

Numéros / n° 9 - Inharmonique (1977) de Jean-Claude Risset : création, ré-création et avenir

« Analyse créative en recherche-crédation : trois exemples d'appropriation à partir du recodage d'*Inharmonique* de Jean-Claude Risset »

Alain Bonardi

Résumé

L'objectif de cet article est de montrer comment les auteurs se sont approprié le travail d'analyse-recodage d'*Inharmonique* de Jean-Claude Risset dans leurs propres créations. Cette version augmentée et révisée d'un précédent article publié dans les *Actes des Journées d'informatique musicale 2020* présente tout d'abord la modélisation orientée objet-opératoire. Sont ensuite examinées des pièces issues du processus d'analyse-recodage : *Les cris du sixième cercle* (2019) de João Svidzinski, hérite de la notion d'objet et la prolonge. *Dans la nef de nos songes 1* (2019) d'Alain Bonardi et *TêTrês* (2001) d'António de Sousa Dias, ont recours de manière explicite aux codes de Risset. Ces trois cas d'héritage se révèlent essentiels pour le recodage d'*Inharmonique*. Cette approche d'analyse-crédation est au cœur de notre activité de compositeurs et de chercheurs.

Abstract

The aim of this article is to show how the authors have appropriated Jean-Claude Risset's analysis-recoding work of *Inharmonique* in their own creations. This extended and revised version of a previous article published in the *Proceedings of the Journées d'informatique musicale 2020*, first presents the object-oriented modelling. It then examines works that have emerged from the analysis-recoding process : João Svidzinski's *Les cris du sixième cercle* (2019), inherits and extends the notion of object. *Dans la nef de nos songes 1* (2019) by Alain Bonardi and *TêTrês* (2001) by António de Sousa Dias explicitly use Risset's codes. These three cases of inheritance will turn out as essential for the re-coding of *Inharmonique*. This analysis-creation approach is at the heart of our activity as composers and researchers.

Introduction

L'articulation entre la composition, l'analyse et l'interprétation du répertoire musical numérique dans un cadre universitaire a été abordée en détail dans la thèse de João Svidzinski (Svidzinski, 2018).

L'approche orientée objet-opératoire a constitué la méthode prédominante de son travail. Suite à cette recherche, nous avons mené en 2019 et 2020 un projet intitulé « Créer, (re)créer, valoriser la musique faisant appel à l'informatique au XXI^e siècle », soutenu par le conseil scientifique de la Maison des Sciences de l'Homme Paris Nord (MSH PN). Le noyau principal du projet était la création mondiale en concert de la version temps-réel interactive de la pièce *Inharmonique* (1977) de Jean-Claude Risset. Ce concert a eu lieu le 28 novembre 2019 à l'auditorium de la MSH PN.

Ce travail soulevait des questions importantes concernant la pratique musicale numérique actuelle, ainsi que la préservation de ce répertoire. Au-delà de la re-crédation de cette pièce, le projet visait à établir un héritage musical et scientifique de Risset. Dans la conjoncture actuelle de la production musicale faisant appel à l'informatique, cette appropriation passe notamment par la pratique de la recherche-crédation.

Pour l'aboutissement de la nouvelle version d'*Inharmonique*, un travail approfondi sur le recodage des

codes originaux de cette pièce a été réalisé. De nouveaux procédés ont été ajoutés afin que la pièce fasse appel à de nouvelles technologies disponibles en musique mixte (Battier, 2003).

Une des particularités de notre approche est donc le rapprochement de l'analyse-recodage et de la création musicale, sous la forme d'allers-retours entre ces pratiques. Ce rapport entre analyse et création a été mis en lumière lors de la création de la version temps réel interactive de la pièce de Risset : en effet, les quatre autres pièces du programme du concert sont des œuvres des compositeurs et chercheurs impliqués dans le recodage d'*Inharmonique*. Les nouvelles pièces constituent à la fois un hommage et un prolongement de la pensée musicale de Risset. Ces pièces sont : *Le bruit d'une tête qui frappe contre les murs d'une très petite cellule* (2000) de José Luís Ferreira ; *TêTrês* (2001) d'António de Sousa Dias ; *Les cris du sixième cercle* (2019) de João Svidzinski et *Dans la Nef de nos songes 1* (2019) d'Alain Bonardi⁽¹⁾.

Une première étape du recodage d'*Inharmonique* a été réalisée par José Luís Ferreira (décédé en 2018) et António de Sousa Dias. Pour la deuxième étape, les deux autres compositeurs-chercheurs, co-auteurs de cet article, ont rejoint l'équipe. Dans cet article, nous allons étudier la manière dont les trois derniers compositeurs-chercheurs ont repris des apports scientifiques musicaux d'*Inharmonique*. Cette discussion se déroulera sous l'angle de la modélisation orientée objet-opératoire. Bonardi, dans sa pièce *Dans la Nef de nos songes 1*, et Sousa Dias, dans *TêTrês*, se sont approprié de manière explicite les codes de Risset. *Les cris du sixième cercle*, quant à elle, est un exemple implicite d'analyse-recodage : il s'agit d'une recherche approfondie et intime sur la composition du son lui-même et sur la place de la voix dans un contexte mixte. Ces deux perspectives d'appropriation coexistent dans le même espace de concert.

Dans un premier temps, nous allons aborder la modélisation orientée objet-opératoire, toile de fond de notre recherche. Ensuite, le recodage des codes d'*Inharmonique* sera analysé, pour enfin tracer un parallèle entre la re-création de cette œuvre et les nouvelles pièces.

1. Modélisation orientée objet-opératoire : le recodage d'*Inharmonique*

L'association avec la recherche de Sousa Dias pour le recodage d'*Inharmonique* était l'une des perspectives de la thèse de doctorat de Svidzinski (2018), qui, entre 2016 et 2018, a analysé la pièce acousmatique *Songes* (1979) de Risset. Cette œuvre, qui date de la même époque que *Inharmonique*, partage certains principes avec elle. Les codes originaux en langage *Music V* sont parfois identiques. C'est le résultat d'une pratique courante des compositeurs-chercheurs de cette époque se tournant vers le partage des codes (et des cartes perforées) (Mathews *et al.*, 1969). Le programme *Music V* facilite cette tâche par le biais de structures et routines PLF (Risset *et al.*, 2002). L'instruction PLF (*play first pass*) permet ainsi d'élargir le langage par l'introduction de sous-routines programmées par l'utilisateur lui-même. Les structures, quant à elles, sont des conteneurs d'informations : par exemple, un objet sonore encapsulé avec une liste de composantes, chacune avec une fréquence, une amplitude et une durée bien déterminées. Avec les instructions PLF, les structures sont instanciées en permettant de réaliser des variations sur une structure à partir d'une spécification de paramètres globaux différents de ceux de cette dernière : ce programme modifie en conséquence la fréquence, l'amplitude et la durée sur toutes les composantes.

1.1 Composition et analyse orientée objet-opératoire

L'approche orientée objet-opératoire est fondée sur la notion d'objet telle que décrite par Horacio Vaggione (1991, 1995). Celui-ci considère un objet comme « une catégorie opératoire, c'est-à-dire un concept technique développé pour réaliser une action musicale donnée, capable d'encapsuler des niveaux temporels différents dans une entité complexe, qui néanmoins, a des limites précises, et qui peut ainsi être manipulée à l'intérieur d'un réseau » (2007). Cette définition « canonique » de l'objet chez Vaggione, permet de créer un parallèle entre la pratique compositionnelle de Risset, par rapport aux sous-routines PLFs et aux structures, et l'objet-opératoire. Dans cette perspective, l'approche orientée objet-opératoire peut être appliquée à l'analyse musicale de pièces faisant appel à l'informatique musicale, ainsi qu'à la composition musicale de ce répertoire. Dans le premier cas, il s'agit d'analyser les réseaux opératoires, c'est-à-dire la composition interne des objets, ainsi que les interactions entre eux. Dans le second cas, il

s'agit de créer le réseau comprenant toutes les échelles temporelles et les dimensions du composable.

1.2. Structures et sous-routines PLFs comme objets

Lorsque Vaggione a introduit sa propre notion d'objet, cette dernière se fondait surtout sur le paradigme de la programmation informatique orientée objet, proposée et élaborée à partir des années 1980. Même s'il s'agit d'une approche informatique, cette idée est appliquée dans un cadre purement compositionnel. Dans notre recherche doctorale, nous avons également exploré cette notion avec une méthode informatique orientée objet, par le biais du langage JavaScript. Cela a été réalisé dans un cadre analytique dans la pièce *Pianotronics 3* d'Alain Bonardi (2016), ainsi que dans des cas purement compositionnels, dans nos propres pièces, comme *Taurus* (2017) pour saxophone et électronique et *Tyqué* (2018) pour clarinette et électronique.

Le recodage d'*Inharmonique*, notamment les sous-routines PLFs, réalisées avec JavaScript, apporte une nouvelle lumière sur l'analyse orientée objet-opérateur des codes *Music V* de Risset. Or, la syntaxe JavaScript prévoit de manière explicite la définition d'objets.

Un des algorithmes les plus emblématiques développé par Risset est la sous-routine PLF 6, qui permet de synthétiser des simulacres de sons de cloche et de gong. Cet algorithme fonctionne de la manière suivante : à partir d'une structure, comportant un certain nombre de composantes (de 5 à 20 environ) - avec des fréquences, amplitudes et durées indépendantes, la sous-routine PLF instancie la structure en calculant tous les attributs de chaque composante par un rapport généré à partir des paramètres d'une note fondamentale (Risset et al. 2002).

Dans un objectif de préservation du répertoire composé avec *Music V* faisant appel à des sous-routines PLF, de Sousa Dias a travaillé sur des méthodes de portage de cette œuvre. Après une première tentative inaboutie de transcription en C++, le compositeur-chercheur a développé un *package* en *OpenMusic* (Sousa Dias, 2003). Une nouvelle méthode a été développée par le même compositeur-analyste en utilisant JavaScript (Sousa Dias, 2018). Ce langage permet entre autres d'être utilisé dans le logiciel *Max*, et surtout de préserver la compréhension des processus d'engendrement du code original. Ce portage correspond aux codes analysés ci-dessous (Figures 1 et 2). Les algorithmes sont ensuite couplés avec *CSound*. Ce langage, à l'instar de JavaScript, permet de conserver un principe idiomatique similaire au style de la syntaxe de *Music V*.

La figure ci-dessous montre le fondement de la syntaxe des structures avec cette sous-routine en JavaScript (figure 1). La fonction `struct_plf6_com` reçoit les arguments : numéro de la structure, numéro de chaque composante, fréquence, durée et amplitude respectives. Grâce à la fonctionnalité `new Object()`, chaque composante se comporte comme un objet logiciel.

Figure 1. Fonction `struct_plf6_com()` dans le code JavaScript du recodage de la sous-routine PLF 6.

```
// =====
function struct_plf6_com( num_struct, c_num, freq, dur, amp ) {
// =====
  this[ num_struct ].comp[ c_num ] = new Object();
  this[ num_struct ].comp[ c_num ].freq = freq;
  this[ num_struct ].comp[ c_num ].dur = dur;
  this[ num_struct ].comp[ c_num ].amp = amp;
}
```

La fonction suivante `struct_plf6` (Figure 2) appelle `struct_plf6_com`, puisqu'elle gère tous les attributs de chaque structure. Dans cette partie du code, les données relatives à chaque structure sont chargées dans le code. Dans un deuxième temps, à chaque fois qu'elle est appelée, une fonction recalcule un index qui multiplierait chaque paramètre en rapport avec les valeurs de fréquence, amplitude et durée données lors de l'instanciation de la structure.

Figure 2. Fonction `struct_plf6()` dans le code JavaScript du recodage de la sous-routine PLF 6.

```
// =====
function struct_plf6( num_struct, num_comp, amp_glb, freq_sub, i_num ) {
// =====
/* ***** PLF 6
- [0] un numero de structure,
- [1] le nombre de composantes de la structure,
- [2] l'amplitude globale realisee par la structure telle que definie,
- [3] la frequence subjective resultant de la structure, utilisee comme
frequence de reference d'apres laquelle la structure sera transposee
pour les enonces NOT affectes par la PLF6,
- [4] le numero de l'instrument devant realiser les composantes,
- trois donnees de reference pour chaque composante:
    frequence,
    duree,
    amplitude.
***** */
var ar_args = new Array;
for(var i=0; i<arguments.length; i++) ar_args[ i ] = arguments[ i ];
if(iverbose != 0) post("plf6 arguments: <",ar_args,">\n");
// this = new Object();
this[ num_struct ].num_comp = num_comp;
this[ num_struct ].amp_glb = amp_glb;
this[ num_struct ].freq_sub = freq_sub;
this[ num_struct ].i_num = i_num;
this[ num_struct ].comp = new Array( num_comp );
this[ num_struct ].durMax = 0;
}

```

Cet exemple corrobore ce qui a déjà été établi lors de l'analyse de *Songes* : Risset utilise une logique orientée objet sans même faire appel à des langages orientés objet, comme le pratiquaient de nombreux programmeurs grâce aux structures en langage C. Cela provient plutôt des usages et savoir-faire liés à la période pionnière de l'informatique musicale : par exemple, le recours à des cartes perforées.

Le recodage aux sous-routines PLFs avec JavaScript contribue d'abord à la pérennité de l'œuvre de Risset et nous permet surtout de mieux comprendre son usage, sans avoir recours à un langage comme *Fortran*. Notre pratique créative a montré que les fondements du raisonnement chez Risset sont similaires aux pratiques actuelles de la musique mixte. Dans *Les cris du sixième cercle*, la notion d'objet est au cœur du réseau opératoire développé. Dans *Dans la nef de nos songes 1*, le modèle d'une structure de son de cloche de Risset est repris à l'identique. Grâce aux recodages des programmes originaux de Risset, les données internes de ces structures ont été rendues disponibles pour ce nouveau cas de création ⁽²⁾.

Le recodage effectué peut être illustré avec le célèbre catalogue *An Introductory Catalogue of Computer Synthesized Sounds* de Jean Claude Risset (1969). Exemple numéro 430, *three successive approximation of a belle sound* (Risset, 1969) ⁽³⁾, recodé dans l'audio 1.

Audio 1. Exemple de cloche générée avec la sous-routine PLF 6 : son de cloche 430 générée par recodage en JavaScript et CSound, (Bonardi *et al.* 2023).
<https://api.nakala.fr/embed/10.34847/nkl.a0db89n9/50bd8c626833417deefd14c7d125f085bca6fd8d>.

2. L'héritage d'*Inharmonique* de Jean-Claude Risset

En plus des pièces *Dans la Nef de nos songes 1* et *Les cris du sixième cercle*, deux autres pièces données lors du concert du 28 novembre 2019 montrent le rapport entre analyse et création. Lorsque José Luis Ferreira et António de Sousa Dias ont entamé (avec la participation de Risset lui-même) le recodage d'*Inharmonique* au début des années 2000, ce processus a tout de suite nourri certaines pièces des deux compositeurs-analystes. Dans le programme du concert du 28 novembre 2019, deux de ces compositions ont été données. *Te?Tre?s* (2001) doit beaucoup à l'analyse-recodage de Risset, surtout par rapport aux procédés issus de l'analyse harmonique (Sousa Dias, 2003). *Le bruit d'une te?te qui frappe contre les murs d'une tre?s petite cellule* (2000) de José Luis Ferreira, exprime quant à elle un sentiment de proximité esthétique avec Risset. Cette pièce a été primée au concours organisé par le Festival « MU?SICA VIVA 2001 », Risset faisant partie du jury.

2.1. L'héritage d'*Inharmonique* dans *Les cris du*

sixième cercle

Les cris du sixième cercle (2019) pour soprano et électronique temps réel a été créée lors du même concert de la nouvelle version temps réel d'*Inharmonique*. Cette pièce, composée alors que son auteur réalisait le recodage d'*Inharmonique*, garde des traces inconscientes de la pièce mixte de Risset, comme l'écriture vocale dans un contexte d'interaction de la voix avec le monde sonore numérique. À première vue, l'appropriation d'*Inharmonique* se situe à un niveau superficiel. Par exemple, la grande inspiration à 10' 41''⁽⁴⁾ d'*Inharmonique* ressemble à l'inspiration finale dans *Les cris du sixième cercle*. Dans les deux pièces, la voix ne chante aucun texte, seulement des voyelles et des éléments vocaux (Risset, 2018) ; les deux pièces offrent un grand degré de liberté à la soliste. Cependant, l'héritage le plus marquant, d'après notre propre auto-analyse, est l'approche de la voix comme une composante du son lui-même et le recours à une méthode orientée objet. Cela se vérifie notamment au niveau du code.

Cette pièce mixte, à l'instar de nos autres pièces récentes, utilise une approche orientée objet en JavaScript. Pour cela, nous faisons appel à des définitions de classes. Le code de la figure 3 montre un exemple de classe de la pièce *Les cris du sixième cercle*. Il s'agit des fonctions qui contrôlent des valeurs globales de sept instances. En effet, le moteur sonore, développé en Faust, comporte sept canaux d'un traitement en chaîne avec trois modules : un harmonizer, un délai par ligne à retard et un granulateur. La majeure partie des fonctions contrôlent un module avec un générateur pseudo-aléatoire. Comme il s'agit de sept instances, une valeur différente est générée pour chacune d'entre elles. Il est aussi possible d'attribuer une valeur indépendante pour chaque instance. Cela se produit par exemple dans la fonction *setRmax()*, qui définit la valeur maximale de rarefaction pour chaque canal. Ce procédé s'observe aussi dans les fonctions *G_setSize*, *G_setDel*, *G_setRare* ? qui figurent en commentaires dans le code ci-dessous, car elles ne sont pas appelées dans cette classe⁽⁵⁾.

Figure 3. Code JavaScript de l'objet *ClassTest()* dans *Les cris du sixième cercle*.

```
var classTest = function() {
    On();

    Inatrix(1);
    Inatrix2(2);
    Fmatrix(1);
    Gmatrix(1);
    Hmatrix(2);
    Amatrix(3);
    setRamp(5000);

    E1(1000,50, 5000, 5000, 5000, 5000, "A.5");
    E2(1000,40, 5000, 5000, 5000, 5000, "A.5");
    E3(1000,30, 5000, 5000, 5000, 5000, "A.5");
    by(500, 0.6);

    Flang(1000,10,0.1,0.8,100,400);
    FlangG(0.15);

    del(1,100,50,10,30);
    MHarmo(1, 100,100,50,30,100);

    RandomD(1, 1);
    RandomG(1, 500, 100, 100, 500, 1000, 0.5, 0.8, 0);
    setGSwitch(1);

    setRmax(0.8, 0.9, 0.7, 0.6, 0.5, 0.6, 0.9);

    setGlobalParameters(1, 300, 50);
    H_setHStretch(1, 1, 1, 1, 1, 1);
    setFd(0.4);

    //G_setSize(100, 200, 300, 400, 500, 600, 700);
    //G_setDel(1000, 2000, 3000, 4000, 5000, 3000, 2000);
    //G_setRare(1, 0.3, 0.5, 0.8, 0.9, 0.1, 1);
    setFdx(0.4);

    ByHarmo(0.3);

    displayAll();
}
```

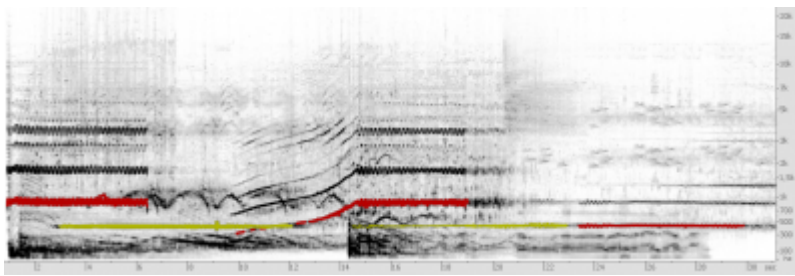
Dans cet objet, trois fonctions contrôlent des échantillonneurs. Ses paramètres définissent l'échantillon joué ? dans le cas ci-dessus « A.5 » ?, les vitesses de lecture ? avec une plage de valeurs définie comme valeur maximale et minimale ? et les vitesses d'instanciation ? avec un contrôle de rarefaction, similaire au paramètre de la granulation. Le début des *Cris du sixième cercle* fait appel à

cet objet. La figure 4 montre une analyse spectrale des 30 premières secondes de la pièce. La fréquence fondamentale de la voix chantée est soulignée en rouge. Il s'agit de la note *la* (880 Hz) qui descend d'une octave (440 Hz) après 24 secondes. Nous avons surligné en jaune une fréquence interne de l'objet sonore, la note *la* (440 Hz). Dans les premières minutes, la soliste anticipe une composante de l'objet, alors que pendant environ 20 secondes, elle reprend la même fréquence. Ce comportement est assez caractéristique d'*Inharmonique*, dans laquelle le « mariage » entre la voix chantée et les objets sonores musicaux est flagrant (Risset, 2018), (Castellengo, 2018).

Cette pièce mixte de Risset peut être comprise comme un « concerto » pour voix chantée et sons inharmoniques. La voix devient régulièrement une composante des sons complexes. Par exemple, vers 9:15?, dans un passage dit des « percussions fluides », la voix anticipe et rejoint les composantes des cloches, constituant une sorte « d'éclairage sur les hauteurs complexes des simulacres des cloches » (Castellengo, 2018). Cette caractéristique est aussi manifeste à 5:18?, où la note chantée correspond exactement au deuxième partiel de la cloche avec un rapport d'inharmonicité. L'indication sur la partition « assez blanc?, pur » accentue le caractère de composant d'un objet complexe.

Contrairement à la pièce de Risset, dans *Les cris du sixième cercle*, l'objet sonore ne contient pas seulement des composantes stables, c'est-à-dire des attributs fixes dans des structures. Dans notre pièce mixte, les objets sont constitués surtout par des fonctions de contrôle global, où l'aleatoire joue un rôle essentiel. Cela s'harmonise parfaitement avec le haut degré de liberté de la voix.

Figure 4. Analyse spectrale des premières 30 secondes de *Les cris du sixième cercle*



Audio 2. Premières secondes de *Les cris du sixième cercle* [0:00 à 0:36?], (Bonardi *et al.* 2023).
<https://api.nakala.fr/embed/10.34847/nkl.a0db89n9/d96ee355255a9a92e5cf97bede2380d2f7851b85>

2.2. L'héritage d'*Inharmonique* dans *Dans la Nef de nos songes 1*

La synthèse sonore spatiale est le point central de la pièce acousmatique *Dans la nef de nos songes 1*, c'est-à-dire la mise en espace de procédés de synthèse sonore, en plus des transformations en temps réel d'un son. Il s'agit d'une nouvelle version de l'installation sonore conçue pour l'Hôtel-Dieu de Tonnerre en Bourgogne, présentée pendant l'été 2018 (Bonardi, 2019).

Dans cette pièce, le compositeur a recours au mode de simulacre de cloche de Risset, en lui rendant hommage. Il étend le principe de la PLF 6 en la rendant réursive, c'est-à-dire que chacune des composantes devient à son tour le partiel de référence d'une nouvelle structure. Ainsi, à partir de 11 composantes, on peut en obtenir 121 ($11 \times 11 = 121$ composantes), ou même 1331 au niveau 3 ($11 \times 11 \times 11 = 1331$ composantes). Le compositeur se sert de l'ambisonie, notamment par ses qualités psycho-acoustiques, pour l'agencement des composantes dans l'espace. L'aspect temporel joue aussi un rôle important pour construire des relations avec la conduite spatiale. Grâce au recours à des fonctions et une matrice de *mapping* développées en JavaScript dans Max/MSP, il est possible d'assigner les composantes spectrales vers les composantes spatiales. En travaillant par exemple à l'ordre 4 d'ambisonie 2D, on manipule neuf composantes spatiales vers lesquelles sont mixées les composantes spectrales selon différents modes (par fréquence, par date d'attaque des partiels). La pièce a été composée après plusieurs étapes de tests en studio, aussi bien en 2D qu'en 3D⁽⁶⁾. Le rendu sonore est globalement assez équilibré sur le plan spectral comme spatial.

Audio 3. Exemple en stéréo de *Dans la nef de nos songes 1* (Extrait binaural, à écouter au casque) [0?00? à 0?19??.], (Bonardi *et al.* 2023).
<https://api.nakala.fr/embed/10.34847/nkl.a0db89n9/59e3d177efb8ed8a5e5c48075e8801ab0739912f>

Pour ce qui relève du comportement spectral dans l'espace, nous avons appliqué certaines relations essentielles de *Dans la nef de nos songes 1* au travail mené sur *Inharmonique*. En effet, un même principe a été utilisé dans la nouvelle version de la pièce de Risset : pour les objets sonores géométriques avec la sous-routine PLF 5 et 6, chaque composante est assignée à une harmonique circulaire différente. Cette mise en espace d'un son de synthèse complexe le rend plus intéressant et plus vivant que sa simple diffusion en mono. De manière similaire, la répartition des partiels dans les composantes spatiales est fondamentale. Il provoque des phénomènes de filtrage et de masquage spectral, ainsi que des modifications du rendu spatial.

Pour le recodage d'*Inharmonique*, l'espace est également essentiel pour homogénéiser les PLF 6 et 5 avec les autres passages de l'électronique – notamment en stéréo. Pour ces passages, la stéréo est plongée dans l'espace ambisonique sous la forme de deux sources ponctuelles. Ainsi, dans la même représentation ambisonique coexistent deux approches spatiales : le champ diffus pour le portage des codes PLF6 et 5 ; la diffusion sous la forme de sources ponctuelles pour les autres passages de la pièce.

2.3. L'héritage d'*Inharmonique* dans *Têtrês*

Composée en 2001, *Têtrês* (lire T3, TeTrois) exploite l'intégration de différents environnements pour la composition, tels que AudioSculpt, OpenMusic, CSound, Max et ProTools. *Têtrês* est une œuvre multicanale, conçue pour l'orchestre de haut-parleurs de Miguel Azguime (Miso Music, Portugal).

Concernant le jeu de superposition d'espaces et la diffusion, le titre de cette œuvre renvoie à trois références :

- la référence à un type d'appartement (au Portugal, un T-3 désigne un séjour et trois chambres) ;
- la référence à Tetris, le jeu d'ordinateur ayant pour but l'enchevêtrement de petites pièces avec des tailles et des dispositions différentes dans le plus court délai possible ;
- la référence à tétra, élément grecque qui signifie quatre, une allusion au nombre de canaux, lorsqu'on prononce le titre *TeTrois* de manière un peu abrupte et un peu maladroite.

Ainsi, en ce qui concerne l'articulation entre l'espace d'écoute et l'espace de l'œuvre dans *TeTrois*, le but principal est de créer une superposition d'espaces qui tournent entre eux et par rapport à l'auditeur où l'utilisation de l'espace de diffusion est expérimentée d'une façon assez libre. En effet, cette œuvre utilise systématiquement des fichiers stéréophoniques dont les points d'écoute « droite-gauche » sont ré-assignés à des points arbitraires dans l'espace. Ceci implique une composition de mouvements : d'abord le mouvement global défini par le déplacement des deux points d'écoute et d'autre part le mouvement « droite-gauche » parfois déjà présent dans les fichiers d'origine. Toutefois, bien que *TeTrois* soit une œuvre quadraphonique, elle a été conçue pour une écoute frontale, car l'auditeur est supposé rester assis, immobile et les haut-parleurs frontaux ont une importance plus grande que les autres.

Dans cette œuvre, le lien avec *Inharmonique* s'observe principalement à travers deux procédés : le recours à la cascade harmonique, dans le but de créer des textures oscillantes, entre mélodie et arpège et le recours à des sons sans octave fixe, une déclinaison du glissando perpétuel. En particulier, la cascade harmonique est envisagée sous divers aspects et échelles, soit l'introduction de battements proches d'une modulation d'amplitude de faible amplitude, jusqu'à l'effet d'une modulation en anneau conduisant à des sons avec différentes échelles de trémolo ; soit l'exploration de cette procédure dans des sons dont le spectre sonore s'éloigne de l'effet gamme-accord par le recours à des relations entre les différents harmoniques plus éloignés.

Audio 4. Exemple de textures oscillantes dans *Têtrês* [0?29? à 0?57?], (Bonardi *et al.* 2023).

<https://api.nakala.fr/embed/10.34847/nkl.a0db89n9/467cb0f91a7753ba65ac9af1b738d42f3b0ae4ee>

Pour accomplir ces effets, et de pair avec leur implémentation sur CSound, un patch Max⁽⁷⁾ ayant deux caractéristiques intéressantes a été développé :

- la forme d'onde est éditable, soit en temps réel, soit chaque harmonique à la fois, bien que la table utilisée soit limitée à l'utilisation des 64 premières harmoniques d'un son ;
- la forme d'onde peut résulter de l'interpolation entre deux formes d'onde ; ceci crée un double mouvement : le mouvement de l'entrelacement des battements des harmoniques et la transformation de celles-ci concernant la composition du spectre.

Audio 5. Exemple de cascade harmonique dans *Te? Têtrès*, (Bonardi *et al.* 2023).

<https://api.nakala.fr/embed/10.34847/nkl.a0db89n9/4e64c4c63360676078b8ab37c06b68911a1b6540>

Les autres caractéristiques du patch concernent le contrôle des paramètres comme la vitesse de la cascade, qui peut être définie en Hertz ou en secondes.

Figure 5. Patch *_jcr.HarmonicCascade.maxpat*, version 2021, et souspatches

jcr.HarmonicCascade-SPinterpolator.maxpat et *jcr.HarmonicCascade-SPeditor.maxpat*



Conclusion

Dans cet article, nous avons alternativement parlé d'héritage et d'appropriation. De prime abord, nous serions tentés de concevoir l'appropriation comme un processus unilatéral. L'appropriation peut concerner un créateur par rapport à ses propres recherches-créations antérieures : Risset lui-même s'appuiera sur les développements des matériaux qu'il a mis à disposition de la communauté musicale (Risset *et al.*, 2002). Ce processus était déjà implicite dans les possibilités offertes par la codification des partitions sonores permettant l'avancement de la composition et de la recherche. En effet, ces partitions sonores

peuvent être utiles pour accumuler le savoir-faire : elles servent non seulement en tant que recettes, mais également en tant que partitions pouvant suggérer des idées pour la composition microstructurale. Comme pour l'analyse des partitions musicales traditionnelles, il s'agit d'une manière d'apprendre l'art de la composition ; regarder les partitions sonores est une manière de partager l'expérience avec d'autres en ce qui concerne la composition des sons eux-mêmes. (Risset, 1995)

Ainsi, notre recherche montre que la relation entre création et analyse peut être plus complexe, et se fonder sur des allers-retours. Ainsi, avec le travail de synthèse spatiale, notamment dans *Dans la nef de nos songes I*, nous avons développé une approche qui a été ensuite utilisée dans la nouvelle version d'*Inharmonique*. *Dans la nef de nos songes I* s'est nourrie du modèle de synthèse de cloche d'

Inharmonique, et *Inharmonique* dans sa nouvelle version s'est inspirée pour partie de *Dans la nef de nos songes 1*.

La pièce *Les cris du sixième cercle*, quant à elle, expose une nouvelle dimension opératoire des objets, qui fait place à des générateurs aléatoires qui contribuent à une « dynamisation » de la notion de structure. *TêTrês* montre un exemple d'appropriation encore plus explicite, car le compositeur se sert du système logiciel développé dans le recodage d'*Inharmonique*, ainsi que d'une variation des codes originaux de Risset.

Les trois cas d'héritage, avec le recodage d'*Inharmonique*, cohabitent naturellement dans le même espace de concert, chaque pièce étant authentique et propre à chaque compositeur-chercheur. Ainsi, cela prouve que le recodage ? en prenant en compte toutes les questions de l'archivage, de la pérennisation et de la valorisation d'un répertoire ? et l'analyse-cr ation prennent tout leur sens lorsqu'ils sont r alis es dans le cadre pratique du concert. Ce dernier propose   un moment donn e un r seau de pi ces qui se nourrissent les unes des autres, occasion dans le cas particulier du r pertoire avec  lectronique en temps r el d'une interpr tation fond e non seulement sur les choix des musiciens mais aussi sur la n cessaire remise en chantier et en route des codes informatiques.

Donn es et codes

Bonardi, Alain, Sousa Dia Ant nio, Svidzinski Jo o (2023) « Extraits audio *Inharmonique* (1977), *Les cris sixi me cercle* (2019), *Dans la nef de nos songes* (2019), *T Tr s* (2001)» [Sound] NAKALA.
<https://nakala.fr/10.34847/nkl.a0db89n9>

1. *Sottovoce* (1995) de Jose  Manuel Lopez Lopez (qui n'a pas particip  directement   cette recherche), pi ce acousmatique, rejoint le programme afin de donner un exemple de l'exploration de la voix dans un contexte  lectroacoustique. Malgr  l'importance de cette pi ce, elle ne sera pas abord e dans cet article.

2. Bonardi a utilis  surtout le recodage de Dodge et Jerse (1997).

3. Voir le livret en ligne :

<https://archive.org/details/an-introductory-catalogue-of-computer-synthesized-sounds/page/n77/mode/2up>

4. Toutes les indications de chronom tre d'*Inharmonique* correspondent   la partition originale de l'uvre.

5. Pour ce qui concerne les strat gies compositionnelles orient es objet de Jo o Svidzinski, cf. (Svidzinski, 2018).

6. Pour plus de pr cision sur l'ambisonie, notamment le champ diffus, voir (Colafrancesco, 2015).

7. https://github.com/asousadias/Risset_MaxPatches/tree/master/Risset_02_HarmonicCascade

Pour citer ce document:

Alain Bonardi,  « Analyse cr ative en recherche-cr ation : trois exemples d'appropriation   partir du recodage d'*Inharmonique* de Jean-Claude Risset  », *RFIM* [En ligne], Num ros, n  9 - Inharmonique (1977) de Jean-Claude Risset : cr ation, r -cr ation et avenir, Mis   jour le 11/09/2023

URL: <http://revues.mshparisnord.org/rfim/index.php?id=668>

Cet article est mis   disposition sous [contrat Creative Commons](#)