HAL
open science

Le frequency shifter : enquete dans un repertoire d'oeuvres mixtes produites a l'IRCAM

Olivier Baudry, Alain Bonardi

▶ **To cite this version:**

Olivier Baudry, Alain Bonardi. Le frequency shifter : enquete dans un repertoire d'oeuvres mixtes produites a l'IRCAM. Journées d'Informatique Musicale, May 2015, Montréal, Canada. hal-03104586

HAL Id: hal-03104586

https://hal.science/hal-03104586v1

Submitted on 9 Jan 2021

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.

LE FREQUENCY SHIFTER : ENQUETE DANS UN REPERTOIRE D'ŒUVRES MIXTES PRODUITES A L'IRCAM

Olivier Baudry
CICM-E.A.1572-Université Paris 8
olivierbaudry.eba@hotmail.fr

Alain Bonardi
CICM-E.A.1572-Université Paris 8 et Ircam /
Equipe Analyse des Pratiques Musicales
alain.bonardi@gmail.com

RÉSUMÉ

Dans cet article, nous rendons compte d'une recherche autour du frequency-shifter, module employé dans de nombreuses œuvres mixtes à l'Ircam. Dans ce cadre, nous avons constitué un corpus constitué de huit œuvres mixtes et leurs différentes versions allant de 1981 à 2008. Nous nous proposons d'étudier les implémentations informatiques du frequency-shifter en suivant à la fois les œuvres, les compositeurs et les réalisateurs en informatique musicale. Nous situons l'usage musical des frequency-shifters à travers les œuvres de notre corpus. Nous abordons une classification technique des frequency-shifters à travers les éléments qui le composent, et nous procédons à des comparaisons sonores des implémentations.

1. INTRODUCTION

L'analyse des processus de création musicale ne cesse d'élargir son domaine d'action vers les musiques faisant appel aux moyens électroniques et informatiques. Les contributions récentes sont nombreuses, donnant lieu à un certain nombre de projets de recherche :

- DiP-CoRE [1-2] articule dissémination et préservation des œuvres de musique mixte au travers de la documentation des processus de production.

- GAMELAN vise une modélisation en ingénierie des connaissances des activités de production musicale en environnement numérique [9].

- TACEM se consacre aux liens entre technologies et composition électroacoustique, et propose le logiciel TIAALS pour l'analyse aurale interactive [3].

Certains chercheurs ont étudié plus particulièrement le moment de la reprise des œuvres avec technologie : l'obsolescence des moyens techniques engendre de nouveaux processus de création technico-musicaux, dont l'étude est pertinente pour eux-mêmes mais aussi pour comprendre les processus initiaux [6]. C'est dans ce cadre que nous positionnons nos travaux, consacrés à la transmission des pièces avec électronique et centrés sur le moment-clé de la reprise d'une œuvre.

En parallèle d'une enquête de terrain auprès de réalisateurs en informatique musicale sous la forme d'un questionnaire sollicitant leurs expériences pratiques de transmission, nous commençons une étude centrée sur un dispositif technique précis, classique en musique électronique : le frequency-shifter[1].

Nous étudions ce dispositif car utilisé depuis longtemps dans les musiques électroacoustiques, sa définition n'est *a priori* plus à faire.

Il en existe une définition communément admise : étant donné une fréquence f_0, le frequency-shifter opère simultanément deux transformations sur un son en entrée, la première en ajoutant f_0 aux fréquences présentes dans le son d'origine, la seconde en soustrayant f_0 de ce même spectre. Ce décalage donne des résultats sonores intéressants à deux échelles, microscopique et macroscopique : d'une part, des battements; d'autre part, la transformation d'un son harmonique en un son inharmonique. Le frequency-shifter diffère du modulateur en anneau par le fait qu'il sépare les deux sons résultants : la sortie dite positive propose le son d'origine décalé par ajout de f_0, la sortie dite négative celui obtenu par soustraction de f_0.

C'est essentiellement dans cette séparation des deux sons résultants que nous trouvons les différences d'implémentation : dans le paragraphe 2.3, nous montrons la diversité de mise en œuvre technique du filtre de Hilbert, élément central du dispositif. La diversité règne également au niveau des dénominations et de la prise en compte du dispositif dans les œuvres. Les implémentations techniques et les résultats sonores produits ne sont pas répertoriés systématiquement. Les choix compositionnels sont rarement exprimés clairement.

Nous faisons l'hypothèse de la pertinence d'une étude historique et technique du frequency-shifter comme contribution à l'analyse des processus de création des

[1] Tout au long de cet article, nous employons le terme frequency-shifter en anglais, plutôt que sa traduction française « décaleur de fréquence », qui nous semble moins utilisée même dans la communauté francophone d'informatique musicale.

L'autorisation de faire des copies papier ou numériques, de tout ou de partie de ce travail, pour un usage personnel ou pédagogique, est accordé sans frais à condition que les copies ne soient pas faites ou distribuées dans un but lucratif ou commercial, et que les copies portent ce texte et la citation complète de l'article sur la première page .
© 2015 Journées d'Informatique Musicale, Montréal

œuvres mixtes. Il s'agit de comprendre les évolutions du frequency-shifter comme dispositif technique utilisé par de nombreux créateurs, et d'en envisager les conséquences musicales.

2. METHODOLOGIE DE L'ENQUETE

2.1. Choix d'un corpus d'œuvres

Nous avons entrepris un relevé non exhaustif de l'emploi du frequency-shifter dans des œuvres mixtes créées à l'Ircam de 1981 à nos jours par trois compositeurs : Pierre Boulez, Philippe Manoury, et Philippe Leroux. Nous avons suivi les versions successives de huit œuvres. Pour Boulez, *Répons* (deux versions : 1981, 2006), *Explosante-fixe* (1993), *Anthèmes 2* (deux versions : 1999 et 2005); pour Manoury, *Jupiter* (trois versions : 1987, 1992, 2004), *La Partition du ciel et de l'enfer* (deux versions : 1989, 1996), *Neptune* (quatre versions : 1991, 1993, 1998 et 2008), *Partita 1* (version de 2007) et enfin, *Voi(rex)* de Leroux dans sa version de 2002.

Nous nous sommes appuyé sur la base de données Sidney[2] interne à l'Ircam, qui présente pour chaque œuvre désignée une documentation, permettant de rejouer l'œuvre et qui fournit les patchs à l'intérieur d'archives si celles-ci ont été constituées.

Année / *Œuvre*	1981	1987	1989	1991	1992	1993	1996	1997	1998	2002	2004	2005	2006	2007	2008
Répons, Boulez	1												4		
Explosante-fixe, Boulez						3									
Anthèmes 2, Boulez								3				4			
Jupiter, Manoury		2			3						5				
La Partition du ciel et de l'enfer, Manoury			2				3								
Neptune, Manoury				2		3			4						5
Partita 1, Manoury														4	
Voi(rex), Leroux										4					

Figure 1. Corpus des huit œuvres mixtes étudiées.

Les différentes versions de ces huit œuvres furent implémentées sur cinq configurations matérielle et logicielle différentes, que nous repérons de 1 à 5 sur le tableau de la figure 1 :

- la configuration 1 est la machine 4X avant l'apparition du logiciel Max,
- 2 : la 4X avec Max,
- 3 : la station Next avec Max/FTS [8],
- 4 : Max/MSP sur micro-ordinateur,
- 5 : PureData sur micro-ordinateur.

2.2. Utilisations du frequency-shifter dans les œuvres

Les utilisations du frequency-shifter sont assez différentes d'une œuvre à l'autre de notre corpus, selon deux axes :

- timbral et harmonique : transformation monophonique ou polyphonique du son de l'instrument.

- rythmique : couplage ou pas avec une ligne à retard réinjectée.

2.2.1. Utilisation à caractère timbral et harmonique

Nous remarquons ici l'usage du frequency-shifter de manière monophonique avec dans *Jupiter*, *Voi(rex)*, *Partita 1* et *Explosante-fixe*, monophonique ou polyphonique dans *Anthèmes 2*, polyphonique dans *Neptune*, *Répons* et *La Partition du ciel et de l'enfer*.

Dans *Jupiter*, le frequency-shifter est souvent couplé aux harmonizers et à la réverbération. Le son de la flûte est traité dans l'idée d'avoir différentes proliférations des harmoniques dans le temps. Comme le souligne Philippe Manoury[3] dans la notice de l'œuvre : « *l'idée étant que, partant d'un triple son de flûte, je l'extrapole jusqu'à le rendre méconnaissable, tissant ainsi des liens entre sons connus et inconnus dans une dialectique compositionnelle* ». L'électronique entretient ainsi un dialogue avec le flûtiste.

Dans la dernière pièce du cycle Sonus ex Machina, *Neptune*, pour deux vibraphones Midi, tam-tam, marimba et système électronique en temps réel, le module de frequency-shifting utilisé par Philippe Manoury (dont la conception est issue de l'héritage de *La Partition du ciel et de l'enfer*) est composé de huit sous-modules utilisés de manière séquentielle permettant de traiter le son du vibraphone et ainsi articuler une trame spectrale chargée.

Le frequency-shifter dans *Répons* (1981) de Pierre Boulez traite de manière indépendante l'ensemble des instruments pianos, xylophone, cymbalum, harpe, vibraphone, glockenspiel par différents sous modules au nombre de huit. La encore, il y a une idée de prolifération des harmoniques dans le temps par l'ajout de délais, afin de créer un jeu de questions et réponses avec les instrumentistes.

[2] http://brahms.ircam.fr/sidney/

[3] http://brahms.ircam.fr/works/work/10482/

2.2.2. *Utilisation à caractère rythmique*

Nous pouvons également considérer les frequency-shifters dans une utilisation à caractère rythmique avec respectivement *Anthèmes 2*, *Répons* et *Explosante-fixe* de Pierre Boulez.

Dans *Anthèmes 2,* le module de frequency-shifting est composé de quinze frequency-shifters en série avec l'adjonction de délais (ligne à retard avec réinjection). Chacun des modules permet de multiplier le nombre de lignes musicales, à la manière d'un canon, permettant une prolifération des harmoniques dans le temps et dans l'espace.

Dans le cas de *Répons*, le fait d'avoir un module de frequency-shifting attribué à chacun des instruments permet de créer une rythmique par l'ajout de délais (ligne à retard) de durées différentes dans chacun des sous-modules. De même dans *Explosante-fixe* le module de frequency-shifting comporte deux objets freqshift~ associés à des délais (ligne à retard) pouvant ainsi créer des rythmiques.

2.3. Classification technique des frequency-shifters

Dans chacune de ces œuvres, nous avons observé l'implémentation technique du frequency-shifter. Du point de vue du procédé, nous n'avons trouvé dans notre corpus que des implémentations dites *time-domain* (traitement temporel) et aucune dite *frequency-domain* (traitement fréquentiel). Le frequency-shifter *time-domain* se compose de deux étages : un filtre de Hilbert[4] qui crée un signal complexe en quadrature de phase à partir du son en entrée; puis un module de décalage par multiplication complexe de la fréquence f_0. Ce dernier ne donne pas lieu à des variations d'implémentation. En revanche, le filtre de Hilbert est implémenté de diverses manières, comme le montre la figure 2. Les implémentations correspondent à des approximations de la transformée de Hilbert[5], obtenues soit par des filtres passe-tout en cascade soit par une convolution limitée.

Pour établir ce tableau et sa logique généalogique, nous avons dialogué avec les auteurs des développements informatiques, notamment Miller Puckette, Richard Dudas et Zack Settel[6]. Par exemple, l'abstraction[7] hilbert~ développée par Miller Puckette fait suite à une étape de *reverse engineering*, du code développé par Emmanuel Favreau[8] utilisé sur le processeur d'effet 4X à l'IRCAM en 1982; nous avons sollicité Miller Puckette[9] pour obtenir des détails sur le processus d'implémentation : « *je pense que je l'ai fait en 1990, mais peut-être 1991, dans le cadre du 'portage' de* Pluton [œuvre de Philippe Manoury pour piano midi et électronique temps réel (1988-1989)] *de la 4X à ISPW*[10]*. Les travaux ont débuté en 1990, mais je pense qu'il n'était pas en fonctionnement entièrement jusqu'à 1991. Le premier fichier nommé hilbert~.pd que je peux trouver dans mes sauvegardes est de 2001. Mais je n'ai pas de sauvegardes pour 1999 ou 2000, il aurait pu apparaître un peu précédemment. Il est issu d'une importation directe de celui de ISPW (mais les coefficients ont légèrement changé parce que les arguments pour le filtre biquad Pd sont différents de ceux de la ISPW)* ».

Implémentation	Développeur	Configuration (cf. numéros dans le 2.1)	Années de développement	Remarques techniques
Implémentation freqshift / 4X	Emmanuel Favreau	1	1982	
Implémentation Next/Max/FTS	Miller Puckette	2	1990-1991	Reverse-engineering de l'implémentation d'Emmanuel Favreau sur 4X
Abstractions fshift~ et hilbert~	Miller Puckette	5a (PureData)	1999-2000	Portage sur PureData par Miller Puckette
Abstraction fshift1~ et hilbert~	Zack Settel	4a (collection d'abstractions Jimmies dans Max/MSP)	1993-1998	Résultat identique à l'abstraction fshift~ dans PureData mais avec seulement une sortie positive
Objet freqshift~ dans PureData	Martin Weiss	5b (PureData-extended)	2005-2006	Filtre de Hilbert fondé sur une convolution temporelle de profondeur 200
Objet freqshift~ dans Max/MSP	Cycling74 (code propriétaire)	4b (objet dans le logiciel Max/MSP)	information non disponible	Filtre de Hilbert du 6ème ordre

Figure 2. Relevé des implémentations techniques du frequency-shifter dans le corpus d'œuvres.

Du point de vue de la programmation, nous trouvons deux formes dans les environnements Max et PureData : les objets et les abstractions. Les premiers sont développés en langage C et complètent les traitements inclus par défaut dans le logiciel ; les secondes utilisent les objets

[4] Le filtre de Hilbert est un filtre qui opère à toute fréquence un retard de phase de ¶/2.

[5] La transformée de Hilbert est un filtre non-causal et donc pas réalisable telle quelle.

[6] Nous n'évoquons pas ici l'abstraction 2p2z~, filtre du type biquad~ mais avec d'autres jeux de coefficients, pour lequel nous avons obtenu des éclaircissements de Richard Dudas et Zack Settel.

[7] En Max ou en PureData, une abstraction est un module stocké sous la forme d'un patch indépendant dans un fichier externe, pouvant être appelée depuis un patch parent et incluse dans ce dernier.

[8] Emmanuel Favreau nous précise : « *l'implantation* [du filtre de Hilbert dans la 4X] *enchainait plusieurs allpass* [filtres passe-tout] *du premier ordre. Ce qui était la meilleure solution pour une arithmétique à virgule fixe.* » échange d'email le 20 novembre 2014.

[9] Echange d'email avec Miller Puckette en anglais le 11 novembre 2014.

[10] ISPW : Ircam Signal Processing Workstation développée entre 1981-1989 pour remplacer la 4X [5].

déjà présents dans Max ou PureData. Ainsi, les implémentations fshift~ et hilbert~ de Miller Puckette sur PureData, ainsi que les développements fshift1~ et hilbert~ de Zack Settel sur Max/MSP sont des abstractions. De plus, les environnements PureData et Max/MSP proposent tous deux des objets freqshift~ développés en C plus récemment.

Dans le cas des abstractions, l'étage consacré au filtre de hilbert~ est formé en utilisant des filtres numériques récursifs du second ordre appelés biquad, et implémentés sous la forme d'objets appelés biquad~. La théorie [4] de ces filtres indique qu'ils existent sous deux formes : la forme directe 2, adoptée par Miller Puckette dans PureData, et la forme directe 1 adoptée dans Max/MSP. Les figures 3 et 4 montrent les implémentations de ces abstractions dans les deux cas :

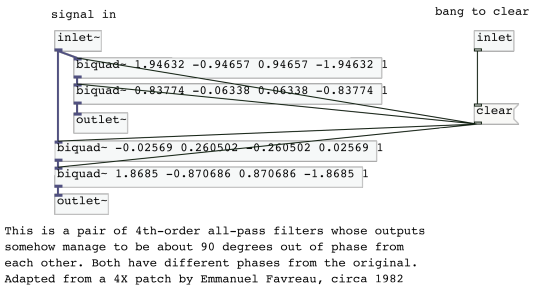

Figure 3. Implémentation du filtre de Hilbert dans l'abstraction PureData

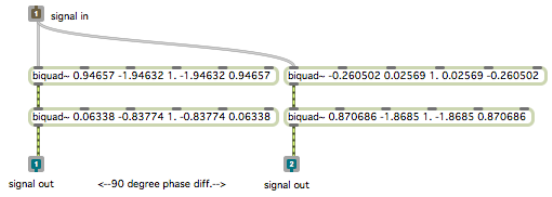

Figure 4. Implémentation du filtre de Hilbert dans Max/MSP (Jimmies)

Nous constatons d'emblée que les implémentations sont presque semblables mais que les coefficients de ces filtres sont différents. Or, en combinant les équations issues de la forme directe 2 (PureData), nous avons pu établir leur équivalence avec celles de la forme directe 1 (Max/MSP), à savoir que le biquad~(fb_1, fb_2, ff_1, ff_2, ff_3) de PureData (forme directe 2) est équivalent au biquad~(ff_1, ff_2, ff_3, $-fb_1$, $-fb_2$) de Max/MSP (forme directe 1). Ainsi, nous avons une équivalence rigoureuse entre les deux filtres de Hilbert de ces abstractions, si ce n'est l'ordre de leurs sorties qui est permuté.

Nous obtenons à l'issue de cette exploration une classification comportant trois implémentations principales désignées par A, B et C :

- implémentation A (Favreau/Puckette/Settel) : abstractions où le filtre de Hilbert est fondé sur une combinaison de 4 filtres passe-tout à 2 pôles et 2 zéros (chacun de ces 4 filtres est codé avec un biquad).

- implémentation B (Weiss) : objet PureData utilisant une convolution sur une profondeur temporelle de 200 valeurs[11] utilisant une forme tronquée de la réponse impulsionnelle du filtre de Hilbert.

- implémentation C (Cycling'74) : objet Max/MSP fondé sur un filtre passe-tout du 6ème ordre, dont le code (propriétaire) est très certainement inspiré de celui de Sean M. Costello dans l'environnement Csound.

2.4. Comparaison sonore des implémentations A, B, C

Dans un premier temps, nous avons comparé les implémentations A et C. Nous avons transcrit en langage FAUST [7] l'abstraction PureData de l'implémentation A. Ainsi, les deux implémentations de type A, issues de Max/MSP (nommée A1) et PureData (nommée A2), ainsi que l'implémentation C ont pu être associées et comparées dans le même patch Max/MSP, comme le montre la figure 5 :

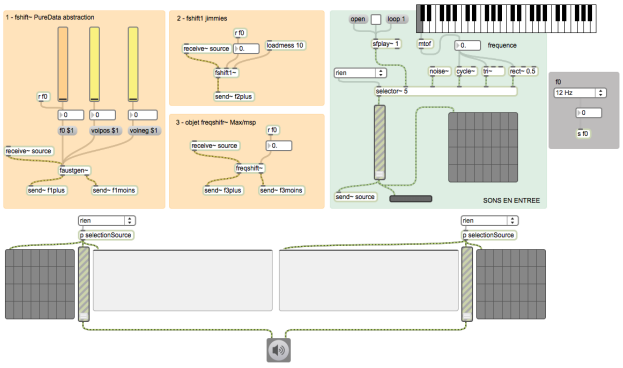

Figure 5. Patch Max/MSP comparant trois implémentations de frequency-shifter.

Nous avons entrepris des comparaisons sonores[12] des rendus obtenus avec les trois modules de frequency-shifting présents sur le patch de la figure 5 (A1, A2, C). Nous avons soumis l'ensemble des trois modules à plusieurs types de sons :

- un enregistrement de sons de piano provenant d'un piano à queue Yamaha C7, avec des accords attaqués *fortissimo*.

[11] Le code en langage C est disponible du fait que PureData est open source.

[12] Il s'agit de comparaisons au niveau signal, nous n'avons pas mené d'étude psychoacoustique sur la réception/comparaison de ces sons par des sujets humains.

- trois sons de flûte provenant de la banque Woodwinds par le biais du plugin logiciel Aria Player, dans l'idée d'avoir des notes tenues. Ces trois samples ont un profil dynamique à allure constante sans attaque. Les samples correspondent respectivement à un Do 4 de 0'47", à un Fa# 4 de 1'07", et à un La 4 de 1'14".

- des sons synthétiques : signaux triangulaire et carré (anti-aliasés).

Dans un premier temps, nous avons comparé les deux abstractions A1 (Max/MSP) et A2 (PureData). Les résultats sonores sont extrêmement proches sur les sons provenant du monde instrumental (enregistrement et sons d'une banque). On note une très petite différence sur les sons synthétiques. Par exemple, avec f_0 = 523 Hz, sur un signal triangulaire (objet tri~ de Max/MSP) accordé sur un si grave à 123,4708 Hz, l'implémentation A2 a une couleur très légèrement différente sur la sortie +, à peine un peu plus claire[13].

Si l'on compare maintenant les sorties positives implémentations A1 (abstraction Max/MSP) et C (objet Max/MSP), on obtient des différences plus sensibles, surtout dans le grave et médium grave comme le montre la figure 6 :

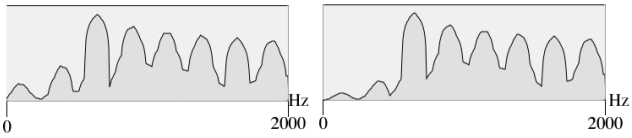

Figure 6. Comparaison sonore des implémentations A1 et C (sorties positives) sur un son triangulaire. A gauche, l'implémentation A1; à droite l'implémentation C. On remarque une différence dans le grave du spectre, dans la partie entre 0 et 400 Hz environ.

La comparaison entre les sorties négatives de A1 et C est moins flagrante, les résultats sont proches. L'implémentation A1 sonne très légèrement plus claire.

L'étape suivante consistera à coder dans le même langage descriptif, à savoir en FAUST, les trois implémentations A, B, C, permettant une comparaison sonore complète sur une même plateforme. Nous évaluerons également leur réponse en fréquence, aussi bien en amplitude qu'en phase.

[13] On ne peut pas comparer les sorties négatives du frequency-shifter entre A1 et A2 puisque l'abstraction A2 pour PureData ne possède pas de sortie négative.

3. CONCLUSION ET PERSPECTIVES

Nous avons constitué un corpus de huit œuvres mixtes composées à l'Ircam entre 1981 et 2008. Nous avons conduit une investigation dans les patchs temps réel des différentes versions de ces œuvres en pointant les différentes implémentations du frequency-shifter, dispositif que nous observons lors de notre enquête technique. Nous avons établi pour l'instant une première classification d'implémentations en trois groupes A, B et C.

Les différences de rendu sonore entre les implémentations A, B, C nous conduisent à poursuivre cette enquête en interrogeant les compositeurs et les réalisateurs en informatique musicale impliqués dans la conception technico-musicale des œuvres, pour comprendre la motivation du choix d'une implémentation lors de la création, et l'éventuel passage à une autre implémentation lors d'une migration. Par exemple, pourquoi l'implémentation A a-t-elle été conservée dans la deuxième version (2005) d'*Anthèmes 2* de Boulez, alors qu'il aurait été possible de passer à l'implémentation C ?

Nous continuerons notre exploration en nous adressant aux auteurs de ces diverses implémentations. De la même manière alors que des implémentations de frequency-shifting existaient, d'autres ont vu le jour, et pas seulement pour des raisons d'obsolescence : serait-ce à la demande des compositeurs ? Le choix de ces implémentations fshift~, fshift1~, freqshift~ ne relèverait-il pas du rôle du réalisateur en informatique musicale, reconnaissant une qualité sonore dans l'une plutôt que dans l'autre ? Nous poursuivrons nos travaux par l'implémentation en langage FAUST des différents frequency-shifters et procèderons à une comparaison des résultats sonores obtenus en effectuant des tests A/B entre les implémentations trouvées dans les versions de patchs des œuvres concernées et la « faustization » des modules équivalents générés.

4. REFERENCES

[1] Boutard, G., Marandola, F., « Dissémination et préservation des musiques mixtes : une relation mise en œuvre », *Actes des Journées d'Informatique Musicale 2014*, Bourges, France, p. 130-134, mai 2014.

[2] Boutard, G., Marandola, F., « Mixed music creative process documentation methodology : outcomes of the DIP-CoRE project », *Proceedings of the 9th Conference on Interdisciplinary Musicology – CIM 2014,* Berlin, Allemagne, 2014.

[3] Clarke, M., Dufeu, F., Manning, P., « Introducing TaCEM and the TIAALS software », *Proceedings of the 2013 International Computer Music Conference*, Perth, Australie, p. 47-53, août 2013.

[4] Kuo, S., Lee, B., Tian, W., *Real-time Digital Signal Processing Fundamentals, Implementations and Applications*, 3rd Edition, John Wiley & Sons, 2013.

[5] Lindemann, E., Dechelle, F., Smith, B., Starkier, M., « The Architecture of the IRCAM Musical Workstation », *Computer Music Journal,* The MIT Press, vol 15, n° 3, p. 41-49, automne 1991.

[6] Mihalic, A., Pottier, L., « Migrer des œuvres avec électronique temps-réel vers Faust », *Actes des Journées d'Informatique Musicale 2011*, Saint-Etienne, France, mai 2011.

[7] Orlarey, Y., Fober, D., Letz. S., « An Algebra for Block Diagram Languages », *Proceedings of the 2002 International Computer Music Conference*, Gothenburg, Sweden, p. 542-547, 2002.

[8] Puckette, M., « FTS : A Real-Time Monitor for Multiprocessor Music Synthesis », *Computer Music Journal*, The MIT Press, vol 15, n° 3, p. 58-67, automne 1991.

[9] Vincent, A., Bachimont, B., Bonardi, A., « Modéliser les processus de création de la musique avec dispositif numérique : représenter pour rejouer et préserver les œuvres contemporaines », *Actes des 23èmes Journées Francophones d'Ingénierie des Connaissances* (IC 2012), Paris, France. p. 83-98, juin 2012.