérience avec Mono-Replay pour Nicolas vient sûrement de son expérience avec le GRM player. En effet, les deux logiciels partagent beaucoup de similarités. Ils permettent tous deux de contrôler la vitesse et la position de lecture d'un échantillon. Ils intègrent tous deux une fonction pour enregistrer et exporter les créations conçues dans celui-ci. Ils sont tous deux au format application stand alone et s'intègrent de la même façon dans un système d'exploitation. Nicolas a donc pu mettre son expérience dans l'utilisation de ce logiciel au service de la prise en main de Mono-Replay. Pour Basile, l'expérience a été perçue moins positivement. Les problèmes techniques liés à la configuration de son ordinateur portable ont freiné sa prise en main. Basile a néanmoins pu parcourir tout le logiciel sur un autre ordinateur et penser sérieusement son intégration dans la création qu'il imaginait. Le format application a été pour lui une véritable contrainte pour son projet de composition. Habitué de tout développer dans l'environnement Max/MSP il était difficile d'intégrer facilement l'application Mono-Replay dans son patch. De plus, il souhaitait laisser chaque musicien intervenant dans sa création, autonome avec son patch Max/MSP et Mono-Replay. Les complications pour utiliser ces deux formats étaient selon lui trop contraignantes pour laisser des utilisateurs novices les utiliser. Pour Paul-Denis et Nicolas, Mono-Replay a été bien intégrée dans leur environnement de travail. Les possibilités expressives du logiciel ont été les points forts de leur expérience. La possibilité de pouvoir facilement concevoir un geste lié à une transformation temporelle a été très appréciée par Nicolas. Paul-Denis, lui, a particulièrement aimé la possibilité de pouvoir gérer simultanément de façon précise et en temps réel la position de lecture et les effets audio.

Pour les trois musiciens le geste qu'ils ont essayé de développer avec Mono-Replay pour contrôler le logiciel venait directement du geste évoqué par l'enregistrement. Le geste évoqué fait référence au substitut gestuel théorisé par Denis Smalley [54] que nous détaillons dans la section 1.2. Pour Nicolas le rythme très marqué de l'enregistrement qu'il manipulait lui évoquait des gestes vio-

lents qu'il a réussi à reproduire avec les trois interfaces montrées figure 4.13. Pour Paul-Denis le geste qu'il souhaitait développer était celui du pianiste compte tenu de la présence d'un son de piano dans l'enregistrement. L'attrait pour la manette de jeu a pris le dessus sur la volonté de développer son propre dispositif. Néanmoins, le geste évoqué et qu'il aurait souhaité utiliser pour contrôler cet enregistrement, était celui d'un pianiste. Pour Basile le geste initial qu'il souhaitait développer était un geste évoquant la diction d'une phrase préenregistrée. Le choix de ces gestes vient du choix de l'enregistrement de voix qu'il voulait manipuler. Compte tenu de ces trois réponses, il est cohérent de penser que pour développer un outil permettant de contrôler un enregistrement de manière expressive il faut mettre à disposition suffisamment de fonctionnalités pour adapter une interaction au type d'enregistrement que l'on manipule.

## Chapitre 6

# Conclusion

### 6.1 Bilan

Ce travail de recherche a permis d’apporter de nouveaux outils pour améliorer l’expressivité des systèmes permettant de contrôler en direct la lecture d’un enregistrement. L’état de l’art présenté en introduction a permis de comprendre comment s’organisaient ces outils et ainsi de les représenter sous forme de systèmes s’exprimant selon les mêmes variables. D’après le travail de Norbert Schnell [53] on peut voir ces systèmes comme trois blocs algorithmiques fondamentaux fonctionnant en synergie. Ainsi, tous les outils permettant de contrôler le déroulement temporel d’enregistrements audionumériques sont composés d’une fonction permettant d’enregistrer, d’analyser et d’étiqueter un son, d’une fonction permettant de capter et analyser un geste ainsi que d’une fonction permettant de générer un son de façon interactive. Cette décomposition permet de segmenter le problème de l’expressivité pour l’observer au sein de chaque fonction.

Lorsque l’on conçoit un scénario d’interaction avec un enregistrement, la première étape consiste à indexer celui-ci d’informations utiles pour sa manipulation. Les méthodes d’analyse dites de MIR sont des outils précieux pour cette tâche. En concevant un outil capable d’intégrer ces méthodes, de les paramétrer ainsi que de corriger les résultats générés on peut déjà réaliser un très grand nombre d’étiquetages différents. On donne en exemple les possibilités d’étiquetages que l’on peut obtenir à l’aide de l’algorithme de détection de

pulse [59] et de détection de transitoire [6]. On peut rapidement en paramétrant ces méthodes et en corrigeant leurs résultats, segmenter un enregistrement selon le rythme des notes jouées, la pulsation prédominante, ses mesures et les structure musicales le composant. En pensant la fonction d'étiquetage de la sorte on laisse la possibilité au concepteur d'interaction d'adapter sa segmentation au type d'interaction qu'il souhaite mettre en œuvre.

Pour concevoir un outil expressif il est important que la fonction permettant de capter et d'analyser un geste puisse s'adapter facilement à l'interaction que l'on souhaite réaliser. Pour cela nous intégrons dans notre système un format permettant d'écrire les fonctions de contrôle temporel selon les mêmes variables. Parmi ces fonctions on distingue des entrées prédéterminées non-temps réel obtenues au moment de l'étiquetage et des entrées temps réel continues ou discrètes obtenues par des interfaces gestuelles. Ces variables servent à calculer un indice temporel correspondant à la portion de l'enregistrement qui est synthétisé [57] [16] [34]. Cette façon d'écrire les fonctions de contrôle temporel facilite leur intégration au sein d'un même système. Ici nous écrivons 6 modes de contrôle temporel mais il est possible d'en écrire un très grand nombre avec ces mêmes variables. Certains modes que nous décrivons sont discrets et nécessitent une analyse du geste pour paramétrer le mode de contrôle temporel. C'est le cas des modes de contrôle temporel basés sur la battue de tempo. Cette brique d'analyse gestuelle vient se placer entre le contrôle discret reçu par l'interface gestuelle et le calcul de l'indice temporel, ce que nous décrivons dans la section 2.3.3.

Pour vérifier si l'organisation de ces fonctions d'étiquetage et d'analyse du geste permet plus d'expressivité nous les implémentons dans un logiciel. Ainsi il peut être pris en main par des artistes et nous pouvons observer son apport dans un processus créatif. Le logiciel que nous développons s'appelle Mono-Replay. Il intègre un outil d'étiquetage entièrement paramétrable basé sur les algorithmes de traitement du signal de la librairie Essentia [7]. On peut déterminer la position de transitoires avec l'algorithme SuperFlux [6] ainsi que la position de pulses avec les algorithmes Multi-feature Beat Tracking [59] et Degara [20]. On peut corriger ces positions ou en ajouter de nouvelles. On peut aussi créer une segmentation à partir de fonctionnalités simples et ainsi

indexer des éléments de signal qui ne sont pas détectables par des algorithmes de traitement du signal. À cette fonction d’étiquetage s’ajoute une fonction de contrôle gestuel qui peut se décomposer en 12 modules de contrôle temporel et 2 modules de contrôle spectral. Les modules de contrôle temporel suivent le formalisme décrit dans la section 2.3.3. Ainsi ils récupèrent tous en entrée des données temps réels continues ou discrètes issues d’interface gestuelle ainsi que des données non temps réels issues de l’étiquetage. C’est avec ces données que les modules de contrôles temporels déterminent l’indice temporel  $\tau(t)$  correspondant à la portion de l’enregistrement à synthétiser à l’instant  $t$  d’une performance. De par le formalisme utilisé pour écrire ces modules il devient facile de rajouter de nouveaux modes de contrôle temporel. Il suffit de s’assurer que ceux-ci retournent bien un indice temporel  $\tau(t)$  en sortie et récupèrent des données d’étiquetage et des données gestuelles au bon format en entrées. Le mode de contrôle temporel qui équipe T-Voks [57] a d’ailleurs été implémenté dans Mono-Replay. Son intégration a été très rapide car ce mode s’exprime déjà selon les variables que nous décrivons ici.

Le module de contrôle spectral permet de paramétrer le vocodeur de phase SuperVP de manière directe à l’aide d’une enveloppe prédéterminée ou de contrôle continu issu d’interface gestuelle. Pour plus d’expressivité la taille de l’enveloppe évolue selon la taille du segment joué. Ainsi on peut faire évoluer un paramètre de transformation spectrale en utilisant l’indexation réalisée précédemment.

Pour bien comprendre comment construire une interaction à l’aide de Mono-Replay nous détaillons plusieurs exemples nécessitant différents modes de jeux, interfaces gestuelles et fichiers d’étiquetage. Ces interactions sont regroupées compte tenu de l’interface gestuelle qu’elles utilisent et chaque étape de conception y est détaillée. Parmi celles-ci on présente des interactions basées sur les manettes de jeux, le Méta-Instrument 4 [14] et des interfaces midi plus standard. Le but ici est d’illustrer la diversité de gestes que l’on peut facilement concevoir avec l’outil Mono-Replay. Ces gestes vont de contrôles continus précis pour déplacer des paramètres temporels et spectraux à des gestes discrets pour définir une vitesse de lecture par battue de pulsation. Ces exemples démontrent le potentiel expressif d’un système comme Mono-Replay. C’est la

possibilité d'intégrer de nouveaux modes de contrôle temporel, de créer des fichiers d'étiquetages sur mesure et d'avoir accès à toutes les variables du logiciel par le biais d'interfaces gestuelles qui ont permis de concevoir toutes ces interactions. En organisant le logiciel comme décrit dans le chapitre 2 on facilite la conception et la compatibilité d'un nombre important d'interactions.

Le système que nous avons pensé puis implémenté dans Mono-Replay a ensuite été pris en main par trois compositeurs dans le cadre de leur création. Le but ici est d'observer la place de l'outil dans le processus de création afin d'évaluer son apport créatif. Pour cela nous réalisons trois entretiens semi-directifs à l'aide d'un guide d'entretien disponible section 5.3. Les questions posées sont volontairement ouvertes afin de créer un espace de dialogue où le musicien peut décrire scrupuleusement chaque aspect de son expérience avec Mono-Replay. L'analyse de ces entretiens sert à titre d'évaluation qualitative. Contrairement à une évaluation quantitative il n'est pas nécessaire ici d'avoir un nombre d'échantillons important pour tirer des conclusions pertinentes sur le logiciel. En analysant ces entretiens on comprend les avantages et les limites du système que nous avons développé. L'aspect expressif a été souligné par les trois musiciens bien que l'un d'entre eux n'ait pas pu aboutir au résultat qu'il souhaitait. L'un des musiciens a pu développer son propre dispositif pour utiliser le système lors d'une représentation scénique qui a été filmée<sup>1</sup>. Les possibilités offertes par Mono-Replay lui ont permis d'aller plus loin que certains outils notamment par la possibilité selon lui de "montrer le geste" qui sert à contrôler une transformation temporelle. Pour le deuxième musicien le véritable apport expressif résidait selon lui dans le choix de l'interface gestuelle utilisée pour contrôler le logiciel. Dans ce cas il s'agit de la manette de jeu dont le mapping est décrit dans la section 4.1. L'artiste a utilisé cette interface pour faire varier la position de lecture en appliquant des transformations spectrales sur un enregistrement de Nina Simone et du groupe Three Degrees. La matière sonore ainsi créée a été intégrée dans un arrangement plus complexe<sup>2</sup> à la manière d'un sample. Le dernier musicien a rencontré des problèmes liés à l'architecture de son ordinateur et la prise en main du logiciel

---

1. <https://youtu.be/15k080rCriI>

2. [https://drive.google.com/file/d/1hZl27uR\\_eitd\\_tWzsvT\\_G6kILBwk\\_VAo/view?usp=sharing](https://drive.google.com/file/d/1hZl27uR_eitd_tWzsvT_G6kILBwk_VAo/view?usp=sharing)



a été difficile. Néanmoins il a pu parcourir l'intégralité des fonctionnalités de celui-ci et réfléchir sérieusement à la manière dont il pourrait intégrer cet outil dans sa création. Selon lui l'application a un intérêt plus pédagogique que créatif. Par ailleurs selon lui le format compilé du logiciel pose un problème pour être facilement intégré dans son environnement logiciel de création.

Les différents points abordés dans ce travail permettent de conclure sur l'apport créatif du système que nous étudions. De par l'organisation des fonctions qui le composent il est possible grâce à ce système de concevoir des interactions gestuelles complètes allant d'une lecture simple d'un enregistrement à des modes complexes pour contrôler l'indice temporel  $\tau(t)$ . Les 11 modes de contrôle temporel qui y sont intégrés sont écrits selon un formalisme simple. On peut ainsi facilement en ajouter de nouveau et ainsi proposer d'autres scénarios d'interactions.

## 6.2 Perspectives

### 6.2.1 Vers une évaluation quantitative

Lors des échanges qui ont eu lieu dans le cadre des ateliers SMAC décrits dans la section 4.1.1, des questionnaires portant sur l'ergonomie des logiciels Mono ont été conçus par Gabriela Patino-Lakatos. Ces questionnaires ont pour but d'estimer l'utilité potentielle de ces logiciels dans un cadre éducatif ou pédagogique. Ils sont soumis à des enseignants ou des étudiants lorsque ceux-ci arrivent à terme d'un atelier pédagogique avec un logiciel Mono ; ateliers durant lesquels un enseignant accompagne un groupe d'élèves dans l'utilisation du logiciel. Ainsi un groupe de 7 et 4 enfants a rempli ces questionnaires après avoir pris en main Mono-VueSon et Mono-Ovisir dans le cadre d'ateliers pédagogiques.

Ces questionnaires ont aussi été conçus pour Mono-Replay. Il existe un questionnaire axé sur l'ergonomie du logiciel et un questionnaire portant sur l'expérience d'utilisation avec un projet pédagogique et artistique. Pour le moment seul le deuxième questionnaire qui est disponible annexe I a été rempli par les 3 étudiants qui ont participé à l'atelier décrit dans la section 4.1.2. Le contexte ici étant adapté à ces questionnaires dans la mesure où j'ai accom-

pagné ces étudiants dans l'utilisation du logiciel tout au long de ces ateliers.

Ces questionnaires mélangent des questions fermées ou à choix multiple et quelques questions ouvertes. Pour que cette évaluation soit significative il faudrait continuer à faire remplir le questionnaire par d'autres utilisateurs de Mono-Replay. Pour compléter l'évaluation que l'on présente dans notre travail il serait intéressant de continuer à soumettre les questionnaires à d'autres utilisateurs.

### 6.2.2 Intégrer Mono-Replay dans d'autres formes de création

L'utilisation de Mono-Replay pour aboutir à une création artistique complète a été une expérience très enrichissante pour le développement de l'outil. Les artistes qui ont utilisé le logiciel ont chacun une pratique différente du contrôle temporel. Ainsi ils ont chacun souligné des aspects particuliers qu'ils souhaiteraient voir se développer au sein de l'outil pour le futur. Ces informations sont cruciales pour les futurs développements de Mono-Replay.

Pour obtenir encore plus d'expressivité il serait intéressant d'intégrer le logiciel dans d'autres formes d'expressions. On pourrait imaginer une création faisant intervenir plusieurs instances du logiciel ou encore une création faisant intervenir d'autres instruments que Mono-Replay. Ainsi on observerait le système dans un autre contexte et en récupérant l'expérience du musicien avec l'outil on soulignerait d'autres pistes d'améliorations pour développer l'expressivité du logiciel.

C'est dans cette optique que nous avons organisé le deuxième concert avec le musicien Nicolas Rivière. L'idée ici est d'utiliser le même dispositif pour contrôler Mono-Replay que pour *SUPERNOME DIVA 2 - contemplation du vide* mais cette fois-ci pour improviser avec un autre musicien. Cet autre musicien est Elias Cheyrou qui utilise des samplers et des synthétiseurs pour jouer aux côtés de Nicolas Rivière. La représentation de cette nouvelle pièce intégrant Mono-Replay a eu lieu le 4 février 2022 au dernier étage de la tour Zamanksy à Sorbonne Université.

### 6.2.3 Rythmic Control Unit : une librairie de fonctions pour le contrôle temporel

Une des spécificités du logiciel Mono-Replay est qu'il possède 12 modes différents de contrôle temporel. Pour permettre de les faire tous coexister au sein de la même architecture ces 12 modes ont été écrits selon les mêmes variables. Dans le code du logiciel ces modes prennent l'allure de "boîtes" représentées figure 3.10 qui contiennent un code spécifique pour calculer une valeur d'indice temporel qu'elles renvoient. En respectant cette forme d'écriture il est possible d'ajouter autant de fonctions de contrôle temporel que l'on souhaite tout en gardant de la simplicité dans leur gestion : une fois qu'une "boîte" est ajoutée il suffit d'indiquer son nom dans les fonctions de menu déroulant et de gestion de mémoire pour que le mode de contrôle temporel soit compatible avec Mono-Replay.

Cette simplification permet de faciliter des gestions plus expressives de ces modes de contrôles comme par exemple le système d'interpolation de la mémoire décrit section 3.6.2. C'est aussi cette organisation des modes de contrôles qui nous permettra facilement d'intégrer l'idée du "Beat Repeat" énoncée par le compositeur Nicolas Rivière lors de l'entretien disponible en annexe D. On pourrait facilement à chaque instant  $t$  d'une performance, changer de mode de contrôle temporel tout en gardant le déroulement de l'indice temporel  $\tau(t)$  selon le mode précédent. C'est en grande partie ce travail d'unification de l'écriture des modes de contrôles temporel qui permet de passer d'un mode à l'autre sans difficulté pendant une performance.

Les modes de contrôle temporel sont des fonctions intervenant en amont du moteur audio. Les calculs de l'indice temporel  $\tau(t)$  se font sur des données de contrôle et non sur du signal. Ces fonctions sont donc compatibles avec la plupart des langages de programmation. En écrivant ces fonctions dans un langage comme FAUST par exemple on pourrait convertir les modes de contrôle temporel dans d'autres langages de programmation audio différents de Max/MSP. L'avantage serait de faciliter l'intégration de ces fonctions dans d'autres architectures comme des synthétiseurs ou des samplers. On pourrait alors imaginer les modes de tap-tempo de Mono-Replay intégrés à d'autres instruments comme le RolandSP404 ou la MPC de Akai.

Outre la perspective de développer facilement des fonctions de contrôle temporel compatibles avec plusieurs architectures, le formalisme décrit dans ce travail ouvre aussi la perspective de voir ce logiciel sous la forme d'une librairie de fonctions Max/MSP dédiées au contrôle temporel comme suggéré par Basile Chassaing dans l'entretien F. Les "boîtes" intégrant les fonctions de contrôle temporel deviendraient des objets Max/MSP qui calculeraient l'indice temporel correspondant à la position de l'enregistrement à synthétiser pour un mode en particulier. Il suffirait alors pour l'utilisateur de connecter ces boîtes en entrée du moteur audio de leur choix pour lire l'échantillon à la position de lecture indiquée. Pour développer une librairie de la sorte il est essentiel de passer par une écriture unifiée des fonctions qui la composent.

#### **6.2.4 Développer un projet pédagogique autour du contrôle temporel**

Les logiciels développés au sein de PUCE MUSE sont essentiellement utilisés dans le cadre d'ateliers pédagogiques. Ils abordent chacun un concept bien particulier de la création assistée par ordinateur et sont conçus pour être facilement pris en main par des utilisateurs novices. Au fil des années PUCE MUSE a développé une véritable expertise dans la conception d'outils informatiques pour la pédagogie. Récemment, le logiciel Méta-DJ a été installé sur les ordinateurs portables distribués dans les classes de collège du Val de Marne. Ainsi les enseignants peuvent utiliser ces outils en classe et développer des projets pédagogiques autour de la création musicale comme représentés figure 6.1.

Mono-Replay est le premier logiciel développé à PUCE MUSE dans le cadre d'une thèse. L'ambition de départ pour la structure était de créer un outil pour contrôler de manière expressive le déroulement temporel de n'importe quel enregistrement. L'outil devait s'adresser aussi bien à des utilisateurs novices que à des utilisateurs experts. La phase de développement s'est organisée sous la forme d'échanges avec Serge de Laubier afin d'ajouter les fonctionnalités jugées nécessaires pour arriver à ce résultat. Pour le moment l'outil a été pris en main seulement dans le cadre des ateliers décrits sections 4.1.1 et 4.1.2. Seuls des utilisateurs habitués à la manipulation de logiciels pour la création ont



FIGURE 6.1 – Découverte de l'application Méta-DJ développé par PUCE MUSE par des élèves de 6ème du collège Francine Fromond à Fresnes dans le cadre d'un atelier pédagogique.

donc utilisé ce logiciel. Cela a permis d'avoir des retours pointus sur les perspectives d'amélioration du logiciel. Si on observe le profil d'utilisateur habituel des autres logiciels Mono, très peu sont expérimentés dans la manipulation de logiciel pour la création. Il s'agit souvent de collégiens, lycéens et étudiants de conservatoire qui découvrent l'outil informatique pour la création dans le cadre de ces ateliers.

Mono-Replay est un outil adapté pour développer des ateliers autour du concept de division rythmique. Dans les exemples de segmentations abordés dans la section 2.2.1 on décompose la *Gymnopédie n°1* de Érik Satie en événements rythmiques, en pulsations et en mesure. Il serait intéressant d'étendre ce principe à d'autres enregistrements de manière à s'en servir comme support interactif pour un enseignement sur la notion de division rythmique. Ainsi Les élèves pourraient placer la position de la tête de lecture au début d'une noire, d'une blanche, d'une mesure ou d'un mouvement.

En plus de la visée pédagogique une telle utilisation permettrait de continuer le travail d'évaluation quantitative de l'outil au travers des questionnaires conçus par Gabriela Patino-Lakatos disponibles en annexe I. Chaque groupe

d'élèves qui utilisera le logiciel dans le cadre de ces ateliers pourra remplir ce questionnaire.

# Bibliographie

- [1] Blanchet Alain. *L'entretien / Alain Blanchet, Anne Gotman; sous la direction de François de Singly*. «L'»enquête et ses méthodes. Armand Colin, Paris, 2e éd. refondue edition, DL 2007. (Cité en page [151](#).)
- [2] C. Ariza. The dual-analog gamepad as a practical platform for live electronics instrument and interface design. In *NIME*, 2012. (Cité en page [120](#).)
- [3] M. Battier. What the grm brought to music : from musique concrète to acousmatic music. *Organised Sound*, 12, 2007. (Cité en page [16](#).)
- [4] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on speech and audio processing*, 13(5) :1035–1047, 2005. (Cité en page [59](#).)
- [5] J. P. Bello, C. Duxbury, M. Davies, and M. Sandler. On the use of phase and energy for musical onset detection in the complex domain. *IEEE Signal Processing Letters*, 11(6) :553–556, 2004. (Cité en page [59](#).)
- [6] S. Böck and W. Gerhard. Maximum filter vibrato suppression for onset detection. In *Proceedings of the 16th International Conference on Digital Audio Effects (DAFx-13)*, pages 55–61, Maynooth, Irlande, Septembre 2013. (Cité en pages [54](#), [56](#), [90](#), [91](#), [109](#), [137](#), [138](#) et [174](#).)
- [7] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X. Serra. Essentia : An open-source library for sound and music analysis. octobre 2013. (Cité en pages [52](#), [89](#), [90](#), [98](#) et [174](#).)

- [8] R. Bricout. Les incarnations du sampler au xxe siècle : l'avènement du musicien-luthier. In *Electroacoustic Music Studies Network International Conference*, INA-GRM et Université Paris-Sorbonne, France, juin 2008. (Cité en page 16.)
- [9] C. Cadoz and M. M. Wanderley. Gesture - Music. In Ircam-Centre Pompidou Marcelo Wanderley et Marc Battier, editor, *Trends in Gestural Control of Music*. mars 2000. cote interne IRCAM : Cadoz00a. (Cité en pages 14 et 132.)
- [10] A. A. Correya, D. Bogdanov, L. Joglar-Ongay, and X. Serra. Essentia.js : A javascript library for music and audio analysis on the web. In *ISMIR*, 2020. (Cité en page 54.)
- [11] P. Couprie. *La musique électroacoustique : analyse morphologique et représentation analytique*. Theses, Paris-Sorbonne, décembre 2003. (Cité en page 31.)
- [12] P. Couprie. EAnalysis : aide à l'analyse de la musique électroacoustique. In *Journées d'Informatique Musicale*, pages 183–189, Mons, Belgium, mai 2012. (Cité en page 64.)
- [13] P. Couprie. EAnalysis - Sound-Based Music Analysis. Journées d'analyse musicale - Musique savante/musiques actuelles : articulations, décembre 2014. Poster. (Cité en page 45.)
- [14] P. Couprie. Le Méta-Instrument : genèse et évolution d'un nouvel instrument. *Musique, images, instruments*, 17 :230–245, 2018. (Cité en pages 14, 129 et 175.)
- [15] C. D'Alessandro. Révolution (industrielle) dans la facture instrumentale. In *Arts, sciences et techniques. Histoire des arts : actes de l'université de printemps, 26-27 mai 2011, château de Fontainebleau*, pages 101–106. CNDP, juin 2012. (Cité en page 13.)
- [16] C. D'Alessandro, X. Xiao, G. Locqueville, and B. Doval. Borrowed Voices. In *International Conference on New Interfaces for Musical Expression NIME'19*, NIME'19 Proceedings of the International Conference on New Interfaces for Musical Expression, pages 22–24, Porto Alegre, Brazil, juin 2019. (Cité en pages 67, 100 et 174.)



- [17] S. De Laubier, H. Genevois, L. Feugère, G. Bertrand, V. Goudard, S. Le Beux, B. Doval, and C. d'Alessandro. ORJO et la META-MALLETTE 4.0. In Thierry Dutoit, Todor Todoroff, and Nicolas d'Alessandro, editors, *Journées d'Informatique Musicale 2012*, page 227, Mons, Belgium, mai 2012. Université de Mons. (Cité en pages 35 et 84.)
- [18] S. de Laubier and V. Goudard. Meta-instrument 3 : a look over 17 years of practice. pages 288–291, 01 2006. (Cité en page 14.)
- [19] S. De Laubier and V. Goudard. PUCE MUSE - LA MÉTA-MALLETTE. In *Journées d'Informatique Musicale*, Albi, France, mars 2008. (Cité en pages 35 et 84.)
- [20] N. Degara, E. A. Rua, A. Pena, S. Torres-Guijarro, M. E. P. Davies, and M. D. Plumbley. Reliability-informed beat tracking of musical signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1) :290–301, 2012. (Cité en pages 59, 90, 94, 134 et 174.)
- [21] S. Dixon. Onset detection revisited. septembre 2006. (Cité en pages 49, 50, 51 et 54.)
- [22] Chris Duxbury, Juan Pablo Bello, Mike Davies, and Mark Sandler. A combined phase and amplitude based approach to onset detection for audio segmentation. In *Digital Media Processing for Multimedia Interactive Services*, pages 275–280. World Scientific, 2003. (Cité en page 59.)
- [23] G. Essl and S. O'Modhrain. An enactive approach to the design of new tangible musical instruments. *Organised sound*, 11 :285 –296, décembre 2006. (Cité en pages 32 et 37.)
- [24] M. Goto. An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research*, 30(2) :159–171, 2001. (Cité en pages 23 et 44.)
- [25] M. Goto. A chorus section detection method for musical audio signals and its application to a music listening station. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5) :1783–1794, 2006. (Cité en pages 23 et 45.)
- [26] M. Goto. Active music listening interfaces based on signal processing. In *Proceedings of the 2007 IEEE International Conference on Acoustics*,

- Speech, and Signal Processing*, volume 4, pages 1441–1444, Mai 2007. (Cité en pages 23 et 45.)
- [27] M. Goto and Y. Muraoka. A beat tracking system for acoustic signals of music. In *Proceedings of the Second ACM International Conference on Multimedia*, MULTIMEDIA '94, page 365–372, New York, NY, USA, 1994. Association for Computing Machinery. (Cité en page 23.)
  - [28] V. Goudard. John, the semi-conductor : a tool for improvisation. In Sandeep Bhagwati and Jean Bresson, editors, *International Conference on Technologies for Music Notation and Representation (TENOR'18)*, Proceedings of the 4th International Conference on Technologies for Music Notation and Representation, Montréal, Canada, mai 2018. (Cité en page 95.)
  - [29] P. Grosche and M. Muller. A mid-level representation for capturing dominant tempo and pulse information in music recordings. International Society for Music Information Retrieval Conference, 2009. (Cité en pages 55 et 57.)
  - [30] K. F. Hansen and R. Bresin. The skipproof virtual turntable for high-level control of scratching. *Computer Music Journal*, 34(2) :39–50, 2010. (Cité en page 22.)
  - [31] J. Haury. La pianotechnie ou notage des partitions musicales pour une interprétation immédiate sur le métapiano. In *14èmes Journées d'Informatique Musicale*, pages 9–17, Grenoble, avril 2009. ACROE. (Cité en pages 24, 25 et 31.)
  - [32] S. Le Beux, B. Doval, and C. d'Alessandro. Issues and solutions related to real-time TD-PSOLA implementation. In *Convention of the Audio Engineering Society (AES 2010)*, page 6, London, UK, Unknown Region, 2010. (Cité en page 67.)
  - [33] Adrien Lefèvre. LES N-LECTEURS DU GRM PLAYER. In *Journées d'informatique musicale*, Paris, France, May 2017. (Cité en page 158.)
  - [34] G. Locqueville, C. D'Alessandro, S. Delalez, B. Doval, and X. Xiao. Voks : Digital instruments for chironomic control of voice samples. *Speech Communication*, 125 :97 – 113, 2020. (Cité en pages 25, 26, 67, 75, 100 et 174.)

- [35] T. Lucas, C. D'Alessandro, and S. De Laubier. Mono-Replay : a software tool for digitized sound animation. In *NIME 2021 (International Conference on New Interfaces for Musical Expression )*, Shanghai, China, juin 2021. PubPub. (Cité en page 39.)
- [36] T. Lucas, S. De Laubier, and C. D'Alessandro. Animer la musique sur support. In N. Hérold, C. Guichaoua, and T. Mays, editors, *Acte musical et environnements informatiques : actes des Journées d'Informatique Musicale 2020 (JIM 2020)*. Université de Strasbourg/Faculté des Arts, 2020. (Cité en pages 39 et 64.)
- [37] P. Masri. *Computer modelling of sound for transformation and synthesis of musical signals*. PhD thesis, University of Bristol, 1996. (Cité en page 59.)
- [38] T. Mays and F. Faber. A notation system for the karlax controller. In *NIME*, 2014. (Cité en page 14.)
- [39] P. Mion. *L'envers d'une oeuvre : De Natura Sonorum*. 1983. (Cité en page 137.)
- [40] M. Morise, F. Yokomori, and K. Owaza. World : A vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Transactions on Information and Systems*, E99.D(7) :1877–1884, 2016. (Cité en page 26.)
- [41] F. Nicolas. Les transitoires de cordes dans mosaïc (devenu modalys). In *Ircam*, mars 1991. (Cité en page 48.)
- [42] J. Oswald. Plunderphonics, or audio piracy as a compositional prerogative. In *The wired society electro-acoustic conference*, Toronto, 1985. (Cité en page 19.)
- [43] G. Patiño-Lakatos. Expérience à l'Institut Gustave Baguer d'Asnières (2012-2014). Research report, Agence Nationale de la Recherche ; Université Paris 8 ; Puce Muse - Centre de Création de Musique Visuelle, octobre 2015. (Cité en pages 35, 84, 121 et 149.)
- [44] G. Patiño-Lakatos. Jeu interactif avec des orchestres d'instruments réels numérisés. Research report, Région Ile de France ; Institut Jean le Rond

d’Alembert ; Puce Muse - Centre de création de musique visuelle, April 2017. (Cité en page 149.)

- [45] G. Patiño-Lakatos. Image numérique interactive, articulation intersémiotique et construction subjective : une expérience pédagogique de visualisation du geste sonore. *Cliopsy*, Avril 2018. (Cité en page 149.)
- [46] G. Patiño-Lakatos, H. Genevois, and B. Navarret. De la sensation a la sémiotique vibrotactile. Médiations pour l’expérience musicale. *Hybrid. Revue des arts et médiations humaines*, July 2020. (Cité en page 149.)
- [47] Gabriela Patiño-Lakatos. Expérience avec les Méta-Touches au Lycée Toulouse-Lautrec. Research report, Agence Nationale de la Recherche ; Université Paris 8 - Saint-Denis ; Puce Muse - Centre de Création de Musique Visuelle, October 2015. (Cité en page 149.)
- [48] G. Peeters. A large set of audio features for sound description (similarity and classification) in the cuidado project. Ircam, Analysis/Synthesis Team, Paris, France, avril 2003. (Cité en pages 23 et 47.)
- [49] DJ Q-bert. Scratchlopedia breaktannica : 100 secret scratches. Technical report, DVD : SECRT001-DVD, 2007. (Cité en page 20.)
- [50] A. Roebel. A new approach to transient processing in the phase vocoder. In *Proceeding of the 6th International Conference on Digital Audio Effects*, septembre 2003. (Cité en pages 44, 88 et 110.)
- [51] A. Roebel and X. Rodet. Efficient Spectral Envelope Estimation and its application to pitch shifting and envelope preservation. In *International Conference on Digital Audio Effects*, pages 30–35, Madrid, Espagne, September 2005. cote interne IRCAM : Roebel05b. (Cité en page 112.)
- [52] A. Sakran, A. Sherif, and S. Eldeen. A review : Automatic speech segmentation. *International Journal of Computer Science and Mobile Computing*, 6, avril 2017. (Cité en page 63.)
- [53] N. Schnell. *Playing (with) Sound Of the Animation of Digitized Sounds and their Reenactment by Playful Scenarios in the Design of Interactive Audio Applications*. PhD thesis, University of Music and Performing Arts Graz, Autriche, octobre 2013. (Cité en pages 33, 42, 43 et 173.)

- [54] D. Smalley. Spectromorphology : explaining sound-shapes. *Organised Sound*, 2(2) :107–126, 1997. (Cité en pages 28, 37, 130 et 171.)
- [55] F. J. Varela, E. T Thompson, and E. Rosch. The embodied mind : Cognitive science and human experience. *MIT Press*, 1992. (Cité en page 32.)
- [56] M. M. Wanderley and P. Depalle. Contrôle gestuel de la synthèse sonore. In Hermès Science Publications, editor, *Interfaces Homme-Machine et Création Musicale*,, pages 145–164. Décembre 1999. cote interne IRCAM : Wanderley99c. (Cité en page 14.)
- [57] X. Xiao, G. Locqueville, C. D’Alessandro, and B. Doval. T-Voks : the Singing and Speaking Theremin. In M. Queiroz and A. Xambó, editors, *NIME 2019 International Conference on New Interfaces for Musical Expression*, Proceedings of the International Conference on New Interfaces for Musical Expression, pages 110–115, Porto Alegre, Brazil, juin 2019. UFRGS. (Cité en pages 25, 38, 44, 67, 77, 100, 174 et 175.)
- [58] J. Zapata, A. Holzapfel, M. Davies, J. Oliveira, and F. Gouyon. Assigning a confidence threshold on automatic beat annotation in large datasets. *Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012*, 01 2012. (Cité en page 90.)
- [59] J. R. Zapata, M. E. P. Davies, and E. Gómez. Multi-feature beat tracking. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4) :816–825, 2014. (Cité en pages 52, 59, 61, 62, 63, 72, 90, 94, 103, 107, 108, 109, 124, 126, 134, 139 et 174.)



# Annexes





Annexe A

Article de la conférence Journées  
d'Informatique Musicale 2020

## Animer la musique sur support

Thomas Lucas, Serge de Laubier, Christophe d'Alessandro

► To cite this version:

Thomas Lucas, Serge de Laubier, Christophe d'Alessandro. Animer la musique sur support. Hérold, N.; Guichaoua, C.; Mays, T. Acte musical et environnements informatiques : actes des Journées d'Informatique Musicale 2020 (JIM 2020), Université de Strasbourg/Faculté des Arts, A paraître. hal-03010761

**HAL Id: hal-03010761**

**<https://hal.archives-ouvertes.fr/hal-03010761>**

Submitted on 17 Nov 2020

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

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

## ANIMER LA MUSIQUE SUR SUPPORT

*Thomas Lucas*  
Puce Muse, Rungis  
Institut Jean Le Rond d'Alembert  
Sorbonne Université, CNRS, Paris  
thomas.lucas@puce-muse.com

*Serge de Laubier*  
Puce Muse, Rungis  
sdl@puce-muse.com

*Christophe d'Alessandro*  
Institut Jean Le Rond d'Alembert  
Sorbonne Université, CNRS, Paris  
christophe.dalessandro@  
sorbonne-universite.fr

### RÉSUMÉ

Cet article présente les principes, ainsi qu'un outil logiciel, pour « animer » de façon expressive les musiques sur support – c'est-à-dire dont le déroulement et le contenu musical sont a priori fixés. Deux étapes sont nécessaires pour cela : 1) l'analyse préalable de l'enregistrement pour la mise en place d'ancres temporelles (notes, pulsations, événements) ; 2) l'utilisation de l'analyse-synthèse et d'interfaces de contrôle pour manipuler le déroulement temporel de la pièce en temps réel. La première étape d'analyse et de marquage consiste à élaborer différentes collections d'ancres, liées à une représentation gestuelle explicite ou sous-jacente du contenu sonore. La seconde étape permet à l'utilisateur musicien de moduler le déroulement temporel et d'autres aspects du son grâce aux ancres, à l'aide d'une cellule de captation du geste et d'un module de transformation audio. Ce propos est illustré par deux exemples très différents : la *Gymnopédie n°1* pour piano d'Erik Satie et « Accidents/harmoniques », extrait de *De Natura Sonorum* de Bernard Parmegiani. Nous discuterons leur rôle dans la conception d'une interaction expressive des différences au niveau de l'indexation du fichier audio et du geste expressif utilisé pour « animer » chacun de ces exemples.

### 1. INTRODUCTION ET ÉTAT DE L'ART

#### 1.1. Animer un son

L'expression « animer un son » désigne, de façon métaphorique, l'interaction d'un utilisateur avec une application sonore interactive qui permet de diffuser des musiques enregistrées [21]. Le degré zéro de l'animation, à savoir le geste minimal, consiste à lancer et à arrêter la diffusion de l'enregistrement. On peut aussi considérer le son enregistré sur un support comme une matériau inanimé qui peut être en quelque sorte (ré)animé et réincarné lors d'une performance. Animer un son désigne alors le processus qui articule l'automation – au sens des logiciels audionumériques – et l'interaction, afin d'agir de façon expressive sur le déroulement de l'enregistrement et de varier une matière sonore a priori fixée et toujours identique. Le terme

« animer » souligne les aspects expressifs de l'interaction : le geste utilisé doit permettre d'exprimer au mieux des intentions musicales lors de la performance.

Notre travail repose sur l'hypothèse qu'il est possible d'associer des formes ou des mouvements sous-jacents au contenu d'un enregistrement sonore. À l'écoute d'une œuvre, un auditeur associe perceptivement des substituts gestuels, ou « *gesture surrogacy* » [22]. Dans la *Gymnopédie n°1* de Satie, cette représentation découle de la partition et fait directement référence aux gestes physiques de production du son par le pianiste : il s'agit d'un substitut gestuel de deuxième ordre [22]. Dans « Accidents/harmoniques » de Bernard Parmegiani, une représentation gestuelle directe semble plus difficile à établir. Le continuum harmonique présent au début de cette œuvre électroacoustique n'évoque pas de geste instrumental et sa perception repose davantage sur les évolutions spectromorphologiques du matériau sonore : il s'agit d'un substitut gestuel éloigné [22].

Le but de cette recherche est d'élaborer un instrument sous la forme d'un outil logiciel permettant de contrôler de façon expressive le déroulement temporel d'une musique sur support. La musique est diffusée en manipulant à l'aide de capteurs une représentation gestuelle sous-jacente élaborée à partir du son perçu. La mise en œuvre d'un tel projet implique deux étapes principales.

La première étape consiste à analyser le contenu audio, afin de placer des ancres sur des éléments temporels significatifs directement liés au discours musical. Divers critères permettent de placer ces ancres. Nous examinerons ainsi en détail deux méthodes automatiques pour placer des index sur les temps ou les notes d'un morceau.

La seconde étape est celle de la définition de l'instrument, au sens des capteurs et de l'interaction. Il s'agit de définir les relations synergiques entre les pôles suivants : capteurs et analyse du geste ; analyse audio et détermination d'une collection d'index ; moteur audio et actions sur le son. Avant de décrire nos propres travaux, il est important de mentionner les travaux antérieurs visant à animer les musiques sur support fixe.

## 1.2. Musique concrète, *plunderphonics*, *turntablism* et *reenactment*

L'utilisation de supports fixes en tant que moyens pour créer de la musique a été désignée par l'expression « musique concrète » et formalisée par Pierre Schaeffer dans les années 1950 [20]. On parle de « *plunderphonics* » [19] – littéralement piller (*to plunder*) le son (*phonics*) – lorsque les segments audio appartenant à une œuvre musicale sur support clairement identifiable sont repris pour créer de nouvelles compositions.

Ces approches visent à créer de la musique à partir de musiques fixées sur support. Dans les années 1970, une approche portant sur l'animation de musiques sur support a été développée afin de rendre plus vivante et interactive la musique de dance par les disc jockeys. Les premiers instruments des DJs ont été la table de mixage et la platine tourne-disque, d'où le nom de *turntablism* (ou platinisme). Les mécanismes perceptifs mis en place par un utilisateur lors de son interaction avec une application audio interactive sur un enregistrement sonore ont été conceptualisés sous le nom de *reenactment* [21]. Fondé sur les théories énonciatives de Francisco Varela [23], ce concept utilisé par Norbert Schnell [21] pour décrire cette façon de faire de la musique implique les dimensions perceptive et cognitive de l'interaction avec un enregistrement sonore, en rapport avec les actions évoquées par le son même. Afin d'intégrer au mieux la dimension expressive de l'interaction, le concept de *reenactment* se place au centre de notre démarche, au travers de plusieurs questions : quel serait le geste le plus approprié pour interagir de façon expressive avec un son enregistré ? S'agit-il du geste physique évoqué à l'écoute de l'enregistrement ? Comment repérer et marquer dans le flux sonore ce geste sous-jacent ou évoqué ?

## 1.3. Écoute active et analyse du contenu

Une démarche apparentée au *reenactment*, proposée par Masataka Goto, met en œuvre des techniques d'analyse dites de MIR (*Music Information Retrieval*) pour collecter des informations intrinsèques au discours musical, comme un changement de grille harmonique [12]. Le contenu d'un fichier audio est utilisé pour concevoir des interactions nouvelles par un processus d'écoute active – *Active Music Listening Interface* [11]. Grâce à ces applications logicielles, le rôle de l'auditeur se rapproche de celui de l'interprète, grâce aux possibilités offertes pour dérouler la lecture du fichier audio.

Cette démarche d'analyse du contenu est également utilisée pour l'analyse musicale. En s'interrogeant sur l'évolution esthétique du morceau improvisé « *Interstellar Overdrive* » de Pink Floyd au cours de différentes interprétations, Philippe Gonin [10] observe l'évolution du flux spectral pour mettre en évidence la structure musicale du morceau. L'outil d'analyse utilisé place des ancres sur les variations brusques de tonalité, permettant ainsi d'observer la structure globale du

morceau enregistré sans passer par une transcription écrite détaillée. L'accès à une telle information pour notre étude permettrait de placer des index sur des éléments significatifs du discours musical à un niveau macro, directement à partir de l'information contenue dans l'enregistrement musical.

## 1.4. Méta-Piano, Méta-Instrument et synthèse vocale performative

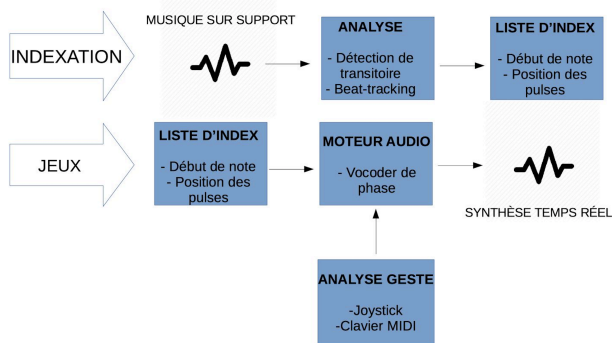
Avec l'avènement de l'audio numérique en temps réel, les rapports entre la musique enregistrée sur divers supports et les interfaces de contrôle sont devenus particulièrement riches et ductiles. Pour les musiques notées sous forme de partition numérique, le Méta-Piano de Jean Haury [13] est une source d'inspiration pour notre démarche. Le Méta-Piano permet à un utilisateur musicien d'interpréter une pièce au piano numérique à l'aide d'un clavier MIDI réduit à quelques touches. Les quelques touches ne contrôlent pas la mélodie, comme au piano, mais le séquençement – rythme, articulation, phrasé, tempo et accentuation – des événements rythmiques sélectionnés, repérés et marqués à partir de la partition. L'utilisateur musicien propose sa propre interprétation par le contrôle de méta-paramètres du discours musical, et non par la production directe de tous les détails du son, comme pour un instrument musical ordinaire. En ce sens, il s'agit d'un méta-instrument. Le Méta-Piano permet de se concentrer entièrement sur les paramètres expressifs d'une interprétation au clavier, en utilisant l'enregistrement comme matériau et comme raccourci permettant de réduire la technique instrumentale proprement dite. Le choix de l'interface de type clavier MIDI pour régir cette application sonore permet à l'utilisateur d'exprimer en temps réel le geste physique évoqué par le son qu'il entend. Même si le morceau de musique n'est pas directement issu d'un enregistrement sonore, mais d'un fichier MIDI, l'interaction proposée avec celui-ci est directement liée au geste physique évoqué par le son. Dans le cas de cet instrument, le clavier MIDI réduit est l'interface la plus appropriée pour contrôler l'instrument de façon expressive. Qu'en est-il des musiques dont les sons qui la constituent n'évoquent pas le geste physique à l'origine de celui-ci ? Serait-il intéressant d'interagir avec leur déroulement grâce à un clavier MIDI ? Nous tenterons d'apporter quelques éléments de réponse dans notre travail.

Une méthodologie semblable à celle du Méta-Piano a été développée pour les Méta-Instruments de Serge de Laubier [4]. Le Méta-Instrument propose une interface beaucoup plus riche que le clavier MIDI, avec plusieurs dizaines de capteurs continus pour les bras, les mains et les doigts. Le module de synthèse est également riche et varié. Un mode d'utilisation, parmi d'autres, du Méta-Instrument est l'interprétation d'un texte musical enrichi de marques temporelles. Les fichiers MIDI sont joués en contrôlant non pas chaque note, mais des blocs temporels qui portent spécifiquement l'expression, et

qui sont définis lors d'une phase d'analyse préalable du contenu musical.

L'animation de la synthèse vocale – ou synthèse vocale performative – a été développée depuis le projet « *Speech conductor* » de Christophe d'Alessandro [6]. La synthèse par blocs temporels à l'aide d'ancres a été proposée dans les systèmes Vokinesis [7, 8] puis Voks [5, 24]. Dans ce cas, ce ne sont pas des fichiers MIDI qui sont joués, mais des échantillons de voix enregistrée. Grâce à un vocodeur, il est possible de changer des paramètres tels que l'intonation, la qualité vocale et la force vocale, et ainsi de jouer avec une voix synthétisée à partir de l'analyse d'une voix enregistrée. Des ancres temporelles permettent de séquencer les syllabes – avec deux points de contrôle rythmique par syllabe –, ce qui permet un jeu musical fluide avec un apprentissage rapide pour l'utilisateur musicien. Dans le cadre de la synthèse vocale, deux points de contrôle rythmique semblent nécessaires, le déclenchement et le relâchement étant également importants pour le jeu musical.

Notre système comprend donc deux grands aspects fonctionnels : la préparation, l'analyse et le marquage ou indexation du contenu de l'enregistrement, puis la définition de l'instrument en tant qu'assemblage d'interfaces pour manipuler le contenu indexé (figure 1).



**Figure 1.** Principes de MonoReplay : phase d'indexation et phase de jeu.

## 2. ANALYSE ET INDEXATION DU CONTENU SONORE

La première étape de notre travail d'ancrage consiste à segmenter le flux audio que l'on souhaite rejouer en blocs temporels susceptibles de porter le contrôle de l'animation du fichier audio. Cette segmentation peut se faire selon plusieurs échelles temporelles : les notes jouées, les temps, la mesure ou encore les phrases musicales. Cette phase d'analyse vise à placer dans le flux temporel des points – ou ancres – afin de repérer les phases ou blocs temporels que l'on souhaite manipuler.

Pour contrôler de façon expressive le déroulement d'une musique enregistrée, il est indispensable d'obtenir préalablement des informations significatives sur les aspects temporels du discours musical pour ensuite les manipuler. Le tempo, le rythme, la position des pulsations ou encore la position temporelle des notes jouées dans un enregistrement sont des informations que

l'on récupère manuellement ou à l'aide d'algorithmes de traitement du signal. L'automatisation – ou pas – de cette étape d'indexation n'a vraisemblablement pas d'impact direct sur le résultat au moment du jeu – c'est-à-dire pour rejouer de façon expressive un enregistrement de musique –, mais permet de construire plus rapidement le répertoire musical utilisé.

### 2.1. Indexation de la *Gymnopédie n° 1* de Satie

La *Gymnopédie n° 1* de Satie est une célèbre pièce pour piano solo. À l'aide d'outils de traitement du signal que nous détaillons plus loin, nous élaborons trois collections d'index temporels enregistrés dans un fichier texte : 1) indexation à la mesure, avec la position temporelle du premier temps ; 2) indexation à la pulsation ; 3) indexation à l'événement, avec la position temporelle de chaque début de note ou groupe de notes jouées ensemble.

### 2.2. Indexation d'« Accidents/harmoniques »

« Accidents/harmoniques » est le deuxième mouvement de *De Natura Sonorum*, une pièce sur bande de Parmegiani. Ce mouvement ne possède pas de structure métrique identifiable ; c'est pourquoi nous ne cherchons pas à déterminer ici de pulsation ou de tempo. À l'aide d'outils de traitement du signal que nous détaillons plus loin, nous élaborons une collection d'index par événements : la position temporelle de chaque début d'événement sonore – qui prend le statut de note – ou de groupes d'événements sonores – c'est-à-dire des groupes de notes – possédant des transitoires d'attaque suffisamment brefs et prononcés sur le plan perceptif – est stockée dans un fichier texte. Pour cette œuvre particulière, une partition visuelle a posteriori a été élaborée [17], ce qui est précieux pour vérifier le placement des index.

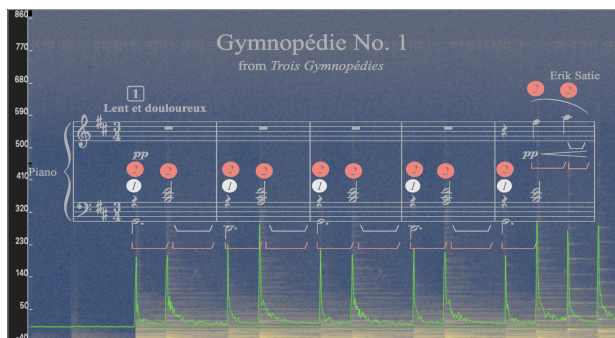
### 2.3. Traitement du signal pour l'indexation

#### 2.3.1. Détection des transitoires d'attaque

La recherche en indexation d'information musicale (*Music Information Retrieval*) a abouti aujourd'hui à des outils d'analyse de contenu audio extrêmement efficaces. En repérant les valeurs au dessus d'un niveau seuil sur la courbe de nouveauté, l'algorithme SuperFlux [9] permet de détecter les transitoires de sons percussifs, ainsi que les transitoires plus doux marqués par un changement de fréquence fondamentale et non par une courte tranche de bruit blanc.

Le flux spectral est la somme de la différence, bande par bande, de l'amplitude du spectre obtenue par transformée de Fourier à court terme. Les maxima observés sur la courbe représentative du flux spectral – appelée aussi courbe de nouveauté – indiquent les brusques variations spectrales dues à la présence d'un nouvel événement, que nous associons à une nouvelle note dans l'enregistrement analysé. En récupérant les

positions temporelles de ces maxima, on obtient une collection d'index où chaque événement important présent dans l'enregistrement est associé à sa position temporelle (figure 2).



**Figure 2.** Détermination de la courbe de nouveauté (en vert) dans le cas de la *Gymnopédie n° 1*, avec indexation à la mesure en blanc (1) et indexation à l'événement en rouge (2).

### 2.3.2. Suivi de pulsation

La pulsation peut être définie comme « l'impulsion périodique induite perceptivement décrivant le mieux l'action de taper du pied sur la musique »<sup>1</sup>. Lorsque l'on écoute une musique possédant une pulsation, il nous arrive très fréquemment d'effectuer des mouvements synchronisés avec le tempo du morceau ; c'est pourquoi nous avons choisi d'intégrer une collection d'index représentant cette pulsation lorsqu'elle existe, afin de pouvoir contrôler le déroulement d'un flux audio en tapant la pulsation. Pour constituer cette collection d'index, nous utilisons l'algorithme de détection de pulsation *Multi-feature Beat Tracking* [25]. Cet algorithme, disponible en Python dans le framework *Essentia* [1], détermine neuf fonctions de détection de transitoire d'attaque selon différentes méthodes, afin de ne garder que celle qui présente le plus de similarité avec les autres. Dans le but d'utiliser notre instrument pour interagir avec des enregistrements très différents les uns des autres, l'algorithme *Multi-feature Beat Tracking* est bien adapté, car il offre diverses méthodes de détection de transitoires, ce qui permet d'aborder la variété du paysage sonore des musiques sur support. Pour un son de corde pincée, les transitoires ne présenteront pas les mêmes propriétés spectrales que pour un son de corde frottée. On applique une fonction d'autocorrélation pour chaque fonction de détection de transitoire pour mettre en évidence les périodicités du signal. Une méthode basée sur des matrices de similarité [14] compare chaque séquence obtenue et sélectionne celle présentant le plus de similitude avec les autres comme la plus représentative de la pulsation de l'enregistrement. Cette méthode comparée à seize autres systèmes de détermination de pulsation présente les meilleurs résultats [14].

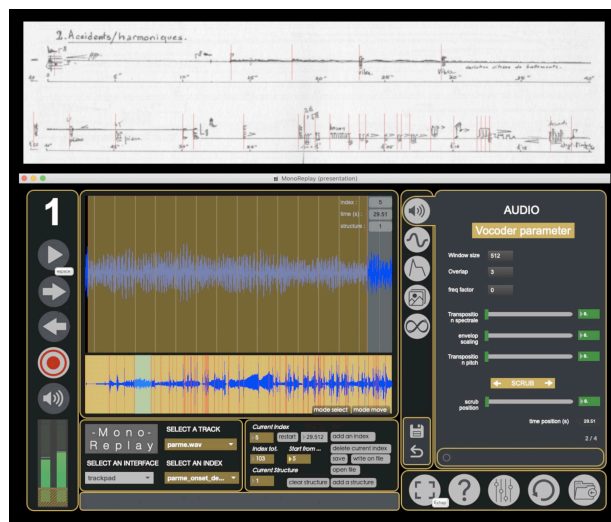
<sup>1</sup> « The beat (or tactus) is a perceptually induced periodic pulse that is best described by the action of "foot-tapping" to the music » [15].

Une fois la collection d'index à la pulsation déterminée, on applique à la main une correction en tenant compte de la partition pour ne garder que les premiers temps de chaque mesure. On obtient ainsi une collection supplémentaire constituée de chaque début de mesure du morceau (figure 2).

## 3. DÉFINITION DE L'INSTRUMENT

Au même titre que d'autres applications audio interactives [21], l'instrument *MonoReplay* s'articule autour de trois pôles : un pôle d'analyse et d'indexation, un pôle de captation et d'analyse du geste, et un pôle de synthèse sonore.

Ces trois pôles doivent fonctionner en synergie, afin d'effectuer le travail d'indexation décrit dans la partie précédente et de proposer des interactions pour agir en temps réel sur le contenu préalablement analysé à l'aide d'un geste musical enregistré par différents capteurs (figure 3).



**Figure 3.** Interface de *MonoReplay* en phase de jeu pour un extrait d'« Accidents/Harmoniques ». En haut : partition reconstituée, avec les index placés pour *MonoReplay*. En bas : oscillogrammes avec les index de jeu.

### 3.1. *Gymnopédie n° 1* de Satie

À l'écoute de la *Gymnopédie n° 1*, il est relativement évident de saisir la relation entre le rendu sonore et sa causalité instrumentale : le pianiste joue les touches de son instrument et le son produit par le contact du marteau sur les cordes est enregistré. Le geste physique du pianiste entre dans notre perception au moment de l'écoute ; c'est pourquoi nous avons choisi d'utiliser comme interface un clavier MIDI pour contrôler le déroulement de cet enregistrement. En appuyant sur une touche du clavier, l'utilisateur musicien déclenche la lecture d'un segment audio du morceau. Plus il reste appuyé longtemps, plus la vitesse de lecture ralentit. Il peut passer au segment suivant en appuyant sur une autre touche du clavier et ainsi superposer la lecture de

la fin d'un segment avec la lecture du début d'un autre. L'utilisateur peut alors contrôler le rythme, le phrasé et le tempo de l'enregistrement au moment de son déroulement. Ce fonctionnement est analogue à celui du Méta-Piano dans son principe, sauf pour ce qui est de la production du son.

### 3.2. « Accident/harmonique » de Parmegiani

Selon Denis Smalley [22], l'écoute d'une musique enregistrée permet d'accéder à une certaine compréhension de celle-ci si nous saisissons la relation entre un corps sonore évoqué et sa cause. Dans le cas des musiques instrumentales par exemple, l'expérience d'écoute d'une performance instrumentale fixée sur support met en jeu un processus de conditionnement culturel. Ainsi, une sorte de connaissance implicite des gestes instrumentaux réalisés au moment de l'enregistrement – intrinsèquement liée à la connaissance de la musique en tant qu'activité du sujet – est mise en jeu. Dans notre étude nous souhaitons identifier les interfaces les mieux adaptées pour contrôler de façon expressive le déroulement de nos enregistrements. Nous faisons l'hypothèse que l'intégration de gestes physiques liés au substitut gestuel, virtuels lors de la création musicale et/ou évoqués lors de l'écoute des enregistrements, sont une clé pour la conception de l'interaction. Nous proposons alors à l'utilisateur deux interfaces pour interagir avec le contenu musical : un joystick permettant de contrôler la position de lecture du fichier et un clavier MIDI permettant de contrôler la vitesse de lecture et la position, comme par exemple le geste au clavier.

En effectuant un mouvement circulaire à l'aide d'un joystick, l'utilisateur musicien contrôle la position de lecture au sein d'un même segment. Pour passer au segment suivant, il suffit de faire un tour complet. Ce geste se rapproche de celui du platinière qui, en contrôlant la vitesse de rotation de ses platines ou en positionnant la tête de lecture sur une position temporelle précise du morceau, interagit de façon expressive avec le déroulement du fichier audio. Pour animer l'enregistrement d'une pièce telle que « Accidents/harmoniques », la complexité de la construction musicale et la multitude de sons utilisés rend difficile la détermination d'un seul geste expressif pour jouer au mieux l'ensemble de la pièce. La plupart des structures sonores perçues dans cette composition évoquent des substituts gestuels de troisième ordre, en suivant la typologie de Smalley [22]. Le geste physique de production du son n'est pas clairement identifiable, mais peut être néanmoins déduit ou reconstruit : lors de l'écoute, le geste évoqué peut être reconstruit en fonction de l'évolution spectromorphologique du son, davantage que par une causalité qui est de fait absente. Le choix du joystick permet d'obtenir un contrôle continu précis du déroulement sur les longues nappes du continuum harmonique entrecoupé de brefs sons percussifs au début du mouvement. Les sons percussifs délimitent la segmentation du morceau et le joystick

permet de naviguer en détail dans les textures de ce continuum.

### 3.3. Traitement du signal pour la transformation

À la différence d'autres applications interactives comme le Méta-Piano par exemple, ce n'est pas un fichier MIDI ou un module de synthèse qui est à la source du son, mais bien un enregistrement que l'on transforme. Ce module de transformation est enrichi de marques correspondant à la fois à la partition et au contenu sonore réel de l'enregistrement : il ne s'agit pas de synthèse mais d'analyse-synthèse, ce qui implique un processus de traitement du signal. Pour cela, un système d'analyse-synthèse est nécessaire. Même s'il ne s'agit pas de voix, ce genre de système porte le nom de *vocodeur*<sup>2</sup>.

SuperVP et groove~ sont deux vocodeurs de phase implémentés sous la forme de *Max externals* paramétrables, permettant d'avoir un contrôle précis sur la position ou la vitesse de lecture d'un segment audio, tout en gardant de très hautes définitions au niveau des textures sonores rejouées. Le SuperVP est utilisé pour contrôler la position de déroulement du fichier audio dans le mode « *Scrub* », alors que l'objet groove~ est utilisé dans le mode « *Speed* » pour contrôler la vitesse de lecture du segment.

## 4. DISCUSSION ET CONCLUSION

Ce travail porte sur les problématiques liées à la question de l'animation d'une musique sur support à travers la segmentation et le contrôle du déroulement temporel, appliqués ici à deux œuvres très différentes.

En récupérant des informations sur le contenu intrinsèque du discours musical comme la pulsation, le tempo et le début des notes, on construit des collections d'index grâce auxquelles on définit des interactions permettant à un utilisateur musicien de contrôler de façon précise le déroulement de ces deux œuvres, compte tenu de leurs spécificités musicales. Le choix des interfaces utilisées s'inspire de deux points de vue complémentaires : d'un côté, on tient compte du substitut gestuel de second ordre – évoquée à l'écoute du morceau de Satie, à savoir le geste du pianiste – et, d'un autre côté, on propose une interaction inspirée du mouvement circulaire de certains platinières contrôlant la vitesse de lecture et la position d'une musique sur support discutée dans le cas de la pièce de Parmegiani. Dans le second cas, le substitut gestuel n'est pas lié directement à la causalité du son entendu, mais permet à l'auditeur-musicien de se focaliser sur les aspects spectromorphologiques de celui-ci, notamment dans le continuum harmonique situé au début du mouvement.

Les musiques écrites sous forme de partition offrent des représentations très explicites à propos du contenu musical, ce qui facilite grandement le travail d'indexation préalable des éléments significatifs du

<sup>2</sup> De *voice coder*, qui signifie codeur de voix.



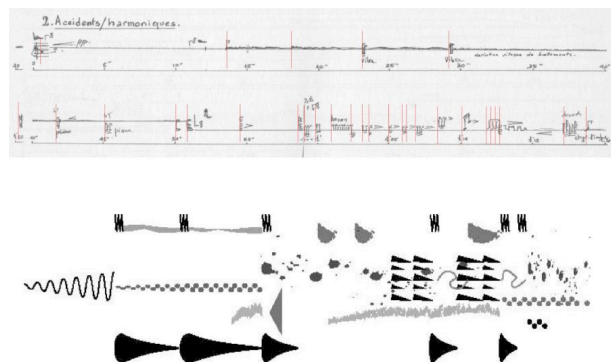
discours musical, notamment à l'étape de correction où l'on vérifie que chaque index déduit par analyse algorithmique correspond à des paramètres du discours musical. Pour les musiques qui ne sont pas représentées sous forme de partition, il est plus difficile de tenir compte d'une structure explicite de la pièce au moment de l'indexation des éléments significatifs du discours musical. Il faut donc passer par une phase d'analyse préalable du contenu, voire de la structure de la pièce, phase dont la partition permet en partie l'économie. Une partition a posteriori peut ainsi être élaborée [17]. Les méthodes de représentation iconiques, par exemple celles élaborées par Pierre Couprie [3], constituent des outils efficaces pour représenter le contenu et la structure de musiques électroacoustiques sous forme graphique. En effectuant ce travail d'analyse préalable, on peut mettre en évidence les éléments saillants d'un enregistrement, effectuer une segmentation fine des structures sonores de base et proposer une représentation intégrant des éléments d'analyse comme la texture des sons, leurs enveloppes temporelles ou encore les hauteurs spectrales.

Pour l'instant, nous affichons seulement la forme d'onde et les index placés sur le fichier audio dans notre interface graphique (figure 4). Cette représentation est assez limitée en termes d'information musicale et ne permet pas d'anticiper la nature du prochain son qui sera joué. Il serait nécessaire d'intégrer une représentation plus musicale, comme par exemple une partition qui défilerait de façon synchronisée avec le son. Pour les musiques sans partition, nous pensons intégrer une représentation iconique qui défilerait elle aussi de façon synchronisée avec le son.

Les pistes de réflexion abordées dans nos propositions d'interfaces nous amènent à constater que, pour chaque son d'un enregistrement que l'on souhaite rejouer, il serait intéressant de tenir compte du substitut gestuel évoqué dans la conception de l'interaction. Cela suppose que pour un morceau de musique électronique ou électroacoustique comportant une richesse sonore aussi importante que « Accidents/harmoniques », il ne faudrait pas se limiter à un seul type de geste pour contrôler le déroulement de l'ensemble de la pièce : il faudrait au contraire un ensemble de gestes correspondant à chacun des substituts gestuels rencontrés lors de l'écoute de l'enregistrement. Les nombreux capteurs du Méta-Instrument 4 en font un outil bien adapté pour ce projet. En concevant des interactions autour des quatre types de capteurs présents sur l'instrument, on peut imaginer une multitude de mappings gestuels grâce auxquels on articulerait le passage d'un index sonore en tenant compte de la spectromorphologie des substituts gestuels évoqués par le son enregistré.

## 5. REMERCIEMENTS

Ce travail a bénéficié du soutien financier de la région Île-de-France et de l'Union Européenne grâce au projet SMAC (FEDER IF0011085).



**Figure 4.** Deux représentations a posteriori de musique sur support : partition graphique d'« Accidents/Harmoniques » par Philippe Mion [17] (en haut) et représentation iconique de *Reflet* de Ivo Malec (0'00" à 1'00") par Pierre Couprie [3].

## 6. RÉFÉRENCES

- [1] Bogdanov, D., et coll. « ESSENTIA : An Open-Source Library for Sound and Music Analysis », Proceedings of the Association for Computer Machinery International Conference on Multimedia, Barcelone, Espagne, 2009.
- [2] Charles, J. F. « A Tutorial on Spectral Sound Processing using Max/MSP and Jitter », *Computer Music Journal* 32/3 (2008), p. 87-102.
- [3] Couprie, P. « Graphical Representation : An Analytical and Publication Tool for Electroacoustic Music », *Organised Sound* 9/1 (2014), p. 109-113.
- [4] Couprie, P. « Le Méta-Instrument : genèse et évolution d'un nouvel instrument », *Musique-Images-Instruments* 17 (2018), p. 231-245.
- [5] D'Alessandro, C., et coll. « Borrowed Voices », Proceedings of the International Conference on New Interfaces for Musical Expression, Porto Alegre, Brésil, 2019.
- [6] D'Alessandro, C., et coll. « Les instruments chanteurs », *Acoustique et techniques* 89 (2019), p. 36-43.
- [7] Delalez, S., D'Alessandro, C. « Vokinesis : Syllabic Control Points for Performative Singing Synthesis », Proceedings of the International Conference on New Interfaces for Musical Expression, Université Aalborg de Copenhague, Danemark, 2017, p. 198-203.
- [8] Delalez, S., D'Alessandro, C. « Adjusting the Frame : Biphasic Performative Control of Speech Rhythm », Proceedings of the Interspeech Conference, Stockholm, Suède, 2017, p. 864-868.
- [9] Gerhard, W., Böck, S. « Maximum Filter Vibrato Suppression for Onset Detection », Proceedings of the 16th Digital Audio FX Conference, Maynooth, Irlande, 2013.



Annexe B

Article de la conférence New  
Interfaces for Musical Expression  
2021

**International Conference on New Interfaces for Musical Expression**

# **Mono-Replay : a software tool for digitized sound animation**

**Thomas LUCAS<sup>1</sup>, Christophe d'Alessandro<sup>2</sup>, Serge de Laubier<sup>3</sup>**

<sup>1</sup>Puce Muse & LAM \* Lutherie-Acoustique-Musique Institut Jean le Rond d'Alembert Sorbonne Université,

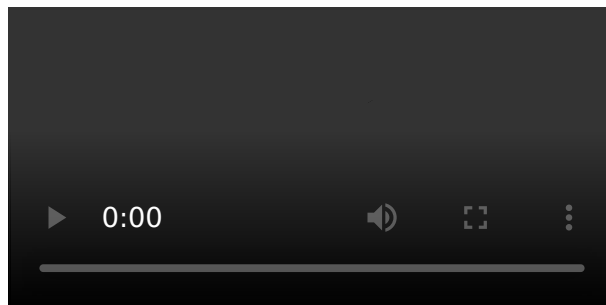
<sup>2</sup>LAM \* Lutherie-Acoustique-Musique Institut Jean le Rond d'Alembert Sorbonne Université,

<sup>3</sup>Puce Muse

**License:** [Creative Commons Attribution 4.0 International License \(CC-BY 4.0\)](https://creativecommons.org/licenses/by/4.0/)

## ABSTRACT

This article describes *Mono-Replay*, a software environment designed for sound animation. "Sound animation" in this context means musical performance based on various modes of replay and transformation of all kinds of recorded music samples. Sound animation using *Mono-Replay* is a two-step process, including an off-line analysis phase and on-line performance or synthesis phase. The analysis phase proceeds with time segmentation, and the set up of anchor points corresponding to temporal musical discourse parameters (notes, pulses, events). This allows, at the performance phase, for control of timing, playback position, playback speed, and a variety of spectral effects, with the help of gesture interfaces. Animation principles and software features of *Mono-Replay* are described. Two examples of sound animation based on beat tracking and transient detection algorithms are presented (a multi-track record of *Superstition* by Steve Wonder and Jeff Beck and *Accidents/Harmoniques*, an electroacoustic piece by Bernard Parmegiani). With the help of these two contrasted examples, the fundamental principles of "sound animation" are reviewed: parameters of musical discourse, audio file segmentation, gestural control and interaction for animation at the performance stage.



Video 1 - Mono-Replay : a software tool  
for digitized sound animation

## Author Keywords

Human-machine interaction, Sonic interaction design, Collective music making

## CCS Concepts

• **Applied computing** → Arts and humanities; **Media arts**;

# 1. Introduction : Principles and related works

## 1.1 Animation of sound

The aim of our work is to design a new kind of instrument allowing for gesture-controlled expressive playback of recorded music samples. The noun "animation" comes from the verb "to animate" for pointing out the nature of this interaction: the performance as a mean of rendering "alive" the fixed material in a recording. "Animating" also means "giving (something) a movement". As the setting in motion involves the displacement of a body and changes of position in space, the concept of animating music stored on medium, indicates some active physical involvement from a person launching a musical process. According to this concept, degree zero of animation of sound would be the minimal gestures on a button to play and stop a recording [1].

"Animation of sound" articulates automation (as defined in digital audio workstations) and gestural interaction, in order to play recorded music samples with expression, i.e. using gesture controls to convey the performer's musical intentions. This term has been initially introduced by Norbert Schnell [2] in his work on interactive audio applications.

## 1.2 Interactive music applications

### Music on storage media as a creative material

Sound recording has a long history that can be traced back to Édouard-Léon Scott de Martinville's *phonotaugraphe* (1853) or Thomas Edison's phonograph (1877), allowing the storage of sound waves on paper or a metal cylinder. Music can be recorded as well on storage media, and the idea of using these recordings and playback tools as creative sound instruments came a few decades later. An early example is the poet Guillaume Apollinaire who, using his own voice in poetic declamation, saw the gramophone as a tool for creating a "vertical, non-declamatory poetry with simultaneous action"[3]. Composer Carol-Bérard proposed (20 years before P. Schaeffer's *musique concrète*) that carefully grouping and associating recorded gramophone sounds could produce well-formed symphonies [4]. However, the first systematic research into composition with fragments of recordings is associated to the *musique concrète* initiated in 1951 by P. Schaeffer. The "Groupe de Recherche Musicale" carried out all kinds of experiments aiming using storage media for musical and sound creation, experiments and interactions[5].

The question of live rendering, or performance, for electroacoustic music has been addressed in various ways. The "Active Listening Interface" [6] by Masataka Goto aims at taking the user of audio playback systems out of her/his sole role of passive listener. "Active listening" means physical interaction with the system to enhance the listening experience: the user can e.g. switch from one chorus to the other, accentuate or diminish the drum part or transpose and slow down specific parts of a song [7].

The idea that the user can be more than a mere passive listener of a particular song and can interfere in its execution totally relates to the evolution of design concepts, from design to with the idea of co-design [8]

The use of Loudspeaker orchestras, or Acousmonium, is another way to perform recorded electroacoustic music [9].

However the most widely used way to animate recording is DJing or "turntablism". Several sophisticated techniques for multiple vinyl records expressive playback have been elaborated, including speed variation, scratching, mixing, digital audio effects etc. Digital virtual DJing has also been developed, allowing for various effects on a graphic tablet[10][11] .

The *Méta-DJ* software developed by *PUCE MUSE* (freely available at [www.meta-librairie.com](http://www.meta-librairie.com)), following the *Méta-malette* project [12] , *Méta-DJ* offers various real-time recordings transformations. It is especially designed for collective practice, along the line of the *OrJo* project [12] (Joysticks orchestra) . *Méta-DJ* has been used in the *Great DJs Orchestra (Le Grand Orchestre de DJs)* made of secondary school pupils playing together a musical piece on their personal computer.

### Sound animation using anchor points: Méta-piano, Méta-instrument and performative voice synthesis

Several methods for sound animation using performative control of anchor points in the timeline of recorded music have been proposed. A sophisticated system for expressive performance of piano music has been developed in Jean Haury's *Méta-Piano* [13]. This system allows a musician to perform a piece on a digital piano using a MIDI keyboard reduced to a few keys (typically 1 to 5 or 8).

Unlike in the actual piano, action on the keys does not control all details of the score, but the sequencing (including rhythm, articulation, phrasing, tempo and accentuation) of selected time events previously marked in the the score. The musician proposes

her/his own interpretation by controlling meta-parameters of the musical discourse, and not by directly producing all the notes in the score, as it is usually the case. The *Meta-Piano* makes it possible to concentrate entirely on the expressive parameters of a keyboard interpretation, using the recording as material and as a shortcut to resume or reduce the instrumental technique itself. Music that would not be playable as such on the piano (e.g. string quartet transcriptions) can be truly and expressively performed using the *Meta-Piano*, even by impaired people.

Complex scores can be performed using similar principles using Serge de Laubier's meta-instrument. Unlike the reduced keyboard of meta-piano, the meta-instrument encompasses a large number of sensors in a sophisticated digital interface [14]. The Meta-instrument is a preferred interface for Mono-replay, as shown in video 1 (0:06 to 0:37).

Voice samples animation, coined performative voice synthesis, has been developed since the *Speech conductor* by Christophe d'Alessandro and colleagues. The *Voks* [15] [16] system is based on temporal blocks concatenation using time anchor points placed on voice samples. Thanks to a vocoder, the performer controls parameters such as intonation, vocal quality and vocal strength. Accurate melodic control is achieved using a graphic tablet. Like in the preceding systems, a preliminary analysis and labelling of the the sound samples is necessary. In this case time anchors are placed for syllable re-sequencing. Two rhythmic control points per syllable are necessary for voice control as triggering and releasing are as important for musical performance.

## Music information labelling

The sound animation methods described above rely on some sort of musical information labelling in the audio stream. Musical information labelling is a two sided process: on the one hand it can be seen as a musical analysis problem, on the other hand as a signal processing problem. From the point of view of signal processing it is possible to automatically extract accurate information on the audio content [17] without any human listening and analysis. For instance, in Philippe Gonin's study [18] an algorithm determines the novelty curve on several improvised versions of the Pink Floyd's *Interstellar Overdrive*. This curve provides information about sudden spectral variations directly from audio code, in this case change of tonality. To find out which algorithm is the most appropriate for retrieving different musical discourse features, theoretical musical knowledge is mandatory. For example, to manually segment a piece of Free Jazz as per its structure, prior to subjecting it to an automatic segmentation, a knowledge of the studied record is indispensable [19]. The bridge between signal

processing and musical analysis is MIR (Musical Information Retrieval). In this field, scientists and musicologists work together to design automatic systems capable of extracting information from musical discourse directly from the digitized audio. MIR is used in Mono-Replay as well for determination of anchor points in the audio stream.

### 1.3 Mono-Replay

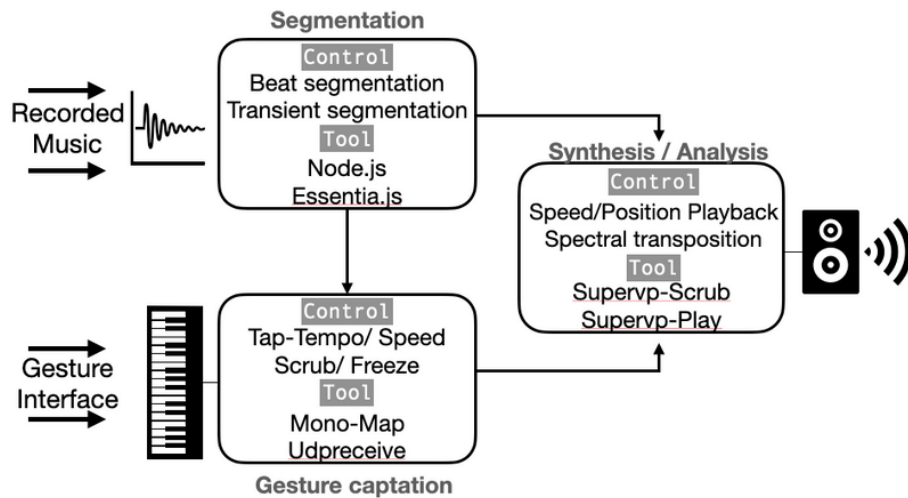
*Mono-replay* is based on principle similar to the *Méta-piano* and *Voks* for the piano and voice, but for any kind of recorded music. Sound animation relies on a preliminary analysis of the sound material in terms of a three-level hierarchical timeline. Anchor points are defined along this time line as explained in Section 2.

Then comes definition of the instrument itself, an instrument being defined as a gestural interface and a set of possible interactions.

A *Mono-replay* performance is based on a specific setting of recording, anchor points, gestural interface, gesture to effect mapping, playing mode and digital audio effects.

*Mono-Replay* embeds two main functional aspects: the segmentation by analysis and labelling of the content of the recording, and then the definition of the instrument as an assembly of interfaces to manipulate the labelled content. Image 1 gives a schematic view of the principles of Mono-Replay and its two main components.

Note that like in the *Méta-Dj* and *OrJo project*, *Mono-Replay* is especially fit to collective practice, as it runs on various platforms, with various interfaces (including cheap video game interfaces). Examples will be given below.

**Image 1**

Schematic view and principle of Mono-Replay. Analyse and segmentation of a recorded file based on puls and transient detection; Gesture captation with complex mapping on Synthesis parameter; Producing audio by controlling Synthesis parameter.

## 2. Musical discourse units and audio labelling

The first stage to design an expressive interaction scenario for digitized sound animation is segmentation and labelling of the audio stream into time blocks that will be used in the musical discourse.

In the case of Mono-Replay, this segmentation is usually based on three layers: the notes or perceived beats layer, the metrics or measures layer and the musical phrases layer.

This first stage aims at positioning hierarchical anchor points in the time flow in order to identify the phases or blocks of time that we wish to manipulate. There are two main ways for producing this kind of segmentation :

- A *Multi-feature Beat Tracking* [20] algorithm allows to create segmentation linked to metric division like measure, whole note, half note, quarter note etc. This method of segmentation is mostly relevant when working on a record showing a clearly perceptible tempo.
- The *SuperFlux* [21] onset detection algorithm is used for music pieces without clearly perceptible tempo. In this case, individual onsets are searched for instead of



regular/periodic beats or pulses.

In the following paragraphs we will describe each of these methods and illustrate them with musical examples.

## 2.1 Segmentation based on pulse detection : The case of “Superstition”

### Beat tracking with *essentia.js*

The beat can be defined as the "perceptually induced periodic pulse that is best described by the action of *foot-tapping* to the music" [22]. When listening to music that has a pulse, ancillary movements synchronised with the tempo of the piece often happen. A segmentation method based on pulse detection (when it exists in the music) is developed, because it allows for intuitive audio stream control based on pulsed beats.

The *Multi-feature Beat Tracking* [23] algorithm seemed fit for this application. This algorithm, available in the free and open source Javascript library *essentia.js* [24], is integrated to *Mono-Replay* using the Max external *node.js*. It determines nine transient detection functions using different spectral methods, in order to keep only the most suitable for the type of sound at hand. As a matter of fact, transients of a plucked string sound do not have the same spectral properties as transients of a rubbed string sound. A method based on similarity matrices compares each of the nine sequences obtained and selects the one showing the greatest similarity to the others as the most representative. This method gave the best results when compared with sixteen other pulse determination systems [25].

### Three level segmentation of *Superstition*

*Superstition* is a famous funk rock piece by Stevie Wonder and Jeff Beck (1972). The recording in the album *Talking Book* is available in stem format consisting of a multi-track file of the main instrumental parts (drums; vocal; bass; brass section; clavinet). This format is often used by DJ's who need to play each instrumental part of the same recording separately. In our case the stems are well suited for an ensemble playing of *Mono-Replay*, each musician expressively controlling a separate instrumental part (image 4). Three levels of segmentation are useful for synchronisation, because they allow for accurate segment selection and easy time jumps (see video 1 - 01:32 to 01:56).

- **First level segmentation : the Beat**

With the *Multi-feature Beat Tracking* algorithm [26] we create a segmentation

according to the pulse of the track. Each segment is bounded by two consecutive pulses. For *Superstition*, the *Multi-feature Beat Tracking* algorithm finds a tempo value of 102.3 BPM. Each segment made by two consecutive pulses has a duration of approximately 0.56 second.

- **Second level segmentation : the Measure**

Once we get a beat segmentation we choose a rhythm signature corresponding to a certain number of pulses per segment. For *Superstition* we choose a number of 4, which is the rhythmic signature of the piece. Each time we count 4 pulses we create a new second level segmentation.

- **Third level segmentation : the Structure**

The last level of segmentation is called “structure” and informs the user about the start of a new composition part. For *Superstition* we create this segmentation manually by indicating the time position of the Intro, Verse 1, Chorus, Verse 2, Chorus, Bridge, Verse 3 and Outro of the track.

A similar type of segmentation is used for any kind of metrical music. An example is given in video 1 (0:06 to 0:37) showing a short piece of Schubert’s string trio in C minor.

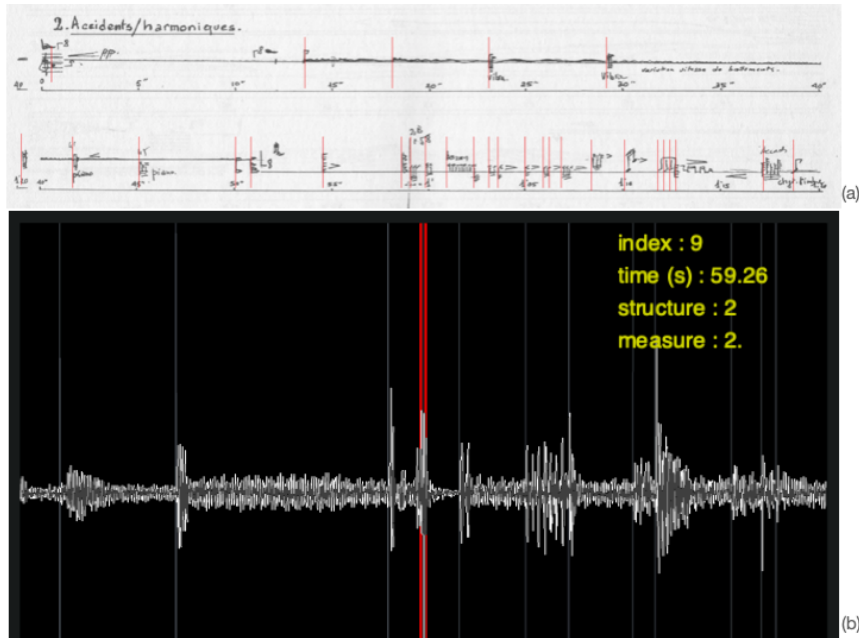
## 2.2 Segmentation based on transient detection : The case of “Accidents/Harmoniques”

Transient detection with *Essentia.js* :

Research in Musical Information Retrieval (MIR) has led today to extremely efficient audio content analysis tools. By identifying values above a threshold level on the novelty curve, the *SuperFlux* [27] algorithm, available in the free and open source Javascript library *essentia.js* [24], can detect transients of percussive sounds, as well as softer transients marked by a fundamental frequency change.

The spectral flux is the sum of the band-by-band difference in the amplitude of the spectrum obtained by a short-term Fourier transform. The maxima observed on the spectral flux representative curve - also called the novelty curve - indicates the sudden spectral variations due to the occurrence of a new event, which we associate with a new note in the analysed recording. By retrieving the temporal positions of these maxima, we obtain a collection of time indexes for important events (i.e. transitions or transients) in the recording (Image 2).

## Segmentation of *Accidents/Harmoniques*



**Image 2**

A representation of the segmentation with the *Superflux* Algorithm on a musical transcription of *Accidents/Harmoniques* (a) and on the waveform display on *Mono-Replay* GUI (b).

*Accidents/Harmoniques* is the second movement of *De Natura Sonorum*, an electro-acoustic piece composed by Parmegiani (see video 1 - 01:56 to 02:17 ). This movement does not have an identifiable metrical structure, so we do not attempt to determine a pulse or tempo here. Using signal-processing tools that we detail below, we develop a collection of event-based indexes: the temporal position of each sound event beginning (first level segmentation) or of groups of sound events - second level segmentation - with sufficiently brief and perceptually pronounced attack transients is stored in a text file.

### 3. Gestural control in performance

Like other interactive audio applications [2], *Mono-Replay* is based on three pillars: audio analysis and segmentation, gesture capture and analysis, and sound synthesis (see Image 1). In this section interactive and expressive animation is discussed. The different playing modes available in *Mono-Replay* for controlling audio synthesis are

presented. As an example, a multi-player usage of *Mono-Replay* for animation of a multi-track version of *Superstition* is described (image 4).

### 3.1 Control interfaces

A *Mono-Replay* instrument is made of a computer running the software and an interface for controlling the performance. *Mono-Replay* belongs to a software suite (the Mono- ) developed at the *PUCE MUSE* company for artistic and pedagogic electronic music applications. This environment offers a common framework for multiple interface selection, sound visualisation, file formats, graphical interface design.

Part of the control for *Mono-Replay* is done using the computer interface (keyboard and mouse). Each musical project is made of the recorded sound and associated time segmentation labels. The projects are loaded and selected using the computer GUI. Then other options for a specific performance are selected: the rhythmic playing mode, spectral modification, audio output.

For many years now, *PUCE MUSE* explores the correlation between instrumental gesture and audio perception in pedagogical contexts [28]. Several types of interfaces can be used in conjunction with *Mono-Replay*. The most sophisticated is the *Meta-Instrument IV*, allowing for accurate control of more than 34 sensors driven by both elbows, wrists and fingers (see video 1 : 0:06 to 0:37) . For pedagogic purposes or laptop orchestras, more usual video game interfaces like joysticks or gamepads can be used as well (see video 1 : 0:36 to 01:56 ).

The control interface must at minimum offer buttons and sliders for time segment selection. Continuous controllers are useful for spectral modifications or continuous time control.

The mapping proposed for controlling *Mono-Replay* is schematised in the figure below :

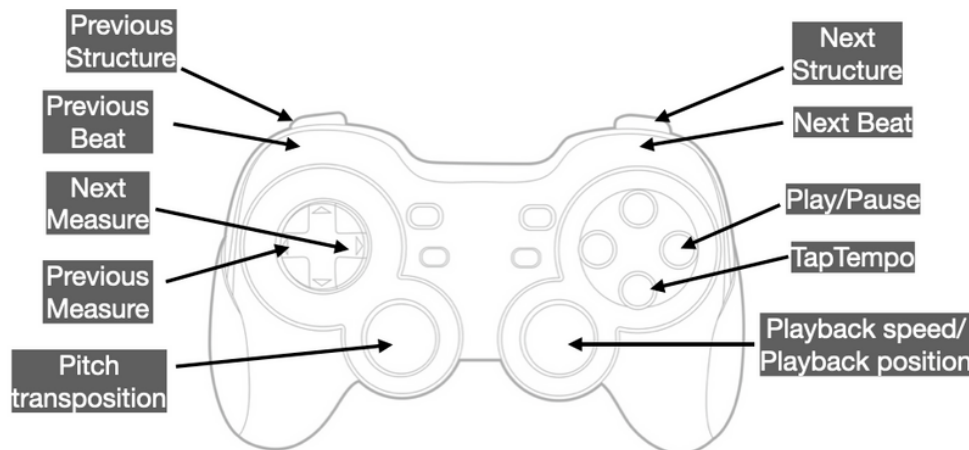


Image 3

A Mapping to control Mono-Replay on Logitech gamepad

This mapping allows for an easy interaction to move from a beat/measure/structure to another. The main control of Mono-Replay are accessible on the palm of the hand. Users are often familiar with the use of Gamepads. This ability helps in a quick start of expressive playing and allows the user to focus on the sound produced thanks to familiar control gestures

### 3.2 Controlling speed and position playback

According to the methodology described in the previous Section a three-tier segmentation has been created for the stems of *Superstition*. It can be used to play back music from a segment to another during performance thanks to 6 playing modes :

- **TapTempo direct** : The playback speed of the audio file is determined by the tempo beaten by the user. If the user executes two consecutive beats with a duration corresponding to the detected tempo of the track, the playback speed is set to 1. For *Superstition* the tempo determined by the *Multi-feature Beat-tracking* algorithm is 102.3 BPM ; which corresponds to approximately 0.56 second between two consecutive beat.
- **TapTempo direct next** : this playing mode works exactly as the TapTempo direct mode with the difference that the new playback speed value is set at the beginning of the next segment.

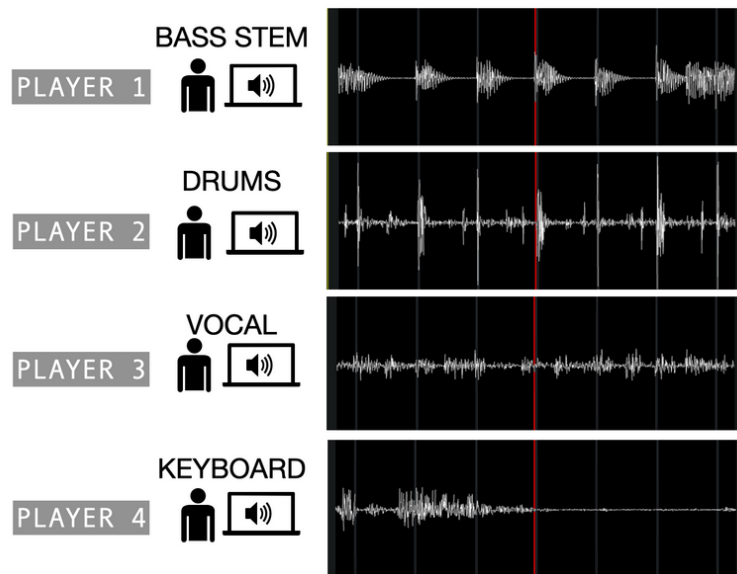
- **TapTempo next** : The playback position and speed are set up at each new pulse beaten by the user. The speed is determined by the duration between two consecutive beats executed by the user and the position is set to the next segment each time the user executes a beat.
- **Speed direct** : The playback speed is set with a slider and can take a value between 2 (twice as fast) and -2 (twice as fast in reverse mode). (see video 1 - 0:36 to 0:56 )
- **Speed freeze** : This playing mode works exactly as the Speed direct mode with the difference that at the end of each segment the speed is set to 0; which generates an audio freeze. (see video 1 - 0:57 to 01:56)
- **Scrub** : In this playing mode we control the position playback with a slider. The slider is bounded with the beginning and the end of the observed segment. (see video 1 - 01:56 to 02:17 )

### 3.3 Polyphonic animation of a stem file

*MonoReplay* is designed for collective music making. To explore the musical usability of *MonoReplay* in this context several workshops with musicians experienced in musical interfaces were organized, aiming at polyphonic rendering of stem files. A Stem file is a multitrack audio format released by Native Instrument in 2004 that enables DJ or live performers to play different separate parts of a song. A stem file contains vocals, drums, bass and keyboard tracks of the same recording. Playing all tracks of the stem file together produces a mixed version of the piece. A stem file for of *Superstition* was used here containing a four separate tracks with drums, bass, vocals and clavinet + brass section. Our setup is made of four personal computers running *Mono-Replay*. Each computer uses the same segmentation file but a different instrumental track.

With the first computer we control the playback's speed and position of the bass track of *Superstition*; with the second one we control the playback's speed and position of the drums track; with the third one we control the playback's speed and position of the vocal track; and with the last one we control the playback's speed and position of the keyboard & brass section track. Every player can move on the track according to the three levels segmentation described below. Each player can choose a playing mode from those described above to control the the playback's speed and position of the record they animate. Experience shows that the best mode to play together with a synchronised phase and speed is the mode TapTempo Direct. This mode allows to control the speed and the phase of the playback each time a user clicks on the mouse.

If one player counts aloud a tempo it's easy for everyone to synchronise and click on their mouse simultaneously. If each player plays consecutive clicks with the same duration and phase, they are controlling the playback's speed and position of the record synchronously.



**Image 4**

Multiplayer use of Mono-Replay : Each player control the speed and position playback of an instrumental part of a stem version of Supersition . The segmentation is the same for each instrumental part and it is based on *Multifeature beat-tracking* algorithm

### 3.4 Spectral transformation and audio effects

Expressive performance in sound animation is mainly based on time control of labelled events. Other relevant sound parameters can be controlled as well. First of all, the sound level or sound intensity is easily controlled. *Mono-Replay* is an analysis-synthesis process. Sound samples are processed using a vocoder. *SuperVP* and *groove~* are two phase vocoders implemented in the form of configurable Max externals, allowing accurate control over the position or playback speed of an audio segment, while maintaining very high definition in the sound textures played back. The *SuperVP* is used to control the position of the audio file in the "Scrub" mode, while the *groove~* object is used in the "Speed" mode to control the playback speed of the segment. The vocoder is useful for many kinds of spectral sound transformation like filtering, phasing, tuning, pitch modification and so on. As the sound samples are

enriched with marks corresponding both to the score and to the actual sound content of the recording, sound transformations and audio effects can be applied to specific segments according to real time controls provided by the interface.

## 4. Conclusion

Various forms of sound animation have been developed since the advent of sound recording. Even the simple gesture of playing back a recording can be considered as a basic form of sound animation. Musique Concrete developed musical composition based on recording, DJing developed musical performance based on recording. The present work merges these musical approaches with the instrumental (in the sense of NIME) approach.

The *Mono-Replay* software allows for building instruments for sound animation. It is a two stage process. In a first stage, the recorded music sample is analyzed and labelled, using MIR techniques for segmentation (beat tracking or transient detection) into musical units.

The main challenge in the segmentation process is to retrieve relevant information from the digitized sound, according to musical structures. *Mono-Replay* embeds a three scales segmentation method for this purpose. The first scale gives time position of rhythmic events or a pulses. The second scale gives time position of a group of rhythmic events or pulses and the third scale gives information on the main composition parts of the piece (e.g. intro, chorus and verse in the case of *Superstition*).

At the performance stage, *Mono-Replay* gives control on these time labels for playing and interpreting the piece. Each control point at the three segmentation scales can be easily reached. This allows for fine timing control, comparable to instrumental control, and thus to collective practice.

To go beyond the control offered by the mouse and the keyboard of the computer, simple or sophisticated interfaces can be used, through a mapping process. A Gamepad for instance allows for keeping all the controls of *Mono-Replay* in the palm of the hand. Familiarity with Gamepads allows to fully engage the player in the performance. The meta-instrument is another kind of interface we tried. Pads and graphic tablets or any other type of NIME could be used as well.





**Image 5**  
The Graphic User Interface of Mono-Replay

As for the computer GUI, for the moment, we only display the waveform and indexes placed on the audio file in our graphical interface (Image 5). This representation is rather limited in terms of musical information and does not allow to anticipate the precise nature of the next sound that will be played. It would be necessary to integrate a more musical representation, such as a score that would scroll synchronously with the sound. One possible way could be to integrate an iconic view of the sound to get a representation based on sound parameters.

Mono-replay is a fully functional application that will be distributed by *PUCE MUSE* in the Mono suite. It will then be used in professional artistic practice and in pedagogic environments, to introduce various audiences (such as secondary classes and conservatory students) to electronic instruments and electronic music.

## Acknowledgments

This work has received financial support from the Ile de France region and the European Union through the SMAC project (ERDF IF0011085).

## Citations

1. T. Lucas, S. de L., C. D'Alessandro. (2020). Animer la musique sur support. In *Acte des Journées d'Informatique Musicale 2020*. Strasbourg. [↗](#)

2. Schnell, N. (2013, October). *Playing (with) Sound Of the Animation of Digitized Sounds and their Reenactment by Playful Scenarios in the Design of Interactive Audio Applications*. Austria. [↵](#)
3. Appolinaire, G. (1991). Oeuvres en prose complètes. In Galimard (Ed.) (Vol. 2, p. 982). P. Caizergues, M. Décaudin. [↵](#)
4. Battier, M. (2007). What the GRM brought to music: from musique concrète to acousmatic music. *Organised Sound*, 12, 189-202. [↵](#)
5. Schaeffer, P. (1966). *Traité des Objets Musicaux*. Paris, France: Seuil. [↵](#)
6. Goto, M. (2007). Active Music Listening Interfaces Based on signal Processing. In *Proceedings of the 2007 IEEE International Conference on Acoustics, Speech, and Signal Processing* (pp. 1441-1444). [↵](#)
7. Goto, M. (2003). A Chorus-Section Detecting Method for Musical Audio Signals. In *Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing* (Vol. 14, pp. 1783-1794). [↵](#)
8. Sanders, L., & Stappers, P. J. (2014). From Designing to Co-Designing to Collective Dreaming: Three Slices in Time. *Interactions*, 21(6), 24-33.  
<https://doi.org/10.1145/2670616> [↵](#)
9. Bayle, F. (1993). Musique concrète, musique acousmatique, propositions... ..positions, (Musique concrète, acuosmaic music, propositions ... ..positions). In Ina-GRM-Buchet/Chastel (Ed.). Paris. [↵](#)
10. K. Falkenberg Hansen, R. B. (2006). Mapping strategies in DJ scratching. In *Proceedings of the New Interfaces for Musical Expression* (pp. 188-191). IRCAM. [↵](#)
11. K. Falkenberg Hansen, R. B. (2010). The skip-proof virtual turntable for high-level control of scratching. *Computer Music Journal*, 34, 39-50. [↵](#)
12. S. de Laubier, H. G., L. Feugère, G. Bertrand, V. Goudard, C. d'Alessandro. (2012). OrJo et la meta-malette. In *Actes des Journée d'Informatique Musicale 2012* (pp. 227-232). Mons, Belgique. [↵](#)
13. Haury, J. (2009). La pianotechnie ou notage des partitions musicales pour une interprétation immédiate sur le métapiano. In *14èmes Journées d'Informatique Musicale* (pp. 9-17). Grenoble. [↵](#)

14. Couprie, P. (2018). "Le Méta-Instrument : genèse et évolution d'un nouvel instrument." (The Méta-instrument: genesis and evolution of a new novel instrument.). *Musique, Images, Instruments, vol 17*, pp 230-245. [↵](#)
15. D'Alessandro, C., Xiao, X., Locqueville, G., & Doval, B. (2019). Borrowed Voices. In *Music Proceeding of the International Conference on New Interfaces for Musical Expression* (p. 2.2-2.4). Porto Alegre, Brésil. [↵](#)
16. D'Alessandro, C., Xiao, X., Locqueville, G., & Doval, B. (2019). Borrowed Voices. In *Music Proceeding of the International Conference on New Interfaces for Musical Expression* (p. 2.2-2.4). Porto Alegre, Brésil. [↵](#)
17. Peeters, G. (2003, 0). *A large set of audio features for sound description (similarity and classification) in the CUIDADO project*. [↵](#)
18. Gonin, P. (2019). Interstellar Overdrive de Pink Floyd et le processus de création à travers le prisme de l'improvisation collective : du chaos à l'ordre ? *Musimédiane*, 11. [↵](#)
19. M. Tahon, P. M., Z. Belgrith, J. M. Chouvel. (n.d.). Segmentation automatique de catégories musicales : Étude exploratoire sur le Free Jazz. In *JIM 2019*. [↵](#)
20. Zapata, J. R. (2014). Multi-Feature Beat Tracking. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 22(4), 812-825. [↵](#)
21. S. Böck, W. G. (2013). Maximum Filter Vibrato Suppression for Onset Detection. In *Proceedings of the 16th International Conference on Digital Audio Effects (DAFx-13)* (pp. 55-61). Maynooth, Ireland. [↵](#)
22. Jehan, T. (2005). *Creating Music by Listening*. [↵](#)
23. Zapata, J. R. (2014). Multi-Feature Beat Tracking. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 22(4). [↵](#)
24. D. Bogdanov, X. S., N. Wack, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salomon, J. R. Zapata. (2009). ESSENTIA: an Open-Source Library for Sound and Music Analysis. [↵](#)
25. Holzapfel, A., Davies, M. E. P., Zapata, J. R., Oliveira, Jo~ao Lobato, & Gouyon, F. (2012). Selective Sampling for Beat Tracking Evaluation. *Trans. Audio, Speech and Lang. Proc.*, 20(9), 2539-2548. <https://doi.org/10.1109/TASL.2012.2205244> [↵](#)

26. Zapata, J. R. (2014). Multi-Feature Beat Tracking. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 22(4), 816-825. [↵](#)
27. S. Böck, W. G. (2013). Maximum Filter Vibrato Suppression for Onset Detection. In *Proceedings of the 16th International Conference on Digital Audio Effects (DAFx-13)* (pp. 55-61). Maynooth, Ireland. [↵](#)
28. Patiño-Lakatos, G. (2015). Expérience à l'institut Gustave Baguer D'Asnieres. In *Internal report, LAM-D'Alembert, Sorbonne Université*. [↵](#)

Annexe C

Manuel du logiciel Mono-Replay


# Mono-Replay

<b>Installation du logiciel</b>	<b>2</b>
Installation pour Windows	2
Installation pour Mac	4
<b>Interface du Logiciel</b>	<b>5</b>
Panneau de lecture :	5
Catégorie de paramètres :	7
<b>Catégories</b>	<b>8</b>
Transformation audio	8
Tap tempo	8
Scrub	9
Speed	9
Paramètre de synthèse	10
Enveloppe	12
Rendu	12
Analyse	13
beat	14
onset	15
<b>Mémoires et Interpolations</b>	<b>15</b>
Mémoires	16
Interpolations	19
Plus d'options	20
<b>Paramètres avancés</b>	<b>22</b>
Instances MonoReplay	22

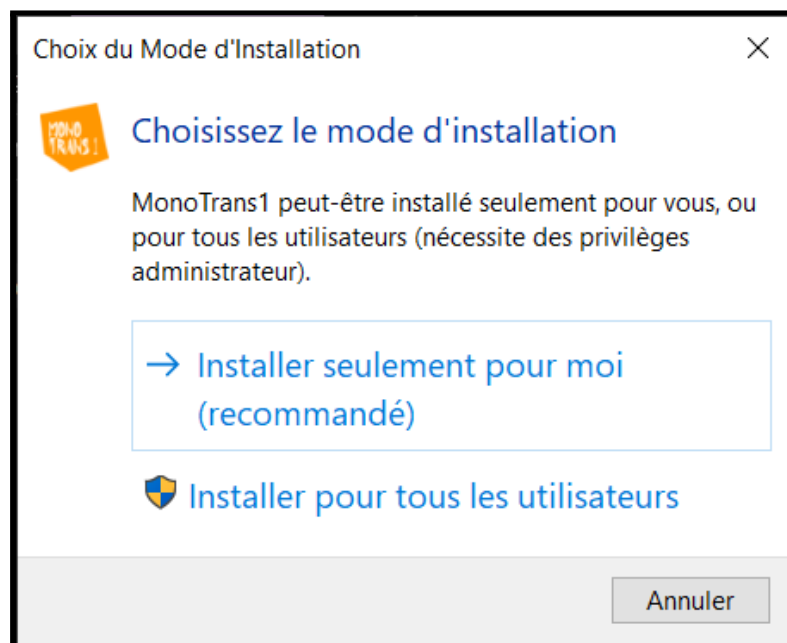
# 1. Installation du logiciel

## 1. Installation pour Windows

Une fois que vous avez récupéré le programme d'installation de MonoReplay sur le site de la [Meta-Librairie](#) il suffit de le lancer et de suivre les instructions. Vous pouvez éventuellement changer l'emplacement d'installation si vous le souhaitez.

 Setup_MonoTrans1_v2.4.1	01/09/2020 14:14	Application	98 389 Ko
---	------------------	-------------	-----------

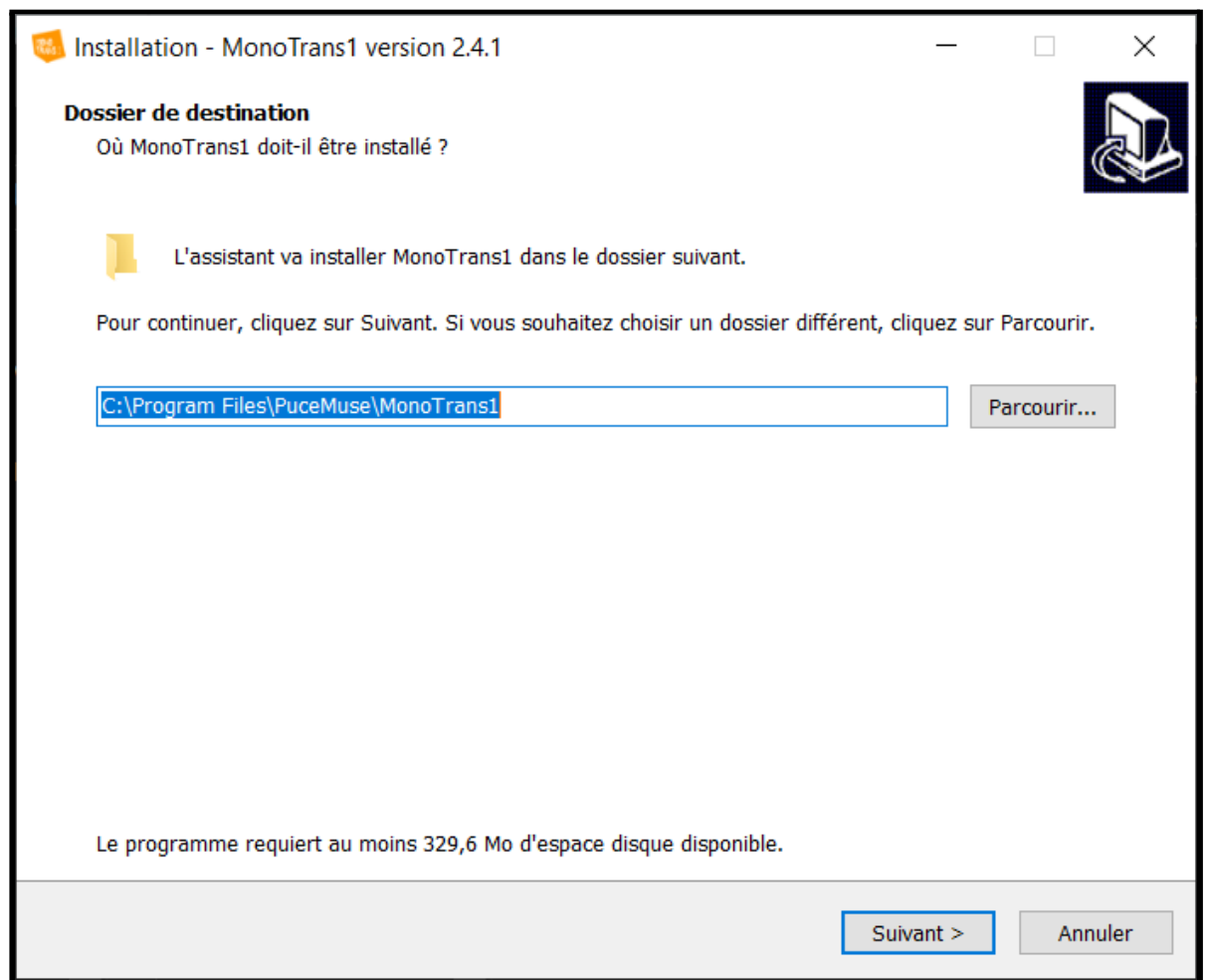
Voici à quoi ressemble le programme d'installation :



Si vous souhaitez installer le logiciel pour tous les utilisateurs, vous aurez besoin des droits administrateurs.

**ATTENTION** : l'installation pour tous les utilisateurs ne fonctionne pas correctement si vous l'exécutez depuis une session non administrateur. Si vous avez le moindre de doute, il est préférable de choisir "*Installer seulement pour moi*".

Une fois votre choix effectué, la fenêtre suivante s'ouvrira :



Il ne vous reste plus qu'à suivre les instructions qui s'afficheront et cliquer sur "Suivant >".



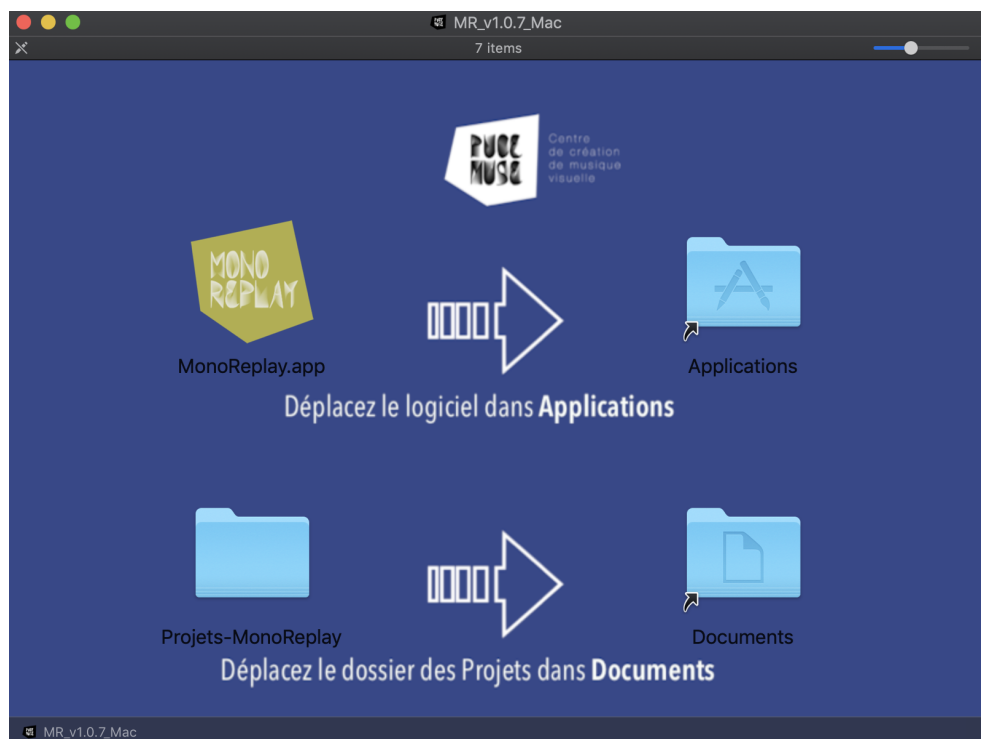
## 2. Installation pour Mac

La première étape consiste à télécharger le logiciel sur le site de la [Meta-Librairie](#). Vous obtiendrez un fichier *dmg*.

Une fois ouvert, vous devriez voir apparaître un nouveau périphérique dans *Appareils* :



En cliquant dessus, vous devriez voir la fenêtre suivante s'ouvrir :



Il ne vous reste plus qu'à glisser-déposer le dossier *MonoReplay* dans *Applications* et le dossier *Projets-MonoReplay* dans *Documents*. Vous pouvez maintenant utiliser le logiciel que vous trouverez dans votre dossier *Applications*.

## 2. Interface du Logiciel



### 1. Panneau de lecture :



← numéro d'instance de votre logiciel (par défaut 1)

← bouton de lecture et pause des fichiers son

← Lance un enregistrement de la session

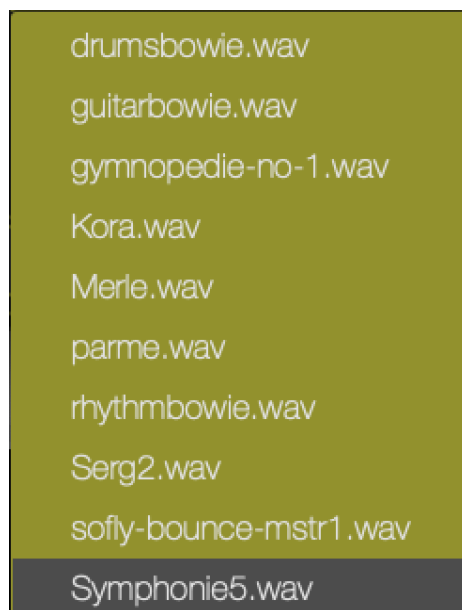
← activer ou désactiver le son

← réglage du volume de sortie

*MonoReplay* est un logiciel vous permettant d'analyser un enregistrement audio pour y déposer des ancrs temporelles en vue de contrôler certains paramètres de lecture.

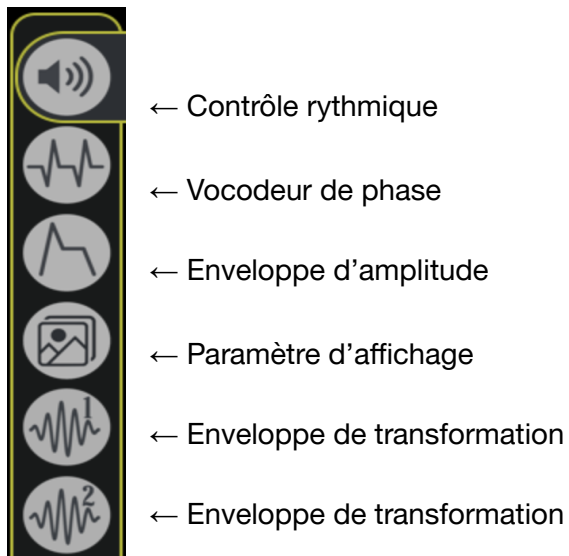


Ici vous pouvez visualiser le nom du fichier actuellement en lecture, sa forme d'onde ainsi que la représentation des index déjà posés. si vous cliquez sur la flèche située à droite du menu déroulant vous pouvez sélectionner un fichier audio provenant de la banque de sons des différents projets (*ici il s'agit du dossier "premier projet"*)

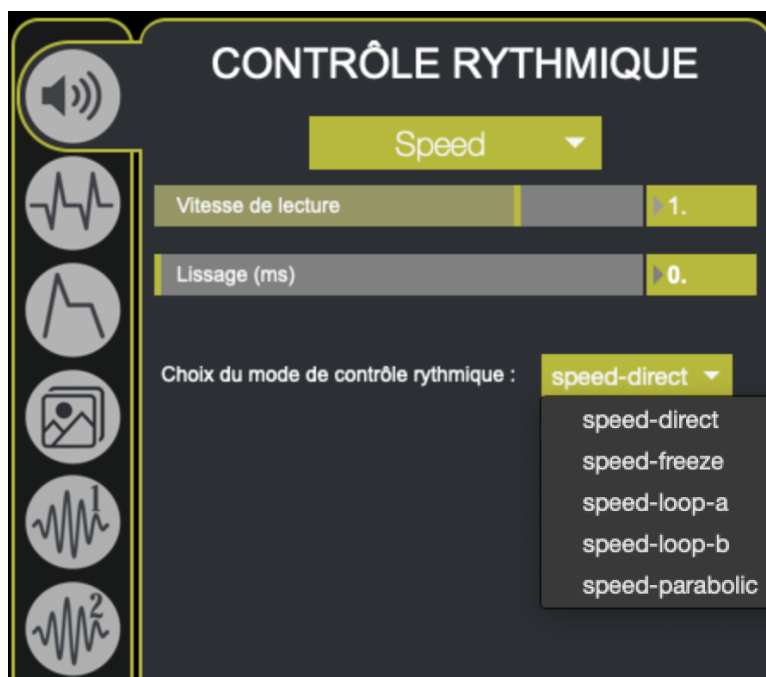


## Catégorie de paramètres :

Vous pouvez accéder aux différents onglets de l'application grâce à ces boutons.



### 1. Contrôle Rythmique



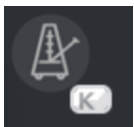
Cet onglet permet de choisir un mode de contrôle rythmique pour contrôler le déroulement temporel de l'enregistrement sélectionné.

Il y a trois grandes familles de contrôle rythmique :

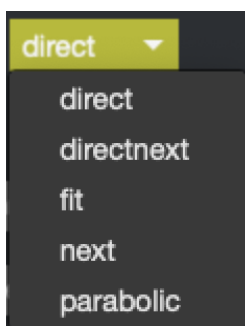
### A) Tap Tempo



→ Les 5 modes de contrôles rythmiques disponibles dans le menu *Tap Tempo* permettent de contrôler de façon discrète la vitesse et la position de lecture d'un enregistrement en battant son tempo.



→ Ce bouton permet de régler la vitesse et parfois la position relative à une pulsation battue. La valeur de BPM (Beat Per Minute) affichée correspond aux nombres de fois que l'écart temporel entre deux battues peut être contenu dans une minute.



→ Ce menu permet de choisir parmi les cinq modes de contrôle rythmique du menu *Tap Tempo* :

- **direct** : Ce mode permet de modifier la vitesse de lecture de l'enregistrement en calculant la moyenne glissante sur n valeurs d'écart temporel séparant deux pulses battues.

Pour déterminer la vitesse on fait le rapport suivant :

(Tempo moyenne glissante)/(Tempo estimé).

Le tempo estimé est déterminé par l'algorithme Multifeature Beat tracking.

- **directnext** : ce mode fonctionne exactement de la même manière que le mode précédent à la différence que la nouvelle vitesse calculée est effective au début du segment suivant.

- **fit** : Ce mode permet de modifier la vitesse de lecture de l'enregistrement en calculant la moyenne glissante sur n valeurs d'écart temporel séparant deux pulses battues.

Pour déterminer la vitesse on fait le rapport suivant :

(tempo moyenne glissante) / (durée du segment actuel).

Cette méthode est appropriée dans le cas où l'on souhaite inclure les courtes variations de tempo dans l'enregistrement joué. Dans le cas d'un suivi de pulse à l'aide de la méthode *Multifeature Beat tracking*, les pulses déterminées ne sont toutes de la même longueur. Cette méthode tient compte de la longueur du segment joué dans le calcul de la vitesse.

- **next** : Ce mode permet de modifier la vitesse et la position de lecture de l'enregistrement en calculant la moyenne glissante sur n valeurs d'écart temporel séparant deux pulses battues.

Pour déterminer la vitesse on fait le rapport suivant :

(Tempo moyenne glissante)/(Tempo estimé).

À la fin de chaque segment la vitesse de lecture passe à zéro ce qui produit un "gel" audio.

À chaque nouvelle pulse battue, la position de lecture s'initialise au début du segment suivant.

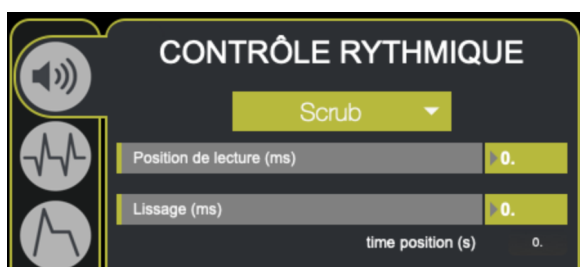
- **parabolic** : L'estimation du tempo calculé par moyenne glissante permet de calculer la vitesse de lecture de l'enregistrement par le rapport suivant :

(tempo moyenne glissante) / (tempo estimé).

Le *tempo estimé* est déterminé à l'aide de l'algorithme *Multifeature Beat tracking*.

La vitesse calculée est ensuite intégrée dans le calcul d'une parabole qui guide l'évolution de la vitesse.

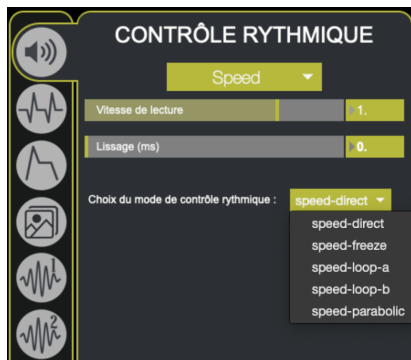
## B) Scrub



→ Ce mode de lecture permet de contrôler la position de lecture dans un segment. Un paramètre de lissage permet de régler le

temps en *ms* pour atteindre la nouvelle valeur de position. L'évolution s'effectue de façon linéaire.

### C) Speed



→ L'onglet *Speed* regroupe tous les modes de contrôles rythmiques qui permettent un contrôle direct de la vitesse de lecture de l'enregistrement. Ces sous-modes sont au nombre de 5.

Un équivalent de la vitesse de lecture est affiché en BPM. Cet équivalent correspond à la pulsation ressentie lorsque l'on modifie la vitesse de lecture.

Un paramètre de lissage permet de régler le temps en *ms* pour atteindre la nouvelle valeur de vitesse. L'évolution s'effectue de façon linéaire.

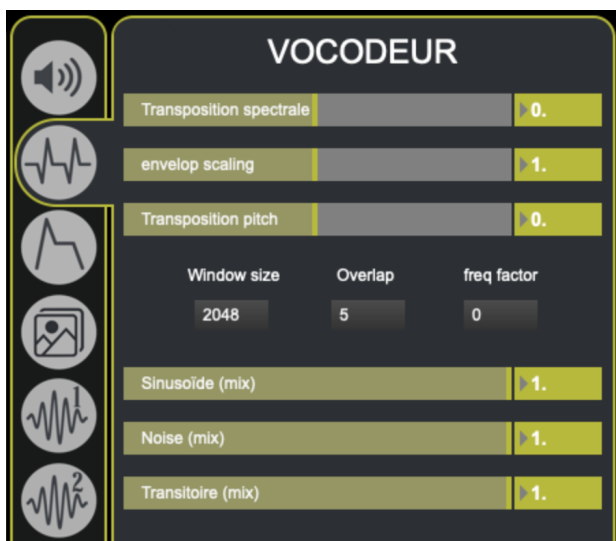
- ***speed-direct*** : Ce mode permet d'avoir un contrôle direct et continu de la vitesse de lecture d'un enregistrement. On peut régler la vitesse de lecture à l'aide du slider présent sur la figure. Le slider propose des vitesses situées entre -2 et 2. À 2 le fichier est lu avec une vitesse de deux fois la vitesse de lecture nominale et à -2 le fichier est lu avec une vitesse de deux fois la vitesse de lecture nominale en sens inverse.
- ***speed-freeze*** : Ce mode fonctionne comme le précédent avec la différence qu'en fin de segment la vitesse de lecture passe automatiquement à zéro. Cela produit un effet de gel audio et permet à l'utilisateur de marquer un temps avant de passer au segment suivant. Le passage au segment suivant se fait à l'aide des boutons prévus pour.
- ***speed-loopa*** : Ce mode lit le fichier audio en boucle pour une vitesse sélectionnée. La lecture s'effectue selon trois repères temporels placés dans le segment. Le premier est placé au début du segment, le deuxième au milieu et le troisième en fin de segment. La lecture débute au premier repère puis une fois le troisième repère atteint la tête de lecture revient au deuxième repère et la boucle audio s'effectue entre le deuxième et le troisième repère.  
La position du deuxième repère est paramétrable directement depuis l'interface. Par défaut ce repère est placé au milieu du segment (*position loop (0-1) : 0,5*)
- ***speed-loopb*** : Ce mode joue aussi en boucle le segment sélectionné selon les trois positions temporelles décrites précédemment. La lecture débute au début du segment puis une fois qu'elle arrive en fin de segment la tête de lecture revient en

arrière jusqu'au repère temporel situé au milieu du segment puis repars vers la fin du segment etc..

Selon le même fonctionnement que le mode *speed-loopa* il est possible de régler la position du deuxième repère temporel à l'aide de l'interface utilisateur.

- ***speed-parabolic*** : Ce mode fait varier la vitesse de lecture d'un segment selon une équation de parabole. La pente de la parabole est proportionnelle à la valeur de la vitesse réglée.

## 2. VOCODEUR



← Grâce à ces contrôles, on peut paramétrer le vocodeur de phase SuperVP.

Les sliders permettent de contrôler la transposition spectrale, l'échelle de l'enveloppe ainsi que la transposition du pitch. Les menus déroulants permettent de paramétrer la taille de la fenêtre d'analyse, le facteur de recouvrement et le facteur fréquentiel de la transformée de Fourier inverse. D'autres sliders permettent de contrôler le mixage des composantes harmoniques, inharmoniques et transitoires du spectre.



(100 cent =  $\frac{1}{2}$  ton ; 1200 cent = 1 octave)

← Ce slider permet de transposer la hauteur du son joué sur une plage de +/- 1200 cent.

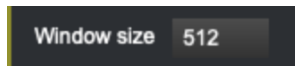


← Ce slider permet de faire varier la mise à l'échelle de l'enveloppe



← Ce slider permet de transposer la hauteur en conservant l'enveloppe spectrale sur une plage de +/- 1200 cent





← Ce menu permet de choisir la taille de fenêtre en sample de la transformée de Fourier



← Ce menu permet de définir la taille de la fenêtre glissante (Taille de fenêtre glissante = Window size / Overlap)



← Ce menu permet de changer la taille de la transformée de Fourier à court terme (Taille de la Transformée de Fourier à Court Terme = Window size \*  $2^{\text{freq\_factor}}$ )



← Ce slider permet de mixer les composantes harmoniques de l'enregistrement

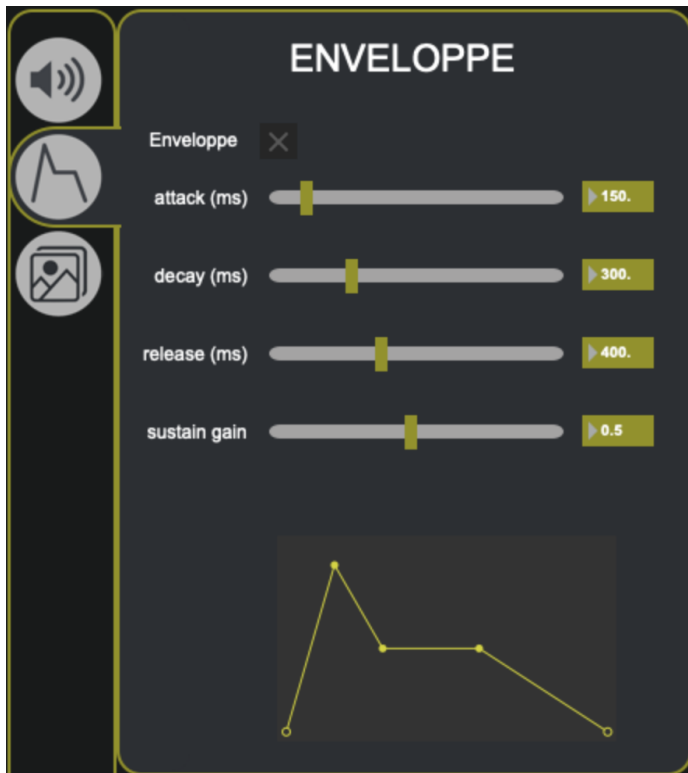


← Ce slider permet de mixer les composantes inharmoniques de l'enregistrement



← Ce slider permet de mixer les composantes transitoires de l'enregistrement

### 3. Enveloppe

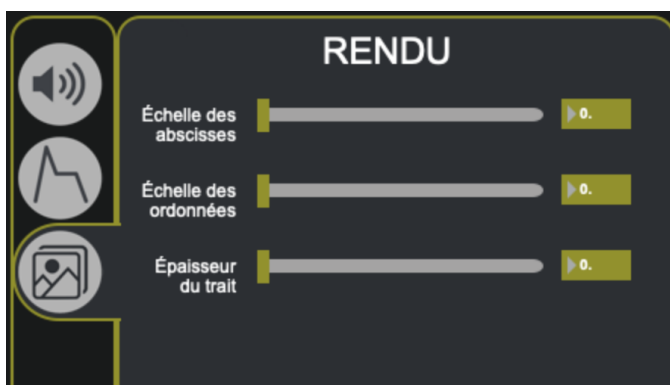


← Dans cet onglet on peut paramétrer et activer une enveloppe d'amplitude dont la durée totale sera la durée du segment.

On peut régler :

- Le temps d'attaque (attack)
- Le temps de diminution (decay)
- Le temps de relâche (release)
- Le gain du plateau de maintien

### 4. Rendu

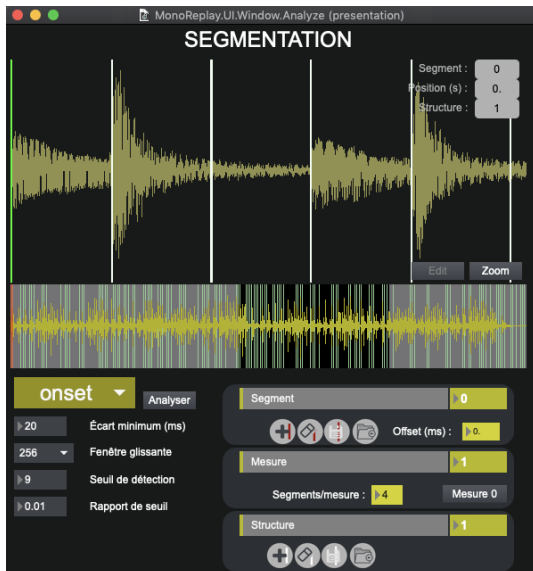


← Dans cet onglet on peut paramétrer certaines options d'affichages de la forme d'onde :

- L'échelle des abscisses pour dilater ou compresser la représentation temporelle
- L'échelle des ordonnées pour compresser ou dilater les niveaux d'amplitudes
- L'épaisseur du trait

## 5. Analyse



Pour ouvrir cet onglet veuillez cliquer sur l'icône suivant :

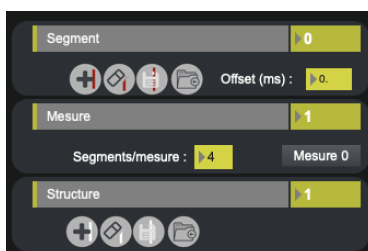


← Dans cet onglet on peut créer et éditer plusieurs segmentation :



← Ici on peut sélectionner deux modes d'interaction avec l'affichage de la forme d'onde en haut de l'onglet "Analyse" :

- mode *Edit* : Permet de sélectionner un segment à supprimer (le segment inclus dans la zone de sélection sera supprimé si on utilise le bouton : ) ou de positionner le début d'un segment à ajouter (le début de la zone de sélection correspond à la position du segment qui sera ajouté si on utilise le bouton : )
- mode *Zoom* : permet d'étirer ou de compresser la forme d'onde.








← Ici on peut naviguer dans l'enregistrement selon 3 ordres hiérarchiques :

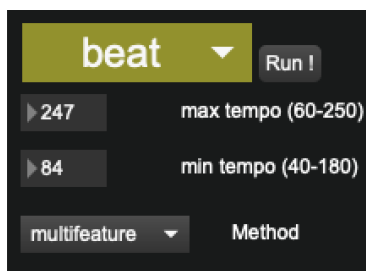
- **Structure** : permet de naviguer dans l'enregistrement par structure. On peut ajouter ou supprimer des structures à l'aide des boutons sous la barre de navigation
- **Segment** : Permet de naviguer dans l'enregistrement par segments. On peut ajouter ou supprimer des segments à l'aide des boutons sous la barre de navigation.

- **Mesure** : Permet de naviguer dans l'enregistrement selon un groupe de segments. On peut définir le nombre de segment à inclure dans le groupe avec la commande *Segments/mesures* et définir le premier groupe avec la commande *Mesure 0*.

Pour les **Segments** et les **Structure** on peut :

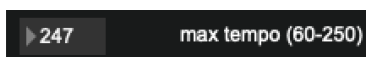
- ajouter un index à la position sélectionné  :
- supprimer l'index courant  :
- sauvegarder les modifications apportées à la collection d'index dans le coll : 
- enregistrer la collection d'index courante dans un fichier texte : 
- ouvrir un fichier texte contenant une collection d'index : 

Détection de pulse :



← Ici on peut lancer une segmentation automatique à l'aide de deux algorithmes de MIR de la librairie Essentia.js

- beat : cet algorithme permet d'estimer la pulse ressentie d'un morceau à l'aide de la méthode MutlifeatureBeatTracker ([https://essentia.upf.edu/reference/std\\_BeatTrackerMultiFeature.html](https://essentia.upf.edu/reference/std_BeatTrackerMultiFeature.html)) et la méthode de BeatTrackerDegara ([https://essentia.upf.edu/reference/std\\_BeatTrackerDegara.html](https://essentia.upf.edu/reference/std_BeatTrackerDegara.html))



→ Ici on peut paramétrer la valeur maximale de tempo à détecter (valeur en BPM).



→ Ici on peut paramétrer la valeur minimale de tempo à détecter (valeur en BPM).




→ Ici on peut changer de méthode d'analyse.

## Détection de transitoires :



- onset : cet algorithme permet de déterminer la position de transitoires d'attaque dans un enregistrement à l'aide de la méthode SuperFluxExtractor.

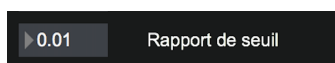
([https://essentia.upf.edu/reference/std\\_SuperFluxExtractor.html](https://essentia.upf.edu/reference/std_SuperFluxExtractor.html))



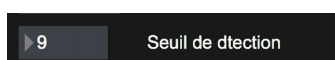
→ Ce paramètre permet de définir l'écart minimum entre deux transitoires consécutifs. Dans le cas d'un enregistrement polyphonique il arrive souvent qu'il y ait un léger décalage entre le déclenchement des différentes notes qui constituent un accord. Ces petits décalages sont détectés par notre algorithme comme le début d'une nouvelle note. Sur le principe c'est bien le cas mais dans la pratique on souhaite souvent placer une seule ancre temporelle au début de l'accord et pas plusieurs ancres pour chaque note qui le constitue. Le réglage de ce paramètre permet d'éviter la détection de transitoire trop rapproché et donc de pallier ce problème. Ce paramètre est aussi très utile pour détecter des transitoires selon une division rythmique. Si on connaît par exemple la valeur de tempo du morceau que je souhaite étiqueter je peux régler l'écart minimum entre deux transitoires sur une valeur de durée proche de celle qui sépare deux pulsations à la valeur de tempo connue et obtenir ainsi un étiquetage selon les transitoires du morceau compte tenu de cette division rythmique.



→ Ces paramètres permettent de régler la taille de la fenêtre d'analyse (frame size) ainsi que la taille de la fenêtre glissante (hop size) pour la transformée de Fourier appliquée sur l'enregistrement avec laquelle on détermine la fonction de détection de transitoire. Plus la fenêtre d'analyse et la fenêtre glissante sera petite et plus on aura de points sur la courbe du flux spectral. On pourra alors mieux détecter les très courtes variations de cette courbe. Néanmoins cela entraînera une résolution spectrale plus faible et donc une diminution de la dynamique de la courbe. Pour bien régler ce paramètre il faut trouver le juste milieu entre une bonne résolution temporelle (nombre de points sur la courbe d'évolution du flux spectral) et une résolution fréquentielle suffisante (nombre de coefficients spectraux pour notre transformée de Fourier) pour détecter suffisamment de transitoire.



→ Ce paramètre permet de régler le seuil de détection sur la courbe représentative du flux spectrale (courbe de nouveauté) à partir duquel on considère qu'un dépassement du seuil correspond à un nouveau transitoire. La valeur de ce seuil correspond au rapport (**courbe de nouveauté**)/(**moyenne de la courbe de nouveauté**) ce qui permet d'être plus sensible au transitoire de faible énergie.



→ Ce paramètre permet de régler le seuil de détection sur la courbe représentative du flux spectrale (courbe de nouveauté) à partir duquel on considère qu'un dépassement du seuil correspond à un nouveau transitoire. La valeur de ce seuil correspond à la différence **courbe de nouveauté - signal moyen** ce qui permet de tenir compte du bruit ambiant dans notre détection de transitoire

## 1. Mémoires et Interpolations

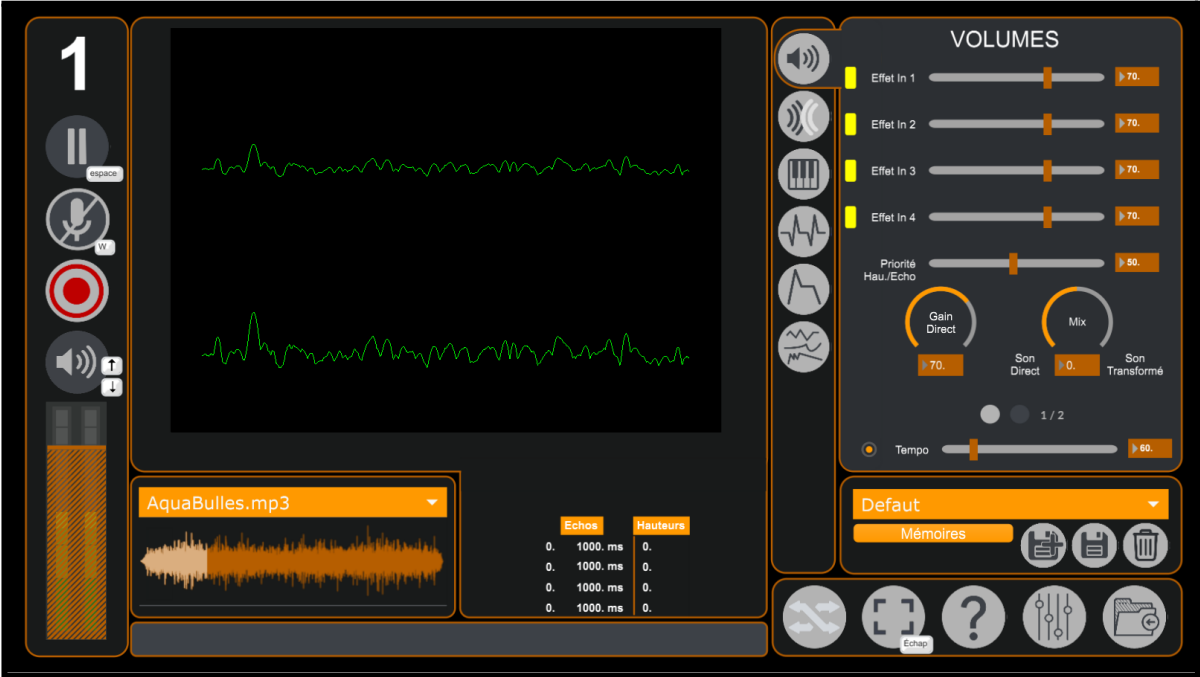
Les mémoires et les interpolations vous permettront de sauvegarder des visuels mais aussi leurs enchaînements.

*MÉMOIRE* → correspond à une sauvegarde de l'ensemble des paramètres accessibles depuis les différents onglets du logiciel.

*INTERPOLATION* → correspond à un enchaînement de plusieurs *Mémoires*.

### 1. Mémoires

Une fois que vous avez obtenu un ensemble de paramètres dont le rendu vous plaît, vous pouvez l'enregistrer.



Pour rendre accessible tous les paramètres liés aux interpolations il faut cliquer sur le bouton suivant :



Vous verrez ensuite un quadrillage apparaître au dessus du rendu visuel et un bouton “Interpolations” dans cette partie de MonoReplay :



Les boutons “Mémoires” et “Interpolations” vous permettent de gérer (modifier, créer ou supprimer) les mémoires ou les interpolations.



→ ces boutons permettent de choisir si vous souhaitez gérer les mémoires ou les interpolations



→ pour sélectionner une mémoire/interpolation à modifier, supprimer ou à charger



→ pour créer une mémoire/interpolation

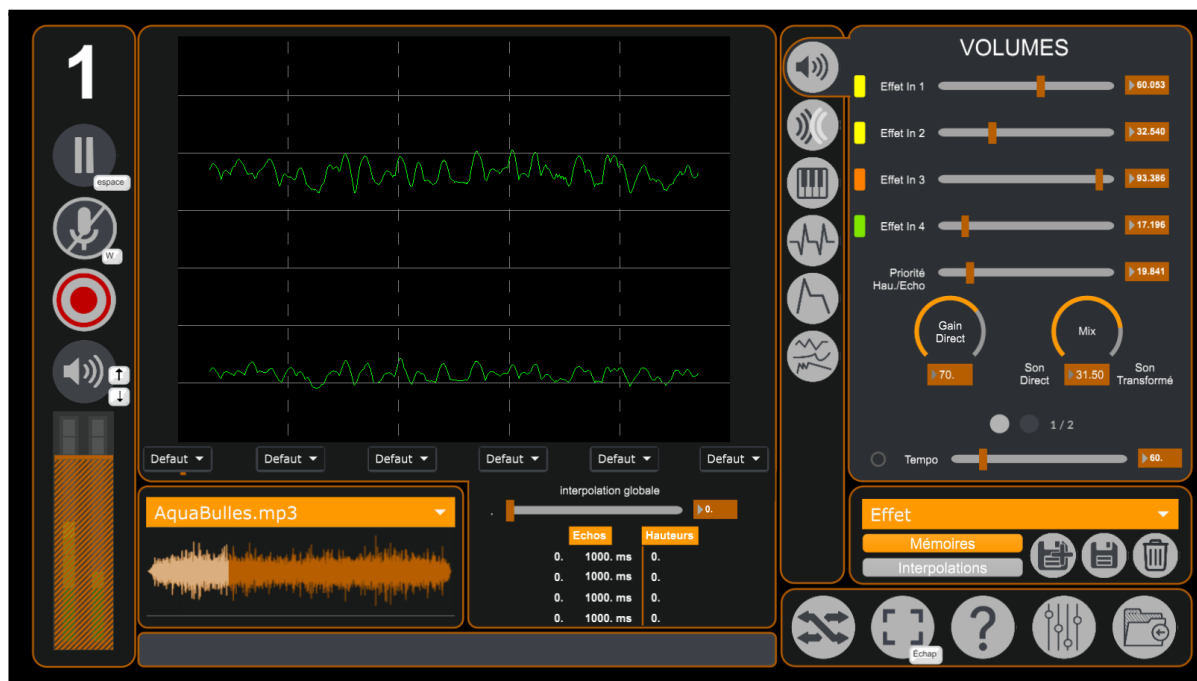


→ pour écraser une mémoire/interpolation



→ pour supprimer une mémoire/interpolation





Vous pouvez ensuite rappeler cette mémoire ou interpolation en la sélectionnant dans le menu suivant :



## 2. Interpolations

Une fois que vous avez quelques mémoires sauvegardées, vous pouvez créer et sauvegarder des interpolations. Pour créer une interpolation, il suffit d'utiliser les menus déroulants situés sous le visuels. Une interpolation est constituée de 6 *états*.



Vous pouvez voir un petit indicateur, qui permet de visualiser votre avancement au sein de l'interpolation.



Pour se déplacer au sein d'une interpolation, il faut utiliser le bouton "*interpolation globale*" :

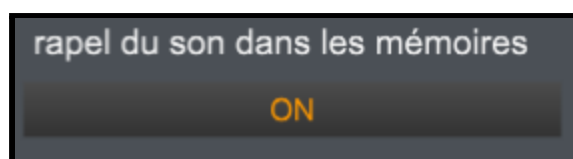


Le logiciel va donc générer toutes les valeurs intermédiaires pour passer progressivement d'une mémoire à l'autre.

Par défaut, l'interpolation rappelle les sons enregistrés avec les mémoires. Pour modifier ce comportement, rendez-vous dans l'onglet *Paramètres* accessible via ce bouton :



Rendez-vous ensuite dans le deuxième onglet *Pref* puis cliquez sur le bouton lié à "*rappel du son dans les mémoires*". Lorsqu'il est sur *ON* le son sera rappelé avec les mémoires, tandis que sur *OFF* le son ne sera pas rappelé avec les mémoires.



### 3. Plus d'options

Vous avez peut être remarqué la grille qui se superpose au visuel :



Cette grille permet d'effectuer des interpolations par catégories : les traces verticales correspondent aux états qui sont affichés en dessous, tandis que les lignes horizontales correspondent aux catégories visibles sur la droite.

Par exemple, si je clique sur l'emplacement indiqué sur la capture d'écran ci dessous :



J'appliquerai au visuel les paramètres de la catégorie *LFO (ECHO)* qui sont enregistrés dans l'état *Effet*.

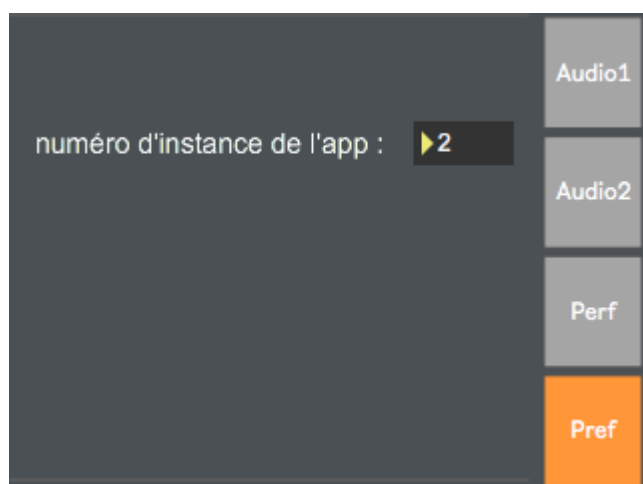
## Paramètres avancés

### Instances MonoReplay

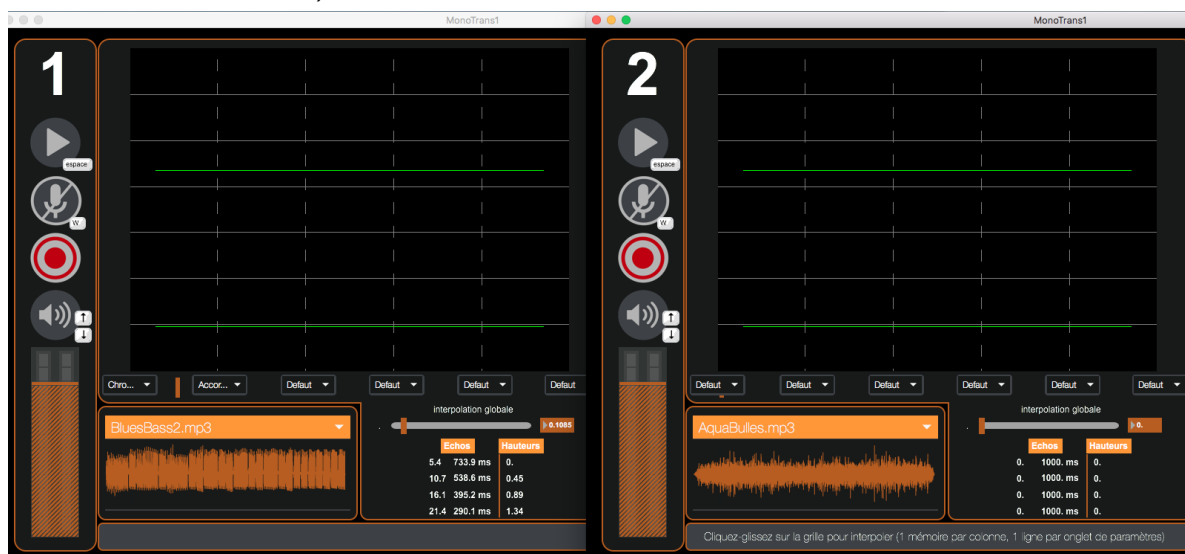
Comme pour MonoVueSon et MetaDJ, MonoReplay peut fonctionner sur une même machine en plusieurs instances et ce de façon indépendante :



Il suffit d'entrer dans le menu de paramétrage Audio de MonoReplay puis dans le sous menu préférence:



Et d'y entrer un numéro d'instance différent des autres (*par défaut une nouvelle instance s'ouvre avec le chiffre 1*):





## Annexe D

### Transcription de l'entretien semi-directif avec Nicolas Rivière

## Entretien semi-directif : Nicolas Rivière

### 1. Introduction :

Cet entretien s'intègre dans un travail de recherche portant sur la conception d'outils expressifs pour le contrôle temporel d'un enregistrement.

Dans ce travail, un modèle permettant d'ajouter, de paramétrer et de piloter plusieurs modes de contrôle temporel a été implémenté dans un logiciel. Ce logiciel a ensuite été utilisé par des artistes/compositeurs pour leurs créations. Pour évaluer le potentiel expressif de ce système, un entretien a été mené à la suite de cette expérience. En récupérant l'expérience subjective de chaque musicien avec cet outil nous pourrions conclure sur l'apport créatif de ce système par rapport aux autres logiciels de contrôle temporel. Une retranscription de cet entretien est disponible plus bas.

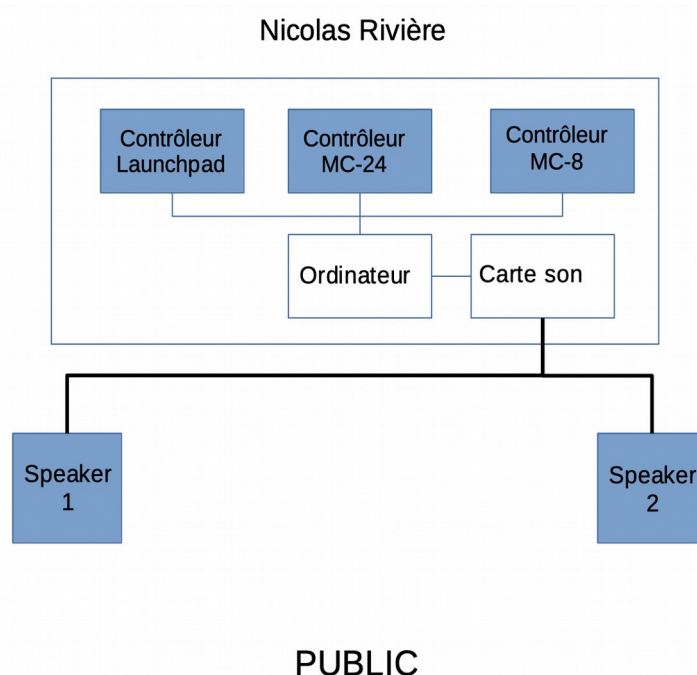


Figure 1: Dispositif utilisé pour jouer la pièce « SUPERNOME DIVA 2 - contemplation du vide »

Le compositeur et musicien Nicolas Rivière a utilisé le logiciel Mono-Replay pour composer et jouer sa pièce SUPERNOME DIVA 2 – contemplation du vide. Un plan de scène illustre le dispositif scénique utilisé lors des restitutions. Comme représenté figure 2 cette pièce est constituée de 3 parties :

- « VIDE » : Dans cette partie, le compositeur utilise le mode « Scrub » du logiciel pour contrôler la position de lecture. Il improvise avec tous les autres paramètres autour du segment 72 d'un enregistrement qu'il a préalablement étiqueté. Il joue par-dessus un enregistrement lu en parallèle.
- « VLASE » : Dans cette partie, le compositeur utilise le mode « Speed » du logiciel pour contrôler la vitesse de lecture d'un enregistrement qu'il a préalablement étiqueté. Il transforme le son à l'aide de l'enveloppe d'effet disponible dans Mono-Replay et des variables directement accessibles.



- « END » : Dans cette partie, le compositeur utilise le mode libre « Scrub » pour contrôler la position de lecture de l'enregistrement. Il improvise avec tous les paramètres de transformation disponible dans Mono-Replay.

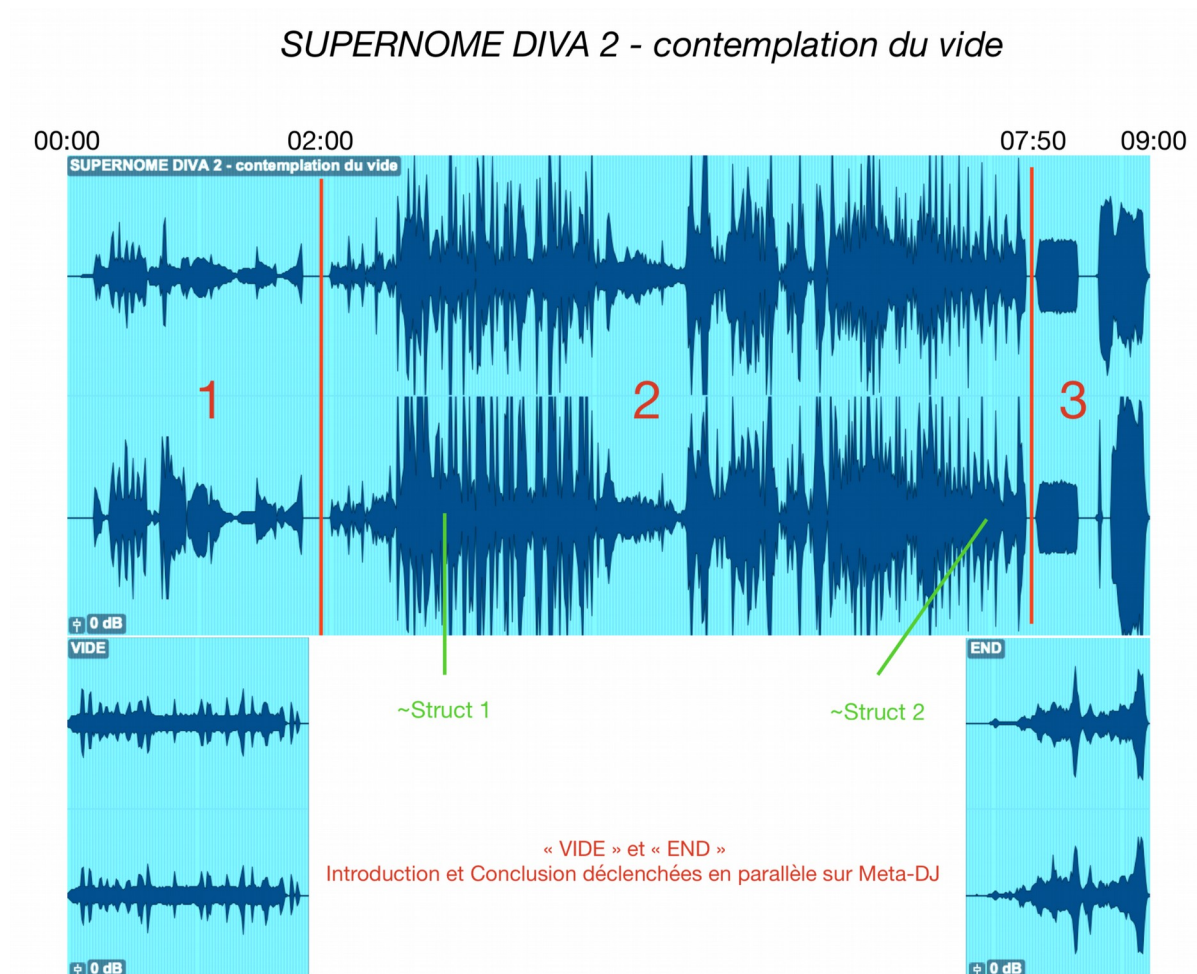


Figure 2: Représentation schématique de la pièce « SUPERNOME DIVA 2 - contemplation du vide » composée et jouée par Nicolas Rivière. La partie 1 "VIDE" est constituée d'un enregistrement manipulé en temps réel et d'un enregistrement joué en parallèle. La partie 2 "VLASE" est une valse enregistrée ou la vitesse de lecture est contrôlée de façon temps réel. La partie 2 "END" est composée d'un enregistrement manipulé et d'un enregistrement joué en parallèle.

Le dispositif utilisé pour jouer la pièce « SUPERNOME DIVA 2 – contemplation du vide » est représenté figure 1. Il est constitué de :

- Un contrôleur MIDI MC-8 composé de 8 faders
- Un contrôleur MIDI MC-24 composé de 24 potentiomètres
- Un contrôleur MIDI Novation LaunchPad Mini composé de 64 boutons
- Un ordinateur avec Mono-Replay et Méta-Dj

Les trois interfaces gestuelles permettent de contrôler en OSC les paramètres de Mono-Replay de la façon suivante :

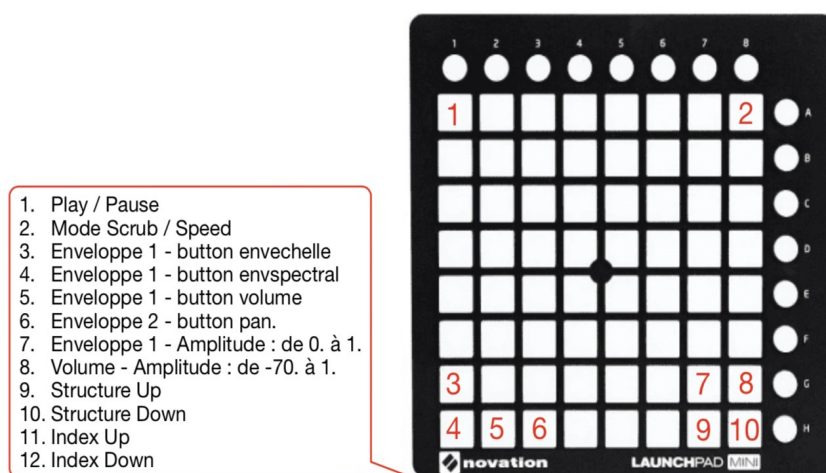


Figure 3 : Mapping utilisé par Nicolas Rivière pour contrôler les paramètres de Mono-Replay avec le contrôleur MIDI Novation LaunchPad Mini



Figure 4 : Mapping utilisé par Nicolas Rivière pour contrôler les paramètres de Mono-Replay avec le contrôleur MIDI MC-24

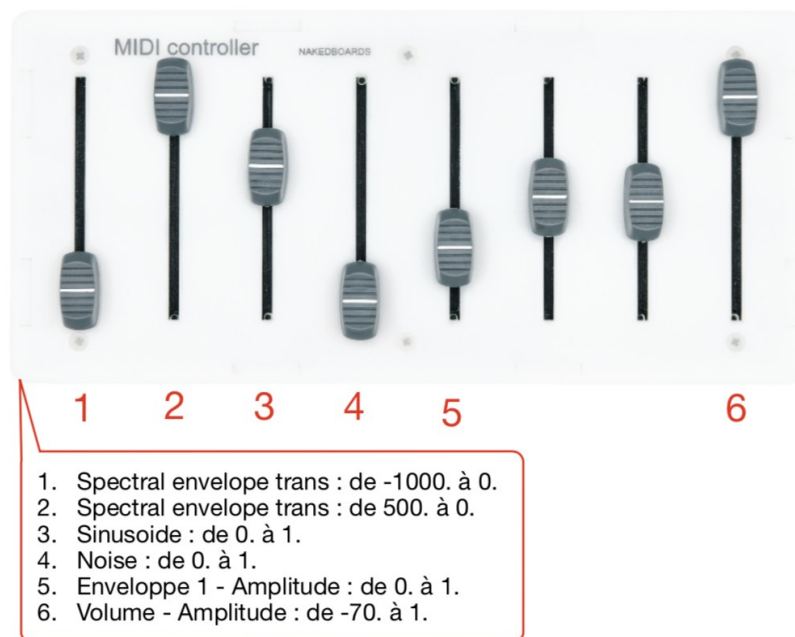


Figure 5 : Mapping utilisé par Nicolas Rivière pour contrôler les paramètres de Mono-Replay avec le contrôleur MIDI MC-8

## **2. Retranscription de l'entretien :**

*Question 1* : Comment utilises-tu le contrôle temporel dans ta création ?

*Nicolas* : J'aime bien envisager la musique que je fais comme une musique de « sample ». Il y a plusieurs échelles. Ça peut aller du micro-sample au maxi-sample. Dans ma création je sample des petites choses que je monte ensemble et que je viens ralentir ou accélérer, puis transposer. Je fais beaucoup de choses sur l'étirement temporel et le contrôle de la vitesse. Je viens dilater le temps, accélérer et ralentir les échantillons. Je contrôle aussi leur ordre d'apparition.

*Interviewer* : Utilises-tu des systèmes autonomes ?

*Nicolas* : Comme logiciels, j'utilise Pro-Tools, le GRM player et MaxMSP. Pro-Tools est mon master enregistreur dans lequel je viens poser toutes mes matières pour les assembler. A ce niveau-là, c'est plus du bidouillage. Avec le GRM player il y'a déjà beaucoup plus de jeux. J'ai mappé ça avec Max/MSP. Je contrôle un certain nombre de boucles que je ralentis et accélère. Je viens ensuite ajouter des effets pour créer des textures. J'exporte le tout dans Pro-Tools. Max/MSP est plus utilisé pour rejouer des choses de manière plus inédite. Je sais ce que je veux, je vais sur Max/MSP et je code ça. Je fais un mini patch. Au niveau des machines j'utilise l'analog rhythm d'Elektron avec des pads. Il y'a aussi des options de sampling qui me permettent de travailler une boucle avec précision.

*Question 2* : Comment penses-tu qu'on puisse améliorer ces outils pour les rendre plus expressifs ?

*Nicolas* : Ce qui est assez difficile c'est que l'expressivité dépend de chacun. Je me rends compte que tout est très limité mais qu'à la fois il y'a des possibilités folles. Il faudrait que je crée mon propre logiciel.

*Interviewer* : Si tu créais ton logiciel, qu'est que tu y ajouterais par rapport à ce qui existe déjà ?

*Nicolas* : Je joindrais la froideur et la précision de Pro-Tools dans l'édition au jeu du GRM Player. Ce que j'aime bien c'est la culture de l'audio dans Pro-Tools. Même si j'utilise un peu de MIDI dans Pro-Tools c'est vraiment fait pour manipuler l'audio. Tu vois ton schéma avec les différentes pistes... tu les superposes et tu vois ce que ça donne. J'aime bien ce côté froid, scientifique de « on colle des trucs on monte etc. ». Je prendrais ça de Pro-Tools avec un truc complètement différent qu'est le GRM player où tu ne peux pas vraiment être précis. Il y'a un truc très flottant dans le GRM Player où tu poses différentes boucles. Tu n'as pas 40 000 paramètres. Il y en a 4. Tu mappes avec Max/MSP et de là tu peux bidouiller à foison.

*Interviewer* : Si je comprends bien ce serait de prendre des outils qui existent déjà de façon séparée et de faire un outil qui mélange des fonctions intéressantes de Pro-Tools et du GRM Player.

*Nicolas* : Je me dis que je suis frustré de passer de l'un à l'autre et j'aimerais bien que tout soit ensemble. En même temps, je sais que dans ma pratique je veux avoir tout sur le bout des doigts et contrôler le son précisément. Je pourrais faire un patch Max/MSP ou utiliser un logiciel qui me permet de faire ce que je veux. On parlait de déstructurer une boucle de jungle. Je pourrais avoir Pro-Tools pour déstructurer ma boucle ou un GRM player avec MaxMSP pour bidouiller des choses. Je ne suis pas encore arrivé au point de me dire que j'ai un outil avec lequel je peux faire une boucle et la torturer dans tous les sens.

*Question 3* : Comment les interfaces gestuelles que tu as utilisées t'ont aidé dans ta création ?

*Nicolas* : C'est une question d'expressivité. C'est de pouvoir avoir tous les paramètres sous le bout de doigt et les gérer. J'ai un contrôleur MIDI où j'ai 8 faders , un contrôleur MIDI où j'ai 24 potentiomètres et un Launchpad. Chacun est mappé sur Mono-Replay. Sur les 8 fader, j'en utilise 6. Il y en a 4 qui contrôlent le menu vocodeur. Ils agissent comme des « filtres ». J'ai le contrôle de la transposition spectrale, j'ai le mix-sinusoïde et le mix-noise. Cela me permet de bien venir « morpher » mon son à certains moments. J'ai un fader qui me permet de contrôler le VCO et un fader qui me permet de contrôler le volume. J'utilise 3 potentiomètres sur les 24. Je les utilise pour contrôler les modes « Scrub » et « Speed ». J'utilise le mode « Scrub » dans les passages où il n'y a pas de métrique. Je me balade dans le son, je ne sais pas trop où je vais mais je sais que je peux sortir certains types de matières. Le mode « Speed » est là pour respecter des passages de composition que j'ai composés en dehors du logiciel et qui sont beaucoup plus métriques. Je ne vais pas réinventer le son mais juste le rejouer en modulant des effets.

J'ai un potentiomètre qui contrôle la vitesse du mode « Speed ». Je peux lire le son à la vitesse normale ou le ralentir. J'ai un potentiomètre qui me permet de contrôler la position de la tête de lecture en mode « Scrub ». Il y'a un potentiomètre qui me permet de passer de la vitesse normale de mon son au freeze total. L'autre potentiomètre me permet de me déplacer à la vitesse zéro. En tout, j'utilise quatre potentiomètres. Les deux derniers me permettent de contrôler la fréquence du VCO.

Sur le Launchpad j'ai un ensemble de boutons qui me permettent d'activer les boutons des VCO. J'ai un bouton qui me permet de déclencher l'ADSR préalablement défini. J'ai pris l'effet sidechain, c'est à dire que l'attaque est bloquée et que le volume remonte après. J'ai un Play, un bouton pour changer de mode, des boutons pour faire « index Up », « index Down », « Structure Up » et « Structure Down ». Je n'utilise ces boutons que dans le mode « Scrub » quand je ne peux me déplacer que dans un segment. Ça me permet de dériver dans mon son. Les structures c'est plus

pour les phases. Je commence mon morceau à un certain temps, je joue en mode « Scrub » et à certain moment je sais que je dois revenir au début donc je fais « Structure down » puis « Play ». Après il y avait la possibilité d'utiliser le Mini-méta... mais ça c'est moins mon mode d'expression. J'ai un mode de jeu « bourrin ». Je joue avec les faders de façon brusque. Je passe de zéro à un sur les temps.

*Interviewer* : Les interfaces que tu as utilisées sont très communes. Le mini-Méta est une interface prototype qui est une version plus petite du méta-instrument et qui nécessite beaucoup d'apprentissage.

*Nicolas* : Il faudrait que j'y passe du temps comme sur un vrai instrument. Il faudrait déjà que j'ai des modes de jeu qui me permettent de jouer avec ça. Il faut prendre en compte les caractéristiques physiques de l'instrument : la résistance des touches, la sensibilité des joysticks, la réactivité du capteur au mouvement... En fait c'est très sensible comme outil. Moi j'ai travaillé avec ces outils qui sont effectivement très communs car ils sont faciles d'utilisation. Il y'a pas trop de risque et il y a quelque chose d'assez simple. Alors qu'avec le Mini-méta il faudrait vraiment un apprentissage. Je pense qu'il faudrait des « cours ».

*Question 4* : Quels gestes musicaux t'évoque l'enregistrement que tu manipules ?

*Nicolas* : À la base j'étais parti sur une valse. Quand j'ai utilisé le logiciel j'ai essayé sur l'enregistrement d'Érik Satie. Il y'avait beaucoup de choses qui étaient pulsées mais pulsées de façon un peu classique, très rubato. Il y avait une temporalité assez valse, assez dansante et assez aérienne. Je me suis donc dit que j'allais faire une valse, mais une valse un peu acousmatique, un petit peu électronique. Je voulais lui donner une forme plus classique à trois temps avec une pulsation qui peut être jouée legato et ensuite y mettre des textures un peu acousmatiques. Et puis ça a évolué tout en restant une valse. Ce que ça m'a évoqué c'était d'avoir quelque chose d'assez quadrillé. Je me suis dit que j'allais utiliser Mono-Replay pour justement venir modifier la temporalité, l'étirer, l'accélérer etc.

Ce n'est pas très acousmatique ce que j'ai fait. J'ai ma propre culture et j'ai fait mon propre chemin à travers l'acousmatique. L'acousmatique, c'est l'art des sons et l'art d'écouter des choses très précises. J'aimerais le relier à quelque chose de plus populaire comme la culture du rythme ou la culture de la mélodie. Ce sont des choses que j'ai envie de garder car pour moi il y a une émotion qui vient de là. J'ai envie d'avoir un super ensemble qui intègre la culture des sons, la culture du rythme et la culture de la mélodie. Dans la culture de la rythmique j'aime bien quand ça tape fort. Si je tape fort dans ma rythmique il va falloir que le geste aille avec. C'est pour ça que je te dis que je suis un « bourrin » dans ma manière de manipuler mes faders. C'est plus des faders de texture, de filtre et de traitement du son. Dans les moments où je « scrub » et que je suis plus dans des recherches expérimentales, je vais jouer plus doucement avec les faders. Mais quand il y a du rythme, je fais des gestes binaires sur le temps. J'aime bien quand ça tape, quand d'un temps à l'autre je passe de quelque chose d'ultra aiguë à un gros kick avec du sub. Dans la musique que j'ai écouté de manière acousmatique comme la musique électronique très rythmée que je peux rallier au métal il y a une certaine culture du son qui me fait penser que c'est vraiment de l'acousmatique. En général j'aime bien les choses violentes. Si je mets du rythme, je vais devoir les rejouer de manière « violente » et être décisif sur mes choix. Il y'a un truc de la mélodie, de l'espace, de l'émotion et c'est là que je vais essayer d'approcher des choses avec plus de sensibilité.

Il y'a une intro, un mouvement 1, une transition, un mouvement 2 et une outro jouée de différentes manières. L'intro est très expérimentale et très « scrub ». Le mouvement 1 est une valse un peu énervée avec des temps forts que je veux marquer avec mes faders. La transition est un peu scrub expérimentale. Le mouvement 2 est quelque chose de beaucoup plus lent et beaucoup plus

mélodique mais avec de gros impacts. Cela me permet de garder le côté un peu intensif que j'avais sur la première partie et en même temps d'avoir des rejets que je peux utiliser pour ralentir des choses.

*Interviewer* : Pour résumer, tu dis que tu intègres complètement l'évocation du geste dans ta façon de rejouer tes samples

*Nicolas* : C'est ce que j'essaie de faire, oui. Cela étant, si je devais rejouer cette pièce je le ferais d'une autre façon. Je pense que je me suis trompé sur pleins de choses. C'est impossible de ne pas se tromper. Le geste permet de se mettre en péril et d'avoir ce côté « vas-y je peux foirer et faire de la merde ». J'ai envie d'avoir ce truc de pouvoir prendre des risques. Comme sur une batterie, si je tape fort, le son sera fort.

*Question 5* : Tes interfaces te permettent de faire des gestes binaires et assez marqués et en même temps elles te permettent d'être très précis. Est-ce qu'il y'a des choses qui te frustrant dans ces outils ?

*Nicolas* : Non pas du tout. j'ai les faders pour les mouvements verticaux , les potentiomètres pour les mouvements circulaires et les boutons. Je pourrais avoir des mouvements plus 3d mais ça ne me manque vraiment pas.

*Question 6* : Comment l'interface graphique de MR tu as aidé dans ta création ? Sachant que l'interface est très semblable à celle des Mono que tu connais déjà.

*Nicolas* : Je ne crois pas qu'elle m'ait aidé. L'interface graphique permet de comprendre la philosophie du logiciel. En explorant les menus, en voyant la page centrale et en voyant les segments défilés, tu comprends que tu peux te déplacer dans un segment. Bien sûr, cela aide à visualiser l'ensemble, mais je ne pense pas que ça m'ait aidé dans ma création. Ça m'a aidé à comprendre les fonctionnalités du logiciel. Une fois que j'ai parcouru les différentes fonctions j'ai quitté ça. Je ne regarde pas trop, j'écoute beaucoup. Je suis parti faire ma valse de mon côté. J'ai fait de la créa avec Mono-Replay mais pour des matières. Dans la manière de rejouer, les visuels sont plus des repères. Par exemple le problème qu'on a eu avec l'overdrive permet de vérifier où on en est en cas de problème. On est très mono centré par la vue. Tous les logiciels s'accompagnent de beaucoup de visuels. En général on se focalise sur le visuel. Si on rencontre un bug on va plutôt utiliser le visuel pour le débayer. C'est plus dans le but de comprendre quelles sont mes possibilités.

*Question 7* :Selon toi, est-ce qu'il y a des choses qui manquent à l'interface graphique ?

*Nicolas* : Comme ça je ne vois pas. Je pense que c'est clair en général. Toutes les applications Mono de la chaîne sont des outils très didactiques et très pédagogiques. Tu mets un gosse devant ce logiciel, tu lui expliques, et il va comprendre. Je pense que c'est bien. Ta fenêtre principale, elle est cool.

*Question 8* : Quels modes de contrôle temporel as-tu utilisé et pourquoi ?

*Nicolas* : « Speed » et « Scrub » . J'avais envie d'avoir deux aspects du contrôle temporel. Un contrôle plus expérimental , « ambient », « freeze » dans la recherche de texture. Je souhaitais pouvoir changer quand je voulais. Je souhaitais avoir un contrôle total dans le son, pouvoir aller dans le micro-temporel. Le « Speed » c'est plus un mode pour rejouer mon morceau un peu comme je l'avais prévu au départ en y ajoutant des effets et en pouvant le ralentir et l'accélérer davantage, et en le modifiant avec les outils disponibles.

*Question 9* : Personne n'a utilisé les modes tap tempo. C'est vrai que ça a été pensé plus comme une fonctionnalité qu'une réelle nécessité. Ça n'a pas été du tout utilisé. Est-ce que pour toi ce sont des choses qui pourraient t'intéresser dans ta création ?

*Nicolas* : Dans le mode tap tempo tu as le mode direct et parabolique. Il y'a des choses intéressantes à faire. Je pense que j'aurais pu les introduire dans ma pièce. Par exemple, « intro » je suis en scrub puis je passe en speed dans ma pièce... en fait je n'ai pas eu le temps.

*Interviewer* : J'ai l'impression que le mode tap tempo est essentiellement utilisé par les DJ. C'est très anecdotique, ce mode permet juste de caler la vitesse de façon plus rapide. Dans le finger drumming, les gens ont besoin d'avoir un contrôle assez direct. Peut-être que se limiter à l'utilisation de ce mode pour faire une création avec ça permettrait de voir les possibilités de ces modes ?

*Nicolas* : Il y a presque une pièce à faire pour chaque paramètre. Il y a tous les outils à disposition pour faire une pièce sur la transposition spectrale par exemple. C'est pour ça que je pense que j'ai fait des « erreurs ». Je pense que pour faire une pièce parfaitement incluse dans l'outil, un son n'a pas de sens sans le geste et le geste n'a pas de sens sans le logiciel. Il faudrait prévoir dans ta compo des sons qui passent au logiciel par le geste. Et que tout soit harmonieusement fait pour que tout prenne sens. Ma pièce on peut la jouer autrement.

*Question 10* : Tu penses que tu aurais pu faire le même travail autrement qu'avec Mono-Replay ?

*Nicolas* : Non, j'ai vraiment pensé cette pièce pour Mono-Replay.

*Interviewer* : Si tu devais faire exactement la même chose avec le même mapping gestuel et le même son qui sort, penses-tu que cela est possible avec un autre outil ?

*Nicolas* : Je pense que dans l'aspect « speed », le contrôle de la vitesse, je peux le faire très facilement ailleurs. Par contre les filtres je ne sais pas comment tu les as fait ... Je n'ai jamais croisé ces effets ailleurs. Dans le mode « Scrub » à la limite maintenant que je l'ai en tête, je l'ai compris et je pourrais le refaire mais ça me prendrait du temps.

*Interviewer* : C'est vrai que tu es aussi développeur d'instruments logiciels...

*Nicolas* : Un poil... Avec mes outils comme le GRM player et Pro-Tools je pense que ce n'est pas possible de faire tout ça. Avec le GRM player je peux avoir une modification de la vitesse mais il faut que je passe par Max/MSP pour le mapper. En revanche, je n'ai pas les textures et les filtres, je ne peux pas faire le scrub et je ne peux pas diviser en segment. Donc non, ce serait très « galère » de faire la même chose que j'ai faite avec Mono-Replay.

*Question 11* : C'est quoi les difficultés que tu as eues pour intégrer le logiciel à ton environnement de travail habituel ?

*Nicolas* : Je n'ai pas eu de réelles difficultés. Il y a eu des petits problèmes techniques qu'on a vus ensemble, mais je retrouve aussi ces problèmes dans le GRM player... je leur envoie des mails et c'est réglé... Non je n'ai eu aucun problème. J'ai l'habitude de faire des bounces dans des applications que j'emmène dans Pro-Tools pour les remonter dedans. J'ai l'habitude de faire des exports etc. Par contre, je pense que quelqu'un qui est dans Ableton et qui a l'habitude de tout faire dans un seul logiciel aura plus de difficultés, à moins d'avoir un Soundflower ou un Jack.

*Question 12 : Comment as-tu choisi les enregistrements que tu manipules ? Est-ce que tu as pris en compte la manière dont tu allais les manipuler ?*

*Nicolas : Pas trop... J'avais juste pris en compte que c'était une valse. J'avais envie de faire une valse par rapport à ce que je disais du logiciel pour jouer rubato et être dans l'expressivité. Je me disais qu'il me fallait du piano. J'avais envie de faire le contrepoint avec quelque chose très « à trois temps », de mettre des impacts forts. J'ai mis du piano et j'ai mis des impacts forts et ensuite j'ai bidouillé pour que ça sonne bien. Le piano c'est de la synthèse. Je crois que dans la première partie il y'a une rythmique que j'ai fait à l'Analog rythme et beaucoup de traitement dans Pro-Tools. Ensuite il y'a plein de sons d'un piano qui tombe dans l'eau. Il y'a un passage qui vient d'enregistrement de Mono-Replay que j'ai retransformé dans Mono-Replay.*

*Question 13 : Comment c'est passé ton expérience avec Mono-Replay ?*

*Nicolas : Bien. ... Je me suis senti limité seulement par rapport à moi-même. C'était juste un combat de faire les choses bien et de faire les choses jusqu'au bout. Par rapport à la technique, quand il y avait des petits problèmes, tu étais là et on a essayé de les résoudre. C'est vrai que lorsqu'on te demande « qu'est-ce que tu veux rajouter ? », du coup tu as envie de tout rajouter. En dehors de ça, du point de vue technique tu as été là.*

*Question 14 : Est-ce que tu penses que ce logiciel a apporté un truc en plus par rapport à tes outils habituels et ce serait quoi exactement ?*

*Nicolas : Ce serait le jeu. L'expressivité. Je pense que je fais une musique très écrite, très montée, très éditée. Je pense que je m'ouvre un peu plus avec tous mes autres concerts. J'essaie d'amener le geste. J'essaie d'être là et d'être instrumentiste. Il y'a un nouveau pan qui s'ouvre à moi et qui est le geste.*

*Question 15 : Quels sont selon toi les perspectives d'améliorations ? Comment on pourrait améliorer l'expressivité du logiciel ?*

*Nicolas : Tu connais mes retours... contrôle du panoramique, mais ça tu l'as rajouté, le vco du volume a aussi été rajouté. J'aimerais bien une fonction random des paramètres et une fonction random des segments. Tu mets « play » et ça passe du segment à un autre de façon aléatoire. J'aimerais bien aussi avoir un random sur les transformations spectrales à chaque fois que tu passes un nouveau segment. Ça peut être un mode à part .*

*Interviewer : Comment contrôlerais-tu cette fonction random ?*

*Nicolas : Tu pourrais vraiment avoir tout un menu là-dessus je pense. Ça te permettrait de créer une installation par exemple avec juste un « bip » qui serait rejoué 100 milliards de fois à différentes vitesses, différentes échelles et différents traitements. L'installation dure trois heures et tu n'as jamais le même son. Il y avait aussi toute la réflexion autour du beat-repeat, la possibilité d'avoir un contrôle temporel qui te permettrait de modifier des choses et de revenir sur quelque chose qui se déroule dans le temps. Ton sample est segmenté. Tu peux venir apporter des modifications pour définir quel sample vient à quel moment et à tout moment tu peux revenir sur ta boucle. Pour résumer, l'idée serait de déstructurer une boucle et d'avoir le contrôle temporel parfait pour revenir à un état précédent.*

*Question 16 : Est-ce que tu penses que tu as des modes de contrôle temporel qu'il serait sympa de rajouter dans Mono-Replay ?*



*Nicolas* : Moi, je n'ai que ces deux grosses améliorations là que je vois. Est-ce que ça fait partie du logiciel ? Je ne suis pas sûr. On parlait de comprendre ce que propose le logiciel. Il faut aussi savoir où il s'arrête. Je te dis ça maintenant, mais une fois que j'aurais le beat-repeat dans les mains je te demanderais certainement de rajouter quelque chose de plus... puis tu ne t'arrêtes jamais quoi et à la fin tu as un Ableton. Surtout que pour moi les Mono sont des outils assez simples. Ils explorent bien un concept. Elles sont très limitées mais c'est avec ces limites que tu donnes des contraintes et que du coup tu explores. J'ai fait une pièce avec Méta-Dj, il y a beaucoup moins de paramètres que Mono-Replay mais c'est intéressant de les explorer à fond. À qui est destiné ce logiciel ? ...je ne sais pas. Est-ce que c'est destiné aux musiciens ? Est-ce que c'est destiné aux personnes qui souhaitent appréhender la lecture d'un enregistrement et le comprendre ?



## Annexe E

### Transcription de l'entretien avec Paul-Denis Kingue Manjole

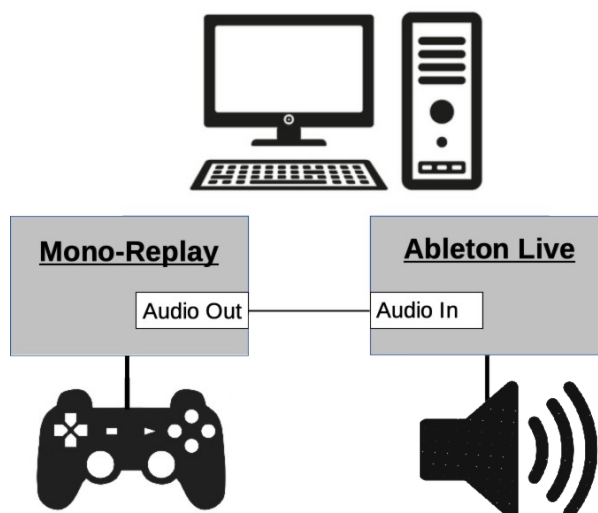
## **Entretien semi-directif : Paul-Denis Kingue Manjole**

### **1. Introduction :**

Cet entretien s'intègre dans un travail de recherche portant sur la conception d'outils expressifs pour le contrôle temporel d'un enregistrement.

Dans ce travail, un modèle permettant d'ajouter, de paramétrer et de piloter plusieurs modes de contrôle temporel a été implémenté dans un logiciel. Ce logiciel a ensuite été utilisé par des artistes/compositeurs pour leurs créations. Pour évaluer le potentiel expressif de ce système, un entretien a été mené à la suite de cette expérience. En récupérant l'expérience subjective de chaque musicien avec cet outil nous pourrions conclure sur l'apport créatif de ce système par rapport aux autres logiciels de contrôle temporel. Une retranscription de cet entretien est disponible plus bas.

Le musicien de Hip-Hop Paul-Denis Kingue Manjole, qui utilise comme nom de scène Pehoz, a pris en main ce logiciel pour créer des échantillons sonores depuis trois enregistrements afin de les intégrer dans un arrangement plus globale. Le dispositif qu'il a utilisé est schématisé ci dessous :



*Figure 1: Dispositif utilisé pour manipuler le logiciel Mono-Replay et enregistrer le son produit dans Ableton Live*

Ce dispositif est constitué d'une manette de jeu DualAnalog Thrustmaster branché à un ordinateur qui fonctionne sur Windows 8. Cette manette permet de contrôler Mono-Replay selon le mapping représenté figure 2. À l'aide du logiciel Jack Audio Connection Kit la sortie audio de Mono-Replay est reliée l'entrée audio du logiciel Ableton Live. Des pistes d'arrangements sont jouées dans Ableton Live et le son généré par Mono-Replay est directement enregistré dans le logiciel.

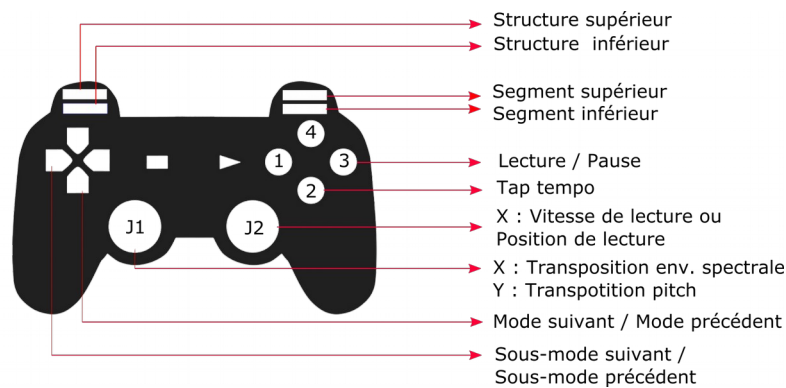


Figure 2: Mapping utilisé pour contrôler le logiciel Mono-Replay à l'aide d'une manette de jeu DualAnalog.

Le morceau «BAR MONOREPLAY »créé à l'aide de ce dispositif est disponible ici : [https://drive.google.com/file/d/1hZl27uR\\_eitd\\_tWzsvT\\_G6kILBwk\\_VAo/view?usp=sharing](https://drive.google.com/file/d/1hZl27uR_eitd_tWzsvT_G6kILBwk_VAo/view?usp=sharing)

## **2. Retranscription :**

*Question 1 :* Comment-utilises-tu le contrôle temporel dans tes créations ?

*Paul-Denis :* C'est une question technique. C'est quoi, le contrôle temporel ?

*Interviewer :* Il s'agit des méthodes permettant de contrôler temporellement le déroulement d'un enregistrement. Par exemple, dans le cas d'un DJ qui fait du scratch, le contrôle temporel qu'il effectue est l'action de bouger la tête de lecture de sa platine pour trouver la matière sonore qui l'intéresse en déplaçant la tête de lecture.

*Paul-Denis :* Pour moi c'est déplacer le « pied » du sample. Une fois que j'ai le « pied » de mon sample, je peux tout créer par la suite.

*Interviewer :* Le « pied », c'est à dire ?

*Paul-Denis :* Le pied, ce n'est pas forcément l'attaque dans le son ou le kick. C'est plutôt la note que j'ai repérée. Ce n'est pas forcément le début au pied (kick) mais c'est plutôt le début d'une note ou d'une fréquence.

*Question 2 :* Quels outils de logiciels utilises-tu pour faire du contrôle temporel ?

*Paul-Denis :* J'utilise Ableton Live de plusieurs manières. Je peux aussi utiliser ma MPC 2000 pour échantillonner comme je veux. Je coupe et je cale le début de l'échantillon sur le fameux « pied » avec ma MPC et ensuite je joue le tout à l'oreille. Avec Ableton, je place des marqueurs sur l'échantillon audio et à partir de ces marqueurs je peux choisir un point de commencement pour jouer mon sample.

*Interviewer :* Tu fais de l'étiquetage temporel avec la MPC et avec Ableton Live ?

*Paul-Denis :* Oui c'est ça.

*Interviewer :* Quelles sont les différences entre Ableton et la MPC 2000 pour faire de l'étiquetage ?

*Paul-Denis* : Avec Ableton Live il y a plus de visuels et c'est plus rapide. Il y a un grand écran et il y a une plus grande palette de fonctions pour bouger l'audio. Sur la MPC, il y a un plus petit écran et il n'y a pas forcément de forme d'onde. Il faut faire le découpage de l'échantillon par rapport au son. À chaque fois que tu tapes sur un pad tu fais une pré-écoute de l'échantillon. Ça prend donc plus de temps.

*Interviewer* : Comment évolue la vitesse de lecture à chaque fois que tu tapes sur le pad ?

*Paul-Denis* : À chaque fois que tu tapes sur le pad, la vitesse de lecture est celle déjà réglée. Tu peux régler la vitesse avec le fader.

*Interviewer* : Est-ce que tu peux changer la vitesse pendant la lecture ?

*Paul-Denis* : Non, tu ne peux pas. C'est un des désavantages de la MPC. Tu dois tout faire étape par étape. C'est une école différente de celle d'Ableton Live.

*Question 3* : Comment penses-tu que l'on puisse améliorer ces outils pour avoir plus d'expressivité ?

*Paul-Denis* : Il faudrait plus d'accessibilité aux marqueurs. Ce n'est pas forcément évident sur MPC ou Ableton Live de placer des repères temporels sur l'audio. Il faut longuement chercher dans le logiciel pour arriver à ces fonctions. Si cette fonction était plus accessible, on gagnerait plus de temps.

*Interviewer* : Donc selon toi, pour améliorer les outils de contrôle temporel, il faudrait améliorer la fonction d'étiquetage.

*Paul-Denis* : Oui, c'est ça.

*Question 4* : Comment s'est passée ton expérience avec Mono-Replay ?

*Paul-Denis* : C'était vraiment cool. Je suis un gamer donc faire du son avec une manette c'est vraiment sympa. Ça casse un peu le côté sérieux des logiciels pour produire du son. Ça amène une interface commune, la manette de jeu, dans un logiciel de son. Ça détend un peu les neurones.

*Interviewer* : C'est donc l'interface qui t'a vraiment marqué. Si on se focalise sur le logiciel dans sa globalité, quel est ton ressenti ?

*Paul-Denis* : C'est beau, c'est intuitif. La prise en main prend peut-être un petit peu de temps au début pour les gens qui n'ont pas l'habitude de l'informatique. Après, c'est assez facile d'utilisation pour ouvrir le projet, gérer les effets... Je n'étais pas trop perdu. La seule difficulté que j'ai eue c'est quand il a fallu que je fasse la mise à jour de Max pour ouvrir MonoMap. Concernant le logiciel en soi, comment l'utiliser, enregistrer, mettre les effets... Ça n'a pas été difficile. J'ai l'impression de découvrir un nouveau logiciel de son un peu comme quand j'ai découvert Ableton Live.

*Interviewer* : Donc ça s'est plutôt bien passé ?

*Paul-Denis* : Oui !

*Question 5* : Comment as-tu choisi les enregistrements que tu voulais manipuler ?

*Paul-Denis* : Je sample beaucoup de sons vintages. J'ai pas mal de palettes de sons que j'aimerais bien sampler. Quand j'ai écouté ces sons, je me suis dit « pourquoi pas ? ». Je savais déjà pourquoi je voulais sampler ces morceaux-là : pour la drum et pour la voix.

*Interviewer* : Tu as samplé un morceau de Nina Simone et... ?

*Paul-Denis* : Et un morceau de Three Degress. C'est le morceau « My baby cares for me » de Nina Simone. Ma fille est fan de ce son.

*Question 6* : Est-ce que tu serais arrivé au même résultat avec un autre logiciel que Mono-Replay (MR)?

*Paul-Denis* : Non. Ce que j'ai apprécié dans MR c'est de pouvoir faire un reverse et gérer l'enveloppe en même temps ; ce que l'on ne peut pas faire dans Ableton. Tu ne peux pas reculer et jouer les effets en même temps. J'ai joué le rythme avec la manette. J'aurai peut-être pu refaire la même chose avec un autre dispositif mais ça m'aurait pris beaucoup de temps.

*Interviewer* : Si tu voulais refaire un reverse avec Ableton tu aurais dû définir chaque position de ton reverse avec des automatisations...

*Paul-Denis* : Oui, c'est ça.

*Question 7* : Qu'est-ce que ce logiciel t'a apporté en plus de tes outils habituels ?

*Paul-Denis* : Le fait de pouvoir manipuler plusieurs effets en même temps. Dans les logiciels habituels, tu peux faire une boucle et gérer les effets à côté mais le rythme reste le même. Quand tu fais des samples de HipHop, tu es calé sur la même boucle et tu rajoutes les effets à la fin. Là, avec Mono-Replay je peux gérer un rythme et un effet en même temps. L'idée du reverse je l'ai eu à la fin. Je me suis rendu compte que le son sonnait mieux en arrière et quand il était pitché. Trouver cet effet m'a pris deux-trois minutes. J'aurai pas pu avoir ce déclic-là sur Ableton. Pouvoir gérer le rythme, les effets et les enveloppes tu ne peux pas le faire sur Ableton. C'était vraiment bien de pouvoir le faire sur MR.

*Question 8* : Quel mode de contrôle temporel as-tu utilisé ? Il s'agit du premier onglet en haut ...

*Paul-Denis* : J'ai utilisé le mode « speed » et le mode « speed freeze ». Pour la drums j'ai utilisé la loop. Après j'ai joué avec l'enveloppe.

*Interviewer* : Donc tu as utilisé les modes « speed direct », « speed freez » et « loop-b ».

*Paul-Denis* : Oui, c'est ça.

*Question 9* : Pourquoi n'as-tu pas utilisé les autres modes ?

*Paul-Denis* : Parce que je ne le sentais pas. C'est peut-être moins mon style de production. Je ne veux pas trop déformer le son. J'aime bien le déformer pour que ça devienne une autre sonorité... en restant à la limite du genre hip hop. J'ai utilisé les autres effets. Ils étaient bien mais pas trop utile dans ce que je voulais faire.

*Question 10* : Quelles ont été les difficultés pour intégrer MR dans ton environnement habituel de travail ?

*Paul-Denis* : C'était surtout pour relier Mono-Replay à Ableton. Ce n'était pas automatique. Avec Reason, par exemple, on peut se mettre en mode « rewire ». Là, ce n'était pas inclus dans le logiciel et il fallait utiliser le logiciel Jack.

*Interviewer* : Comme le format « application » ne s'ouvre pas en Vst c'est un peu difficile de tout installer pour que ça marche dans Ableton live. Tu fais tout passer dans Ableton ?

*Paul-Denis* : J'ai joué dans MR et ensuite je récupère les enregistrements que j'utilise dans Ableton.

*Question 11* : On va parler de l'interface graphique. Il s'agit de ce que tu vois et de comment c'est organisé. Y a-t-il des éléments qui t'ont permis de mieux comprendre comment créer des matériaux intéressants avec MR ?

*Paul-Denis* : Déjà la couleur ; c'est jaune et j'aime bien. De toute façon dans le son tous les sigles sont les mêmes. Le rouge pour « rec », dans les icônes tu as des « waves » et des graphes. Vu que les sigles sont utilisés partout ça m'a aidé à comprendre.

*Interviewer* : Est-ce que les barres, les segments, t'ont aidé ?

*Paul-Denis* : Au début pas trop. C'est juste en bougeant la manette que tu comprends le lien avec ces traits. Normalement ces traits s'ajoutent manuellement. Si tu as une barre c'est que tu l'as voulue.

*Interviewer* : Ok. La segmentation a été faite avec l'éditeur de segment. C'est une proposition de segmentation qui est faite donc ce n'est pas forcément une segmentation que tu as créée segment par segment.

*Question 12* : Est-ce que tu penses que des informations manquent sur cette interface ?

*Paul-Denis* : Peut-être un truc pour accéder aux informations facilement. Un truc d'arborescence un peu comme sur Ableton. Un truc sur le côté qui peut te montrer un peu où sont tes projets, ça serait bien. Après j'avoue que je n'ai pas trop réfléchi à ça sur le moment.

*Question 13* : De façon générale c'est quoi ton ressenti sur cette interface ? Est-ce que ça a été clair pour toi ? Qu'est que ça t'a évoqué la première fois que tu l'as vue ?

*Paul-Denis* : Ça ne m'a pas forcément évoqué quelque chose pour faire des samples mais plus pour faire des enregistrements. Vu qu'il y a un gros bouton « Rec » et que la forme d'onde est affichée je me suis dit que c'était quelque chose pour faire de l'enregistrement ; pas pour du sample ou de la compo mais que c'était plutôt un outil de manipulation audio.

*Question 14* : Maintenant on va parler de l'interface gestuelle, la manette. Comment ça t'a aidé dans ta création ?

*Paul-Denis* : C'est trop bien (rires). Ça change, c'est une super idée. Pour moi c'est la meilleure partie, franchement. C'est un vrai plus. En général, quand tu as une manette dans la main c'est que tu vas passer un bon moment. Moi j'ai toujours vu la musique comme un jeu, je le dis tout le temps. Pour moi Ableton c'est comme un logiciel de jeu. Et là, la manette ça retranscrit carrément l'idée de jouer. Ça confirme ce que je pensais.

*Question 15* : Si tu pouvais choisir une autre interface pour contrôler le logiciel, laquelle choisirais-tu ? Par exemple admettons qu'il y ait une fonction de MIDI LEARN comme sur Ableton avec



laquelle tu pourrais très rapidement brancher n'importe quelle interface et faire du mapping avec. Dans cette hypothèse-là quelle interface utiliserais-tu ?

*Paul-Denis* : En vrai c'est pas mal comme ça.

*Interviewer* : Je parle de l'interface gestuelle et pas de l'interface graphique. Si tu pouvais brancher une autre interface gestuelle, quelle interface tu choisirais ? Entre toutes celles que tu as l'habitude d'utiliser.

*Paul-Denis* : L'interface « Maschine » mais je ne sais pas pourquoi. C'est au niveau esthétique. C'est bien d'avoir un truc esthétique en tant que contrôleur. C'est stupide, hein ? Ça n'a pas de lien avec le fait de faire de la bonne musique ou pas mais pour moi c'est important d'avoir quelque chose de beau.

*Interviewer* : Je pense un peu comme toi.

*Question 16* : J'ai une question un peu bizarre. Dans ma thèse, en introduction, je parle de la théorie d'un chercheur qui s'appelle Denis Smalley. Selon lui, à chaque son enregistré qu'on entend on y associe un geste. Il propose ensuite différents degrés de « représentation gestuelle ». Si tu entends un enregistrement de piano tu vas aussi imaginer le geste du pianiste. En imaginant ce geste tu vas augmenter ton écoute. La théorie derrière c'est de dire qu'à chaque enregistrement entendu un geste est évoqué. Moi j'essaie de savoir si on tente de reproduire ce geste quand on manipule un enregistrement. Par exemple quand je contrôle un enregistrement de piano en le segmentant. Est-ce que je vais essayer de le contrôler avec le geste du pianiste ou est-ce que je vais aller chercher un geste qui n'a rien à voir ?

*Paul-Denis* : Non, carrément en fait tu imites. Enfin moi quand je découpe des sons de piano, au lieu de jouer comme ça ...(appuie sec sur le pad de la MPC) je vais faire comme ça ... (appuie plus doux avec liaison entre les doigts). Si c'est une guitare je vais faire plus ça (imitation du geste de cordes pincées sur la MPC). Je viens de le capter là mais c'est vrai !! Même quand je fais des lives sur scène, quand je manipule des samples de voix par exemple, j'utilise mes deux mains car c'est un peu comme balancer des « vibes ».

*Interviewer* : Admettons que tu manipules un enregistrement de violon par exemple, quel geste ferais-tu ?

*Paul-Denis* : J'appuierais plus comme ça (geste de frottement sur le pad de la MPC). Quand tu utilises un enregistrement tu t'appropries quelque chose d'un musicien. Tu essaies de ressentir le truc. Tu te dis que tu utilises un piano donc tu essaies d'avoir le groove d'un piano. Sans t'en rendre compte tu vas entendre le piano et tu vas te prendre pour un pianiste. Si tu entends une guitare tu vas te prendre pour un guitariste.

*Interviewer* : Moi quand j'entends un son je m'imaginer un geste. Denis Smalley va un peu plus loin. Par exemple si tu prends un son de synthèse ou de la musique de drone dans laquelle tu as de longues nappes sonores qui durent longtemps, tu ne sais plus trop quel instrument joue ce son. Dans ce cas tu n'imagines plus le geste de l'instrumentiste mais plutôt l'évolution physique d'un paramètre du son.

*Question 17* : Quel geste par rapport à l'enregistrement que tu as choisi t'as été évoqué ? Si jamais tu penses à un geste que tu pourrais faire pour rejouer cet enregistrement, quel serait-il ? Par exemple pour le sample de Nina Simone, est-ce qu'il y'a un geste que tu aimerais utiliser pour rejouer cet enregistrement ?

*Paul-Denis* : Dans le vide ce serait des accords de piano.

*Interviewer* : Donc des gestes de pianistes.

*Paul-Denis* : Oui.

*Question 18* : Est-ce que tu peux parler de l'éditeur de segment ?

*Paul-Denis* : C'est vrai que cet éditeur de segment là c'est quelque chose... C'est complexe quand même.

*Interviewer* : Quand tu es sur « edit » tu entends le son de là où tu pointes avec la souris. Mais là tu n'entends pas car tu n'es pas sur la bonne sortie audio.

*Paul-Denis* : Ce n'est pas que c'est compliqué ...

*Interviewer* : C'est un peu complexe effectivement...

*Paul-Denis* : Ce qui peut m'effrayer c'est ça... (Il montre la vision globale du morceau avec tous les marqueurs qui apparaissent dessus) Le truc c'est qu'il y'a trop de segmentation. C'est l'analyse qui fait ça. Mais...

*Interviewer* : Là c'est une analyse par pulsation. L'interface affiche toutes les pulsations du morceau ... ça fait beaucoup. C'est peut-être pas très utile...

*Paul-Denis* : Moi ça me fait peur quand il y a trop de traits. Après il faut peut-être changer l'ergonomie. Peut-être que c'est les couleurs qu'il faut changer. Oui, je pense qu'il faut changer les couleurs.

*Interviewer* : C'est vrai que pour ajouter un segment c'est compliqué car il faut sélectionner le mode « edit » etc. Par rapport à Ableton c'est peut-être moins intuitif. Sur Ableton tu peux faire deux clics pour ajouter un segment. Tu sélectionnes un endroit puis quand tu cliques sur « + » tu ajoutes un segment à cet endroit. C'est vrai que ce n'est pas assez clair. Je l'ai conçu un peu pour me faciliter la tâche. Ça été conçu par moi et utilisé par moi...

*Paul-Denis* : Pour moi il y'a trop d'informations sur la forme d'onde. Après le reste en bas ça va.

*Question 19* : Si on sort de l'éditeur de segment et qu'on regarde toute l'interface, quelles ont été tes difficultés ?

*Paul-Denis* : Ça a été de comprendre les différents effets et sous-effets.

*Interviewer* : As-tu utilisé l'aide contextuelle ?

*Paul-Denis* : Oui mais je ne me suis pas trop focalisé dessus tout le temps. C'est comme des sous-titres de film. Tu regardes seulement si ça te permet de comprendre un peu plus.

*Question 20* : On est arrivé à la fin de l'entretien. On a abordé les thèmes souhaités. On va parler des perspectives du logiciel. Quelles sont les perspectives d'amélioration de MR selon toi ?

*Pehoz* : Plus d'accessibilité dans les menus pour importer les audios. Le menu pour accéder aux audios serait mieux sur le côté et sous la forme d'arborescence. La partie contrôle rythmique devrait prendre une plus grande place. Il faudrait faire une partie import-export des fichiers à part.

*Interviewer* : La grosse amélioration que tu verrais concernerait donc plutôt l'interface graphique ?

*Pehoz* : Oui et plus de couleurs pour les segments. C'est un peu trop riche en information.

*Question 21* : Par rapport aux modes de contrôle rythmique, est-ce que tu imaginerais des modes supplémentaires ?

*Paul-Denis* : Peut-être des modes pour faire des sidechains, faire osciller des paramètres de contrôle depuis l'audio. Peut-être faire varier les modes de lecture autour des boucles pour aller plus loin. Il y'a beaucoup de possibilités pour trouver d'autres idées dessus. Après je n'utilise pas trop d'effets dans mes sons donc...

*Interviewer* : Peut-être faudrait-il rajouter un générateur d'effets ?

*Paul-Denis* : Oui, pourquoi pas des effets comme le phaser, le delay etc. ? Ça peut être cool.



Annexe F

Transcription de l'entretien avec  
Basile Chassaing

## **Entretien semi-directif : Basile Chassaing**

### **1. Introduction :**

Cet entretien s'intègre dans un travail de recherche portant sur la conception d'outils expressifs pour le contrôle temporel d'un enregistrement.

Dans ce travail, un modèle permettant d'ajouter, de paramétrer et de piloter plusieurs modes de contrôle temporel a été implémenté dans un logiciel. Ce logiciel a ensuite été utilisé par des artistes/compositeurs pour leurs créations. Pour évaluer le potentiel expressif de ce système, un entretien a été mené à la suite de cette expérience. En récupérant l'expérience subjective de chaque musicien avec cet outil nous pourrions conclure sur l'apport créatif de ce système par rapport aux autres logiciels de contrôle temporel. Une retranscription de cet entretien est disponible plus bas.

Une création intégrant Mono-Replay a été commandée au compositeur Basile Chassaing. La philosophie particulière du logiciel et des incompatibilités d'intégration dans l'environnement de travail du compositeur ont fait que cette création n'a pu aboutir. Pour comprendre quels aspects du logiciel ont été un frein à ce travail nous avons néanmoins mené un entretien semi-directif de la même manière que les précédents.

### **2. Retranscription :**

*Question 1 :* Comment utilises-tu le contrôle temporel dans tes créations ?

*Basile :* Pour le moment je l'ai utilisé pour une pièce, essentiellement via Max/MSP. C'était sur des petits fichiers de sons assez courts. Il y avait un mouvement sur la tête de lecture du fichier son. Je plaçais la tête de lecture à certains endroits et faisais varier cette position. Je faisais varier la fenêtre de lecture à l'intérieur du fichier son. Il y avait un jeu là-dessus : c'était sur un fichier son puis un autre. Il y avait des déclenchements un peu aléatoires de la lecture des sons puis une manipulation du fichier. C'était avec des capteurs de gestes. En gros c'était avec des accéléromètres et des gyroscopes. La position de vitesse déterminée par un geste des deux mains permettait de définir la plage de lecture du fichier son : à chaque fois que l'interprète mettait ses mains dans une position particulière, cela changeait aléatoirement le fichier son parmi une banque de sons. Le processus continuait dans cet ordre-là.

*Question 2 :* Quels outils logiciels tu utilisais pour ça ?

*Basile :* J'utilisais Max/MSP et l'objet groove.

*Question 3 :* Comment penses-tu qu'il soit possible d'améliorer ces outils pour avoir plus d'expressivité ? Je suppose que tu utilises d'autres logiciels que Max/MSP pour étirer et compresser des segments...

*Basile :* Je ne fais pas trop d'étirement parce que souvent il y a des artefacts et on les reconnaît très vite. Ça brouille la perception. Une oreille un peu connaisseuse va comprendre le time-stretch. Généralement ça me pose problème. En revanche, je compresse souvent le temps. On entend moins de time-stretch. C'est lié à la fréquence d'échantillonnage... C'est quelque chose que j'ai retrouvé dans Mono-Replay à certains moments. Quand tu scrub à certains moments tu entends vraiment ces artefacts.

*Question 4 :* Dans Mono-Replay on utilise un moteur audio SuperVP. Il y a peut-être des solutions dans le paramétrage de l'objet... Comment penses-tu qu'il soit possible d'améliorer ces outils ?

*Basile* : Ce serait de trouver de meilleures façons de ne pas entendre ces artefacts. C'est vraiment le truc qui m'a frappé tout de suite. Le time-stretch, c'est quelque chose de compliqué. C'est vraiment la seule chose.

*Question 5* : Comment s'est passé ton expérience avec Mono-Replay ?

*Basile* : J'ai essayé de mettre tout de suite les mains dans le cambouis d'une certaine manière. J'ai voulu tester le rendu sonore tout de suite. d'abord j'ai commencé par utiliser le premier projet. La version avec piano de Satie. Là-dessus j'ai essayé un peu tous les modes pour découvrir comment le fonctionnement du logiciel de manière rapide. Dans un second temps j'ai essayé d'ajouter un son à moi pour voir comment je pouvais indexer et comment l'indexation fonctionnait.

*Question 6* : Est-ce ça s'est plutôt bien passé ? Est-ce que ça a été compliqué ?

*Basile* : Ça a été compliqué par rapport à ma configuration personnelle. Je n'ai pas le bon MacOS et je ne peux pas trop faire de mises à jour pour des raisons que je tairai ici (rires). J'ai eu des réticences pour mettre à jour ce système. j'ai dû trouver des solutions un peu alternatives et un peu compliquées ; ce qui a un peu retardé l'utilisation du logiciel. Mise à part ça il y a eu des problèmes de versions aussi. On a échangé plusieurs versions. À la fin j'étais un peu perdu. Je me suis posé la question sur quelle version je suis et comment je peux vérifier ça. En écoutant les exemples du premier projet, j'ai trouvé que le logiciel fonctionnait relativement bien. Le Satie marchait bien, le marimba marchait moins bien. Sans rentrer dans le détail sur l'indexation, c'est compliqué de comprendre comment ça fonctionne. D'abord, l'interface n'est pas toujours très lisible. Quand tu es en mode « édition » ou en mode « zoom », le fait de ne pas pouvoir faire dérouler ton fichier dans les deux modes est dommage. Tu ne peux le bouger de gauche à droite qu' en mode « zoom » et pas en « edit ». Parfois, tu as besoin de voir ce qu'il y a avant et après. Je n'ai pas réussi à faire fonctionner la méthode d'analyse. Je n'ai peut-être pas tout à fait compris.

*Interviewer* : La méthode d'analyse est un objet node.js. Il est nécessaire d'avoir les droits administrateurs pour utiliser ces paquets-là en version compilée. Il faut l'installateur pour l'utiliser sans avoir de problèmes de sécurité...

*Question 7* : Comment as-tu choisi les enregistrements que tu souhaitais manipuler ?

*Basile* : Ce que je faisais avant c'était de manipuler segment par segment. Je manipulais le segment 1 puis le segment 2. Finalement ce que fait Mono-Replay c'est la même chose mais tu peux vraiment composer tes enregistrements. J'ai essayé d'un peu indexer sur plusieurs notes et avec des périodes un peu plus grandes. Je l'avais tellement composé qu'il sonnait bien d'une certaine manière mais dès que je commençais à vouloir le rejouer ça sonnait moins bien que ce que j'avais déjà composé sur Reaper. Pour ce type de fichier ça ne marchait pas vraiment. C'était assez complexe parce qu'il faut qu'il y'ai un certain degré d'ouverture du matériau sonore : il ne faut pas qu'il soit trop composé parce que sinon après il est tellement composé que dès que tu le manipules tu t'éloignes de ce que tu avais en tête. Ça ne marche pas, notamment pour les freeze. Parfois il y a certaines choses qui marchent bien et parfois moins. Les sons de piano sur le Satie ça marche très bien. Le son est beau. Tu ne te dis pas que c'est un fichier transformé. Tu ne te dis pas que c'est un freeze. Tu te dis c'est un son. Sur le marimba ça fait moins ça déjà. Sur les sons que j'ai utilisés, il y a eu des trucs un peu étranges par moment. Je me suis dit après que j'allais essayer sur de la voix pour voir comment tu pouvais la distordre et là, pareil. J'ai pris une voix radiophonique, je l'ai rentrée dedans et j'ai fait quelques marqueurs en essayant de jouer sur quelques syllabes. Mais après ça devient un instrument. Il faut vraiment l'envisager comme un instrument. J'avais pour projet de me dire que la pièce sur laquelle je travaille c'est pour du geste. Ce dont j'avais envie, c'était

d'associer du geste pour piloter une voix :qu'il y ait un lien un peu avec la langue des signes, faire un lien entre le geste de la langue des signes et le son ; que le son ne vienne pas de la voix mais du geste. Est-ce que c'est vraiment l'outil approprié ? Est-ce que je veux vraiment étirer le son ? Je suis toujours en questionnement sur comment je l'utilise et si je l'utilise en fait ?

*Question 8 :* Est-ce que tu penses que MR pourrait apporter quelque chose de plus à tes outils habituels ?

*Basile :* En l'état non. Dans le sens où comme tu le disais tout à l'heure, comme c'est une application externe, ça demande un boulot assez important pour l'intégrer dans une architecture Max/MSP qui est ce que j'utilise. Du coup il faut ouvrir un patch Max/MSP et ouvrir plusieurs applications si tu as besoin de plusieurs Mono-Replay. Ça te fait plein de fenêtres à ouvrir, et cela accroît le nombre d'erreurs possibles. Il faudrait pouvoir donner au musicien, au performeur les outils pour qu'ils puissent répéter de façon autonome. S'ils ont 10 000 applications à ouvrir, c'est l'enfer... ça rajoute beaucoup d'éléments dans la chaîne. Fatalement ça éloigne une potentielle reprise de la pièce. Une fois que tu connais un outil dans lequel il y a suffisamment de sous-outils que tu peux exploiter et dans lequel tu peux faire tes propres patch, tu ne vas pas chercher à en développer 10, 20... et les mettre en relation. Ça devient des usines à gaz et c'est une source de bugs beaucoup plus importants. Je pourrais avoir envie d'utiliser cet outil comme un objet Max/MSP, un sous patch qui puisse intégrer un patch de concert ou de répétition. Je pense que ce qui est intéressant dans MR c'est plutôt le côté indexation et de jouer dessus. Je ne vois pas l'intérêt d'avoir toutes les autres fonctions comme le vocodeur par exemple dans ce grand outil qu'est Mono-Replay. Si j'avais une chose à exporter pour mes propres projets ce serait la partie indexation. Le fait que ce soit une indexation automatique, c'est pas mal.

L'intérêt de Mono-Replay serait peut-être d'avoir un truc plus simple que MUBU for max et proposer une segmentation et des modes de jeu. Ce qui m'intéresse c'est la segmentation. Tu insères le fichier et tu demandes une indexation sur les transitoires d'attaque. C'est un gain de temps énorme vu que du coup tu ne le fais pas toi-même à la main.

Pour pas être limité d'un point de vue création je virerais tout ce qui est vocodeur. Je ne mettrais pas le moteur audio dans Mono-Replay.. Sur certains aspects c'est mieux de passer par SuperVP et ainsi de ne pas avoir à choisir le moteur audio. Si j'avais fait comme tu dis, j'aurais utilisé ces aspects sans le moteur audio. J'aurais choisi groove comme je ne connais rien d'autre. J'aurais essayé, ça n'aurait pas bien sonné et j'aurais dit « bon...j'aurais tout essayé » alors que ça aurait seulement été un problème de moteur audio.

*Question 9 :* Comment cette interface graphique a pu t'aider dans ton utilisation du logiciel ?

*Basile :* C'est assez clair dans les onglets etc. Il y a juste une chose qui m'a gêné. Sur la fenêtre de segmentation, il n'y avait plus l'aide contextuelle. ... il me semble. Sinon sur l'interface globale je n'ai pas grand-chose à dire. Concernant l'utilisation de l'application, ce serait bien de pouvoir choisir les canaux pour l'OSC par exemple. En l'état, je ne savais pas quels étaient les canaux choisis et c'était compliqué pour récupérer les informations. Ça serait pas mal de pouvoir le changer si jamais il y a un conflit sur une autre application.

*Interviewer :* Pour résumer, ton ressenti global sur l'interface graphique est que c'est assez clair et que tu aimerais ajouter l'affichage du port OSC et sa configuration.

*Basile :* Oui et ce que je disais tout à l'heure sur la fenêtre de segmentation. Elle pourrait être améliorée sur les fonctionnalités de zoom d'édition etc. Peut-être reprendre ce qui se fait sur open music etc. Il y a des fenêtres de segmentation sur lesquelles tu pourrais reprendre des informations. Il y a des outils un peu standards. Ça serait bien d'avoir des raccourcis. Tu peux avoir envie d'aller vite et avec Mono-Replay je pense qu'il y a des récurrences. Ça permettrait à des utilisateurs de



savoir ce qu'ils ont déjà. En l'état, par exemple, tu ne peux pas sélectionner ton marqueur et juste le bouger un peu, pour le supprimer tu dois te placer un peu après... ce n'est pas très intuitif. Ce sont des petits trucs comme ça qui simplifient la vie et qui font que voilà tu vas avoir plus envie de l'utiliser. À mon avis c'est hyper important. Après, sur le reste, il y a des petites fonctions sur les essais que j'ai fait sur la manipulation et il y a des fonctions que j'aimerais avoir.

*Question 10* : Est-ce que tu penses qu'il y a des éléments dans l'interface qui pourraient servir l'expressivité musicale ?

*Basile* : Je ne sais pas si je vais répondre vraiment à la question mais je pense que oui, il y a un peu d'expressivité... jusqu'à un certain point. En fait, effectivement, sur la gestion du tempo et puis la façon dont le scrub fonctionne, il y a de l'expressivité. Il y a des trucs un peu artificiels comme le freeze, où tu perds tout de suite le lien avec l'expressivité. L'expressivité c'est le lien un peu au geste et un geste freeze parfait ça n'existe pas. Là d'une certaine manière on perd le lien au physique. Par exemple, sur le son de piano du morceau de Satie, le fait qu'on choisisse l'option freeze nous fait perdre en expressivité parce que pendant le moment où tu vas choisir de faire durer ton son sur une certaine durée-, à ce moment-là le son ne sera pas naturel. Et là c'est une des fonctions qu'on essaie de rejouer dans Mono-Replay. Sur le piano tu ne peux pas le faire durer éternellement. Au-delà de 10 secondes ça s'estompe naturellement. Sur l'interprétation de Satie on pourrait encore aller plus loin. On pourrait jouer un peu sur le freeze. Là le freeze que j'ai entendu est très digital. C'est radical. C'est toujours le même freeze. Imaginons que cette pièce était enregistrée par un vent. Tu peux faire durer le son, tu peux l'entretenir. Quand tu fais durer ton son il n'est jamais très droit.

Sur l'utilisation c'est sûr qu'il y a de l'expressivité, et ça c'est ce que je te disais au début mais pour un usage pédagogique. L'idée c'était de rejouer des enregistrements de piano un peu comme le Métapiano sur des enregistrements. Là je pense que ça marche, si l'objectif c'est bien de rejouer des pièces du répertoire qui sont pulsées etc.

Après pour l'usage du point de vue de la création c'est un peu plus compliqué j'ai l'impression.

*Question 11* : As-tu utilisé une interface gestuelle ?

*Basile* : Je n'ai pas utilisé d'interface mais à terme c'était l'objectif.

*Question 12* : Quel geste l'enregistrement que tu souhaitais manipuler tu as évoqué ? Y'avait-il un geste ?

*B* : Oui, car l'objet de ma création c'est de faire un lien entre le son et le geste. En fait j'avais l'idée de faire quelque chose avec les danseurs. En gros leur rythme corporel peut faire varier la vitesse de lecture d'un enregistrement. C'est quelque chose d'un peu chaotique et en fonction des ralentis et des accélérations, ça permet de régler ces paramètres. C'est ça que je souhaitais un peu faire. Je voulais composer des choses qui donnent une impression de régularité avec des accélérations. Avec toujours une idée de débit qui accélère et qui ralentit. Je suis allé trop loin dans la composition. J'ai voulu vraiment composer précisément ces accélérations et ces ralentis. Je me disais « voilà ça m'intéresse qu'à un moment, si les danseurs s'arrêtent, le son s'arrête ». Si le danseur choisit de faire un geste en boucle, que ce soit la même chose pour le danseur. L'aspect gestuel était hyper important. Les danseurs ont des petites bagues/capteurs.

*Interviewer* : Ces gestes-là, arrives-tu à les utiliser ?

*Basile* : Sur l'aspect fin c'est compliqué. Je voulais avoir une moyenne de la vitesse de déplacement du danseur. Quand tu as des gestes très rapides, je voulais pouvoir les détecter comme des gestes

rapides. L'avantage c'est que sur une des deux bagues que j'utilise il y a des boutons. Ça te permet de tricher. Le danseur a conscience qu'il peut faire un geste rapide et associer un clic pour que ça puisse déclencher un son ou un transitoire dans la musique. Au contraire je vais faire un geste lent. Je vais maintenir mon clic et quand je vais le lâcher je vais maîtriser autre chose. Il y a un lien au geste c'est sûr. Je m'étais fait une idée des liens entre les gestes et le son que je voulais travailler et les transposer à d'autres gestes. Je fais un son rapide à la souris. J'ai essayé de reproduire ces gestes-là.

Sur le « Scrub » par exemple c'était clairement le moment où tu as un geste un peu plus lent, où là, du coup tu vas en fonction de ce que tu captes et ça déplace ton curseur dans le fichier son. Il y a une captation un peu plus précise. Alors que quand il y a des mouvements un peu plus hachés, rapides, là, ça pourrait jouer sur le « tap tempo » ou sur « speed ». Comme ça, plus tu es rapide et plus ton fichier est lu rapidement et plus tu es lent, plus c'est lu lentement. Il y a des correspondances qui peuvent marcher je pense mais ça demande vraiment du temps pour que ça marche et ça demande aussi de tricher. Mais ça, c'est plutôt lié au capteur... peut-être aux deux aussi.

*Question 13* : Est-ce que tu penses qu'il y a des perspectives qu'il faudrait souligner par rapport à l'expressivité ?

*Basile* : Sur l'expressivité à mon avis ça marche assez bien. Après sur le time-stretch, quand tu étires, ça me gêne. Le fait de pouvoir aussi choisir entre un freeze standard et une sorte de résonance naturelle. Ce sont les seules choses que je vois qui sont améliorables.

*Interviewer* : J'ai l'impression que ces deux choses sont déjà paramétrables dans Mono-Replay...

*Basile* : Dans la fenêtre freeze ça serait bien de régler les paramètres de celui-ci. Est-ce que c'est un freeze droit ? Est-ce que c'est un freeze naturel avec un fade out ? Pouvoir choisir entre ces deux-là, ça serait déjà très bien. Ça rajouterait une couche d'expressivité.

Pour que ce soit vraiment expressif il faut pouvoir changer constamment d'un geste à l'autre la configuration.

Dans ce logiciel j'ai vu deux grandes catégories. Il y a le côté expressivité sur la lecture et le côté vocodeur et puis enveloppe où là, pour le coup, je n'ai pas vu le lien entre les deux. En fait si tu veux, si on imagine que tu fais varier la vitesse de lecture de ton fichier, après si tu rajoutes une enveloppe telle que moi je l'ai vu, ça te rajoute une enveloppe un peu aléatoire sur laquelle tu n'as pas de contrôle... tu vois ... hormis on/off. Ça me semblait très éloigné. Un des usages qui me semblerait hyper intéressant c'est vraiment ce qui a déjà été fait sur le méta piano. C'est qu'il y ait une sorte de librairie de choses déjà indexées pour des pratiques pédagogiques avec des pièces hyper bien indexées, que les gens puissent rejouer.

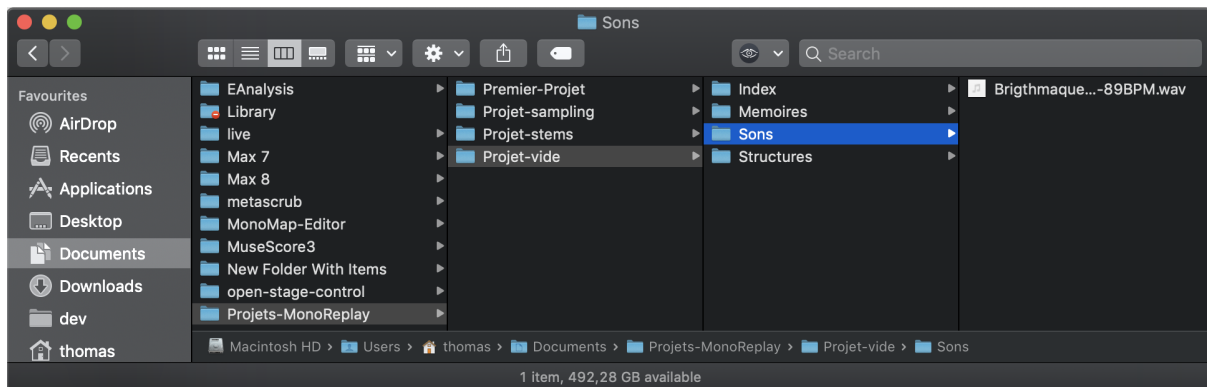
## Annexe G

# Instruction de l'atelier SMAC pour créer un projet à l'aide de Mono-Replay

## Atelier SMAC#4 : Création de projet avec Mono-Replay :

### I) Préparation du Projet-Vide :

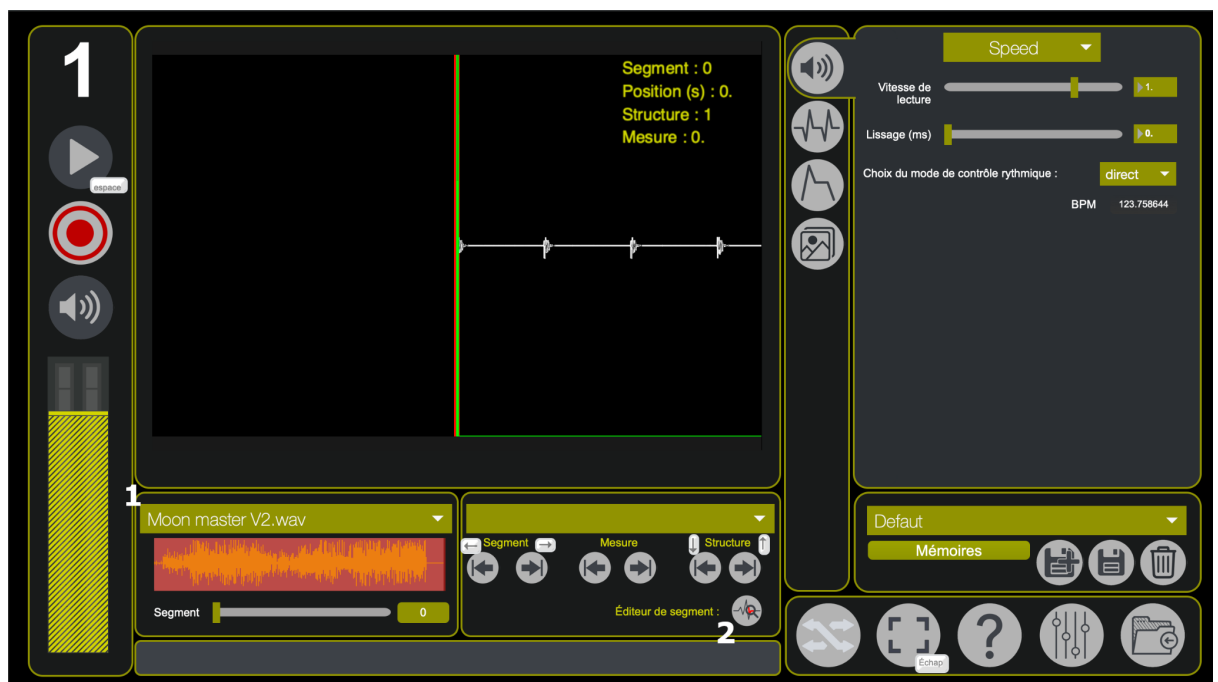
- 1) Glissez le dossier “Projet-vidé” à la racine de votre répertoire Documents (utilisateur mac) comme indiqué dans la figure ci-dessous :



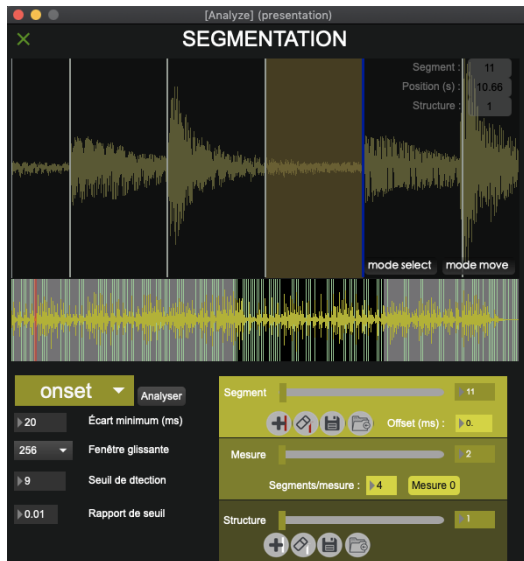
- 2) Placez votre fichier .wav dans le répertoire /Sons à l'intérieur du dossier “Projet-vidé”.  
**Veillez à ce que la fréquence d'échantillonnage du fichier soit 44100 Hz et que la résolution soit de 16 bits. Il ne faut pas de “\_” dans le nom de votre fichier audio.**

### II) Présentation de l'éditeur de segment

- 1) Vérifiez que le fichier audio que vous souhaitez étiqueter est bien sélectionné dans le menu son (1) puis cliquez sur le bouton “Éditeur de segment” (2)

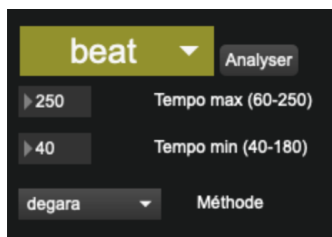


2) La fenêtre ci-dessous s'ouvre :

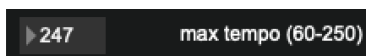


### III) Etiquetage automatique :

1) Étiquetage par détection de pulse :



← Ici on peut lancer une segmentation automatique dites “à la pulse” à l’aide de deux méthodes de détection de pulse différentes (informations dans le paragraphe **méthode**). Le bouton **Analyser** lance la détection.



**Tempo max** → Ici on peut paramétrer la valeur maximale de tempo à détecter (valeur en BPM).



**Tempo min** → Ici on peut paramétrer la valeur minimale de tempo à détecter (valeur en BPM).



**Méthode** → Ici on peut changer de méthode d’analyse.

- La méthode **Degara** : Cette méthode utilise la fonction de détection de transitoire dites par “différence spectrale complexe” puis l’algorithme de suivi de pulse TempoTapDegara pour déterminer les pulsations.

[https://essentia.upf.edu/reference/std\\_BeatTrackerDegara.html](https://essentia.upf.edu/reference/std_BeatTrackerDegara.html)

- La méthode **multifeature** : Cette méthode calcule 8 fonctions de détection de transitoire différentes et estime pour chacune d'entre elles les positions de chaque pulse à l'aide de l'algorithme de suivi de pulse TempoTapDegara.

[https://essentia.upf.edu/reference/std\\_BeatTrackerMultiFeature.html](https://essentia.upf.edu/reference/std_BeatTrackerMultiFeature.html)

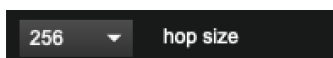
## 2) Étiquetage par détection de pulse :



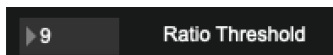
← Ici on peut lancer la segmentation automatique d'un fichier audio selon la position des transitoires d'attaque déterminée par l'algorithme SuperFluxExtractor.  
([https://essentia.upf.edu/reference/std\\_SuperFluxExtractor.html](https://essentia.upf.edu/reference/std_SuperFluxExtractor.html))



**Écart minimum (ms)** → Ici on peut paramétrer le plus petit écart temporel entre deux transitoires.



**Fenêtre glissante** → Ici on peut paramétrer la taille de fenêtre glissante de l'analyse.



**Seuil de détection** → Ici on peut paramétrer le seuil de détection sur la rapport suivant : (valeur de la courbe de nouveauté) / (moyenne de la courbe de nouveauté), utilisez 0 pour le désactiver (convient au transitoire à faible énergie)







**Rapport de seuil** → Ici on peut paramétrer le seuil de ratio pour la détection de pic en se basant sur la différence entre la courbe de nouveauté et la valeur moyenne du signal (convient pour les transitoires noyés dans le bruit ambiant)

### III) Étiquetage manuel :

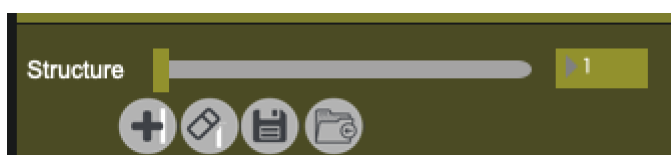
Il est possible de faire un étiquetage de façon entièrement manuel ou de corriger un étiquetage en cours à l'aide de ce menu :

#### 1) Étiquetage dit "au Segment" (niveau 1) :






- ajouter un index à la position sélectionnée  :
- supprimer l'index courant  :
- enregistrer la collection d'index courante dans un fichier texte : 
- ouvrir un fichier texte contenant une collection d'index : 
- **Offset (ms)** : permet d'ajouter ou de soustraire un décalage temporel à tous les index.

#### 2) Étiquetage dit "à la Structure" (niveau 3) :



Il est possible de faire la même chose avec les structures :

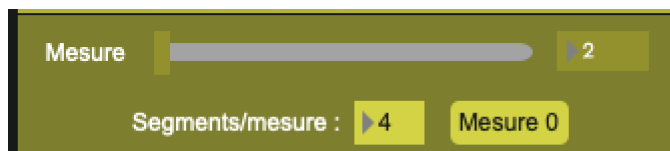
- ajouter une structure à la position sélectionnée : 

- supprimer la structure courante : 
- enregistrer les structures actuelles dans un fichier texte : 

Laisser le nom “structure” proposé lors de l’enregistrement de votre fichier. Ce sera la structure enregistrée avec ce nom qui sera ouverte lorsque vous sélectionnerez le fichier audio.

- ouvrir un fichier texte contenant un ensemble de structure : 

### 3) Étiquetage dit “à la Mesure” (niveau 3) :



**Segments/mesure** : permet de définir le nombre de segments par mesure

**Mesure 0** : permet de définir la position de la première mesure sur la mesure actuellement sélectionnée



Annexe H

## Dictionnaire OSC des variables de Mono-Replay

# Dictionnaire paramètres OSC

## Mono-Replay

Onglet	Nom de variable	Valeurs	Nom commun	Path OSC
Family_1 : Contrôle rythmique	MonoReplay_scrubposition	0_1. 0.	Position de lecture	/MonoReplay/Mode/scrubp osition
Family_1 : Contrôle rythmique	MonoReplay_lissage	0_1000. 0.	Temps pour changer de valeur de vitesse	/MonoReplay/Mode/lissage
Family_1 : Contrôle rythmique	MonoReplay_moyenneglissante	0_20 0.	Nombre d'itération pour effectuer la moyenne glissante	/MonoReplay/Mode/moyen neglissante
Family_1 : Contrôle rythmique	MonoReplay_speedvariation	-2_2 1.	Vitesse de lecture	/MonoReplay/Mode/speedv ariation
Family_1 : Contrôle rythmique	MonoReplay_Button_tapmode	0_9. 0.	Changement de mode de contrôle rythmique	/MonoReplay/Mode/MonoR eplay_Button_tapmode
Family_1 : Contrôle rythmique	MonoReplay_Button_TapTempo Bang	0_1 0.	Tape un tempo	/MonoReplay/Mode/Button_ TapTempoBang
Family_1 : Contrôle rythmique	MonoReplay-position-loop	0_1 0.5	Varie la position de la boucle	/MonoReplay/Mode/position -loop

Onglet	Nom de variable	Valeurs	Nom commun	Path OSC
Family_2 : Vocodeur	MonoReplay_spectralenvelopetrans	-1000_1000. 0.	Transposition de l'enveloppe spectrale	/MonoReplay/Vocodeur/spectralenvelopetrans
Family_2 : Vocodeur	MonoReplay_spectralenvelopescaling	0_2 1..	Modification du timbre	/MonoReplay/Vocodeur/spectralenvelopescaling
Family_2 : Vocodeur	MonoReplay_pitchtransposition	-1000_1000 0.	Transposition du pitch	/MonoReplay/Vocodeur/pitchtransposition
Family_2 : Vocodeur	MonoReplay_sinusoide	0_1 1.	Mixage des éléments harmoniques	/MonoReplay/Vocodeur/sinusoide
Family_2 : Vocodeur	MonoReplay_noise	0_1 1.	Mixage des éléments inharmoniques	/MonoReplay/Vocodeur/noise
Family_2 : Vocodeur	MonoReplay_transitoire	0_1 1.	Mixage des transitoires	/MonoReplay/Vocodeur/transitoire
Family_2 : Vocodeur	Umenu_WinowSize	0_17 7.	Taille de la fenêtre d'analyse	/MonoReplay/Vocodeur/Umenu_WinowSize
Family_2 : Vocodeur	Umenu_Overlap	0_6 4.	Facteur de recouvrement	/MonoReplay/Vocodeur/Umenu_Overlap
Family_2 : Vocodeur	Umenu_FreqFactor	0_5 0.	Facteur de recouvrement bis	/MonoReplay/Vocodeur/Umenu_FreqFactor

Onglet	Nom de variable	Valeurs	Nom commun	Path OSC
Family_3 : Enveloppe ADSR	MonoReplay_AdsrOnOff	0_1. 0.	activation de l'enveloppe	/MonoReplay/Enveloppe/Ad srOnOff
Family_3 : Enveloppe ADSR	MonoReplay_adsrattack	0_1000 150.	temps d'attaque	/MonoReplay/Enveloppe/ad srattack
Family_3 : Enveloppe ADSR	MonoReplay_adsrdecay	0_1000 300.	Temps de chute	/MonoReplay/Enveloppe/ad srdecay
Family_3 : Enveloppe ADSR	MonoReplay_adsrrelease	0_1000 400.	Temps de relâche	/MonoReplay/Enveloppe/ad srrelease
Family_3 : Enveloppe ADSR	MonoReplay_adsrsustain	0_2 0.5	Temps de tenue	/MonoReplay/Enveloppe/ad srsustain

Onglet	Nom de variable	Valeurs	Nom commun	Path OSC
Family_4 : Rendu	MonoReplay_ScaleX	0_10. 0.	Échelle des abscisses	/MonoReplay/Rendu/Scale X
Family_4 : Rendu	MonoReplay_ScaleY	0_10 0.	Échelle des ordonnées	/MonoReplay/Rendu/Scale Y
Family_4 : Rendu	MonoReplay_LineWidth	0_10 0.	Épaisseur du trait	/MonoReplay/Rendu/LineWi dth

Onglet	Nom de variable	Valeurs	Nom commun	Path OSC
Family_5 : Enveloppe 1	MonoReplay_button-hauteur1	0_1. 0.	enveloppe sur la valeur de pitch	/MonoReplay/Enveloppe1/b utton-hauteur1
Family_5 : Enveloppe 1	MonoReplay_button-envspectral1	0_1. 0.	enveloppe sur la transposition spectrale	/MonoReplay/Enveloppe1/b utton-envspectral1
Family_5 : Enveloppe 1	MonoReplay_button-envechelle1	0_1. 0.	enveloppe sur la variation de timbre	/MonoReplay/Enveloppe1/b utton-envechelle1
Family_5 : Enveloppe 1	MonoReplay_button-sin1	0_1. 0.	enveloppe sur le mix des harmoniques	/MonoReplay/Enveloppe1/b utton-sin1
Family_5 : Enveloppe 1	MonoReplay_button-transitoire1	0_1. 0.	enveloppe sur le mix des transitoires	/MonoReplay/Enveloppe1/b utton-transitoire1
Family_5 : Enveloppe 1	MonoReplay_button-noise1	0_1. 0.	enveloppe sur le mix du bruit	/MonoReplay/Enveloppe1/b utton-noise
Family_5 : Enveloppe 1	MonoReplay_env1_amplitude	0_1. 0.5	Règle l'amplitude globale de l'enveloppe 1	/MonoReplay/Enveloppe1/e nv1-amplitude
Family_5 : Enveloppe 1	MonoReplay_env1_frequence	0_100. 50.	Règle le rang des fréquences	/MonoReplay/Enveloppe1/e nv1_frequence

			de l'enveloppe	
--	--	--	----------------	--

Onglet	Nom de variable	Valeurs	Nom commun	Path OSC
Family_6 : Enveloppe 2	MonoReplay_button-hauteur2	0_1. 0.	enveloppe sur la valeur de pitch	/MonoReplay/Enveloppe2/button-hauteur2
Family_6 : Enveloppe 2	MonoReplay_button-envspectral2	0_1. 0.	enveloppe sur la transposition spectrale	/MonoReplay/Enveloppe2/button-envspectral2
Family_6 : Enveloppe 2	MonoReplay_button-envehelle2	0_1. 0.	enveloppe sur la variation de timbre	/MonoReplay/Enveloppe2/button-envehelle2
Family_6 : Enveloppe 2	MonoReplay_button-sin2	0_1. 0.	enveloppe sur le mix des harmoniques	/MonoReplay/Enveloppe2/button-sin2
Family_6 : Enveloppe 2	MonoReplay_button-transitoire2	0_1. 0.	enveloppe sur le mix des transitoires	/MonoReplay/Enveloppe2/button-transitoire2
Family_6 : Enveloppe 2	MonoReplay_button-noise2	0_1. 0.	enveloppe sur le mix du bruit	/MonoReplay/Enveloppe2/button-noise2
Family_6 : Enveloppe 2	MonoReplay_env2_amplitude	0_1. 0.5	Règle l'amplitude globale de l'enveloppe 2	/MonoReplay/Enveloppe2/env2-amplitude
Family_6 : Enveloppe 2	MonoReplay_env2_frequence	0_100. 50.	Règle le rang des fréquences de l'enveloppe	/MonoReplay/Enveloppe2/env2_frequence

Onglet	Nom de variable	Valeurs	Nom commun	Path OSC
Family_7 : Index	MonoReplay_in dexnavigation	0_nb d'index. 0	Changer d'index	/MonoReplay/Index/index navigation

Onglet	Nom de variable	Valeurs	Nom commun	Path OSC
GUI	MonoReplay_ButtonIndexUp	0_1 0.	Passer à l'index supérieur	/MonoReplay/GUI/ButonIn dexUp
GUI	MonoReplay_ButtonIndexDown	0_1 0.	Passer à l'index inférieur	/MonoReplay/GUI/ButonIn dexDown
GUI	MonoReplay_ButtonMesureUp	0_1 0.	Passer à la mesure suivante	/MonoReplay/GUI/ButonM esureUp
GUI	MonoReplay_ButtonMesureDow n	0_1 0.	Passer à la mesure précédente	/MonoReplay/GUI/ButonM esureDown
GUI	MonoReplay_ButtonStructDown	0_1 0.	Passer à la structure Inférieur	/MonoReplay/GUI/ButonStr uctDown
GUI	MonoReplay_PlayPause	0_1 0.	Lancer/arrêter la lecture	/MonoReplay/GUI/ButonPI ayPause
GUI	MonoReplay_MasterVolume	0_1 0.	Monter/baisser le volume	/MonoReplay/GUI/Volume
GUI	MonoReplay_Son	0_1 0.	Couper le son	/MonoReplay/GUI/ButonH P

Onglet	Nom de variable	Valeurs	Nom commun	Path OSC
Memoire	MonoReplay_Memory_Menu _index	numéro de ligne	Mémoires	/MonoReplay/Memoir e/Memoire
Memoire	MonoReplay_Memory_Menu	nom de ligne	Mémoires	/MonoReplay/Memoir e/Memoire/name
Memoire	MonoReplay_Interpolation_M enu_index	numéro de ligne	Interpolations	/MonoReplay/Memoir e/Interpolation

Memoire	MonoReplay_Interpolation_Menu	nom de l'inerpolation	Interpolations	/MonoReplay/Memoire/Interpolation/name
---------	-------------------------------	-----------------------	----------------	--



## Annexe I

# Questionnaire d'évaluation portant sur l'ergonomie des logiciels Mono

# Projet Poly-Map. Questionnaire élève : Retour d'expérience après un projet collectif en classe

[Connectez-vous à Google](#) pour enregistrer votre progression. [En savoir plus](#)

## INFORMATIONS GÉNÉRALES PARTICIPANT

Veuillez remplir s'il vous plaît les informations suivantes:

Nom du cours (si cours en établissement éducatif):

Votre réponse

Nom du projet pédagogique ou artistique:

Votre réponse

