



HAL
open science

Étude et appropriation de l'augmentation instrumentale proposée par Jean-Claude Risset dans l'élaboration de ses duos pour un pianiste

Sébastien Clara

► **To cite this version:**

Sébastien Clara. Étude et appropriation de l'augmentation instrumentale proposée par Jean-Claude Risset dans l'élaboration de ses duos pour un pianiste. Journées d'Informatique Musicale (JIM 2018), May 2018, Amiens, France. hal-01793804

HAL Id: hal-01793804

<https://hal.science/hal-01793804>

Submitted on 17 May 2018

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.

ÉTUDE ET APPROPRIATION DE L'AUGMENTATION INSTRUMENTALE PROPOSÉE PAR JEAN-CLAUDE RISSET DANS L'ÉLABORATION DE SES DUOS POUR UN PIANISTE

Sébastien Clara

Université Lyon – Saint-Étienne
sebastien.clara@univ-st-etienne.fr

RÉSUMÉ

En 1989, Jean-Claude Risset est invité au Media Lab du M.I.T. en tant que compositeur en résidence. Il élabore un dispositif réactif autour d'un piano reproducteur Disklavier commercialisé par la firme Yamaha. Avec cette gamme de piano, Yamaha effectuait une mise à jour au standard de la fin du XX^e siècle du piano dit mécanique. Les pianos mécaniques génèrent de la musique automatiquement et ils ont bénéficié d'une grande popularité au début du XX^e siècle. Le piano reproducteur est quant à lui capable d'enregistrer l'exécution d'une pièce en temps réel et de la reproduire fidèlement. Jean-Claude Risset a détourné la fonction de ce piano reproducteur pour construire un instrument augmenté et compose avec celui-ci une série de pièces qu'il intitule *Huit esquisses en duo pour un pianiste*.

Dans cet article, nous retraçons les différents développements que nous avons réalisés autour des pièces composées par Jean-Claude Risset durant cette résidence. Nous étudierons en premier l'augmentation instrumentale établie par Jean-Claude Risset pour personnaliser et varier les réponses de son automate suivant les particularités d'une interprétation. Puis, nous relaterons notre appropriation de l'augmentation de son instrument.

1. INTRODUCTION

En 1989, Jean-Claude Risset expérimente plusieurs modalités d'interaction entre un musicien et un automate lors de sa résidence au M.I.T. Il conclut ce travail de recherche par la composition d'une série de pièces qu'il intitule *Huit esquisses en duo pour un pianiste*. Son dispositif se compose d'un ordinateur et d'un piano acoustique, mais de type automatophone.

Un automatophone est un instrument de musique couplé à un mécanisme qui permet de simuler le jeu d'un instrumentiste. Il peut être actionné automatiquement ou par un humain. Dans l'imaginaire occidental, l'automatophone correspond à l'orgue de Barbarie des joueurs de rue ou au piano dit mécanique. La culture populaire a utilisé ces pianos comme un personnage des saloons de l'ouest sauvage des États-Unis du début du XX^e siècle, jouant sans peur et

inlassablement des Ragtimes très en vogue à ce moment-là.

Le principe du codage de la musique des automatophones voit le jour à Bagdad au IX^e siècle. La fratrie Banu Musa met au point un automate joueur de flûte, mû par la force hydraulique. Les leviers qui obstruent les trous de la flûte sont commandés par un cylindre à picots. La mélodie est programmée suivant la disposition spatiale des picots sur le cylindre. L'écoulement de l'eau dans le système met en mouvement le cylindre. Les picots déclenchent alors la levée et la descente des leviers et modifient la note émise par la flûte [3].

De nombreux compositeurs¹ se sont appropriés ces automates, mais ces machines génèrent de la musique régulière, inapte à entamer un dialogue expressif avec un musicien. Les automatophones étaient généralement utilisés pour reproduire la musique populaire ou religieuse, mis à part quelques exemples créatifs notables durant le XX^e siècle [5].

L'essor de l'électronique pendant le XX^e siècle a augmenté significativement l'importance de l'automatisation dans nos sociétés et a entraîné la création de nouveaux outils destinés à la production sonore, notamment le synthétiseur, le séquenceur ou l'arpégiateur². Cependant, le principe de codage de la musique du cylindre à picots a été transposé³ sur ces nouveaux instruments, avec sa rigueur métronomique jusqu'à aujourd'hui dans nos stations audionumériques.

À la fin des années 1950, Max Mathews crée le premier logiciel de synthèse sonore. Toutefois, les ordinateurs de cette période demandent plusieurs minutes pour calculer une seconde de son. Dans ces

¹ Joseph Haydn (Hob. XIX: 1–32), Wolfgang Amadeus Mozart (K. 594, 608 et 616), Ludwig van Beethoven (Op. 91, arrangé ultérieurement pour orchestre), Igor Stravinsky (Étude pour Pianola), György Ligeti (certaines études pour piano), etc. À noter que Conlon Nancarrow a consacré sa production au piano automatique.

² La fonction de l'arpégiateur est de transformer un accord en arpège suivant un tempo donné. Il fonctionne en temps réel et il dispose de plusieurs paramètres pour varier ses réponses. À partir des années 1980, cette fonction est régulièrement implantée dans les synthétiseurs. Aujourd'hui, elle est disponible par défaut dans la majorité des stations audionumériques ou par l'adjonction de logiciels tiers.

³ De même pour l'industrie avec l'automatisation du métier à tisser par rubans perforés ou de l'informatique avec son système d'entrée-sortie par cartes perforées.

conditions, l'interaction musicale entre une machine numérique et un humain est impossible. À la fin des années 1960, Mathews et Moore mettent au point le système Groove pour pallier cette carence. Cependant, Groove est un système hybride. En effet, l'ordinateur n'est toujours pas capable de calculer des sons en temps réel, mais il peut générer du contrôle en temps réel. L'ordinateur commande alors un synthétiseur analogique [9].

Les moyens technologiques limitent les possibilités artistiques et c'est à partir des années 1980 que la création de dispositifs interactifs ou réactifs [1] se démocratise. Les ordinateurs deviennent plus puissances, plus petits, plus abordables et ils ont à leur disposition de plus en plus de mémoire. La norme MIDI permet de découpler les contrôleurs et les modules de production sonore et de créer des dispositifs hétérogènes dont l'ordinateur peut devenir un maillon.

L'avènement de la norme MIDI ravive aussi l'intérêt pour les interfaces homme-machines alternatives – quête qui démarre avec les premiers instruments électroniques comme les efforts fournis par Léon Theremin, Maurice Martenot ou encore Donald Buchla, mais pour le moment aucune n'a supplanté la popularité des contrôleurs qui reproduisent la forme des instruments acoustiques ou des équipements de studio [4].

Une autre approche de contrôle d'un dispositif consiste à augmenter les capacités d'un instrument acoustique ou électronique. Au M.I.T., Tod Machover fait partie des pionniers de ce type de démarche avec son projet d'hyperinstrument.

The hyperinstrument project was started in 1986 with the goal of designing expanded musical instruments, using technology to give extra power and finesse to virtuosic performers. Such hyperinstruments were designed to augment guitars and keyboards, percussion and strings, and even conducting. [...] The research focus of all this work is on designing computer systems (sensors, signal processing, and software) that measure and interpret human expression and feeling, as well as on exploring the appropriate modalities and innovative content⁴.

Deux modalités de contrôle sont possibles pour augmenter un instrument. La première consiste à ajouter des contrôleurs à l'instrument. Cette méthode implique l'apprentissage de nouveaux gestes de la part du musicien pour maîtriser ce type d'augmentation. L'exemple emblématique de cette méthode est l'augmentation de la guitare. L'amplification électrique a permis à cet instrument acoustique de faible intensité sonore de devenir un instrument soliste de premier plan

⁴ <http://opera.media.mit.edu/projects/hyperinstruments.html>, dernière consultation le 20/03/18.

dans l'orchestre. Mais l'électricité a aussi permis de modifier radicalement son timbre. L'insertion d'une ou plusieurs pédales d'effet entre la guitare et l'amplificateur fournit des contrôles en temps réel pour altérer le timbre par des gestes inusités par la technique instrumentale traditionnelle de cet instrument.

La seconde méthode pour réaliser un instrument augmenté est transparente pour le musicien. À partir de différents moyens techniques, un système capture une performance instrumentale classique pour s'alimenter en données et générer des réponses sonores. Jean-Claude Risset a utilisé cette dernière méthode d'augmentation pour construire son dispositif. En effet, Yamaha commercialise un nouveau type de piano augmenté de capteurs et de moteurs à partir de 1987. Le standard MIDI permet à ce piano de dialoguer avec d'autres machines équipées de ce protocole de communication. Les capteurs indiquent les touches et les pédales actionnées par un pianiste et les moteurs entraînent les touches et les pédales du piano. Un ordinateur connecté à ce type de piano peut donc « écouter » un pianiste jouer ou commander le piano.

En 1989, les champs technique et conceptuel étaient donc propices pour que Jean-Claude Risset puisse réaliser son dispositif et composer sa série de pièces. Le Disklavier de Yamaha reprend le principe du piano reproducteur élaboré au début du XX^e siècle. Le but de ce piano est de saisir l'expressivité⁵ d'une interprétation en temps réel et de la restituer le plus fidèlement possible. La norme MIDI a permis à Jean-Claude Risset de détourner la fonction de ce type de piano pour fabriquer un instrument augmenté. Dans cet article, nous nous focalisons exclusivement sur la solution développée par Jean-Claude Risset. Toutefois, des compositeurs ou des Réalisateur·e·s en Informatique Musicale ont proposé des dispositifs instrumentaux alternatifs [6].

2. PRÉSENTATION DES DUOS POUR UN PIANISTE

Jean-Claude Risset compose ses différents duos au cours de deux résidences au M.I.T. La première date de 1989 et il nomme ces études *Duo pour un pianiste : huit esquisses pour piano MIDI et ordinateur*. Le titre des pièces évoque l'opération d'augmentation mise en œuvre.

Double. Le pianiste joue seul, puis à la reprise l'ordinateur ajoute des ornements. Ces ornements pré-enregistrés interviennent quand le pianiste joue certaines notes ; leur tempo peut être influencé par le tempo du pianiste.

⁵ Cette problématique est déjà présente au XVIII^e. Dans son traité sur *l'art de noter les cylindres*, Marie-Dominique-Joseph Engramelle signale l'écart entre la notation musicale et la pratique des musiciens et focalise son attention sur la transcription de ces détails d'expression [2].

Miroirs. À chaque note jouée par le pianiste répond la note symétrique par rapport à certaine note du clavier – un procédé utilisé dans la seconde Variation op. 27 de Webern, citée au début (et aussi à la fin, à l'écrevisse). Les centres de symétrie et les retards de réponse sont variés au cours de la pièce.

Extensions. Aux arpèges joués par le pianiste, l'ordinateur ajoute des arpèges transposés.

Fractals. À chaque note jouée, l'ordinateur ajoute cinq notes espacées d'une octave altérée. Alors les mélodies jouées par le pianiste sont étrangement distordues : une montée d'une octave est perçue comme une descente d'un demi-ton.

Agrandissements. Comme dans Extensions, l'ordinateur ajoute des notes, mais les intervalles sont non pas transposés, mais agrandis dans des rapports allant de 1,3 à 2,7, ce qui agrandit les intervalles mélodiques et harmoniques.

Métronomes. Au début, l'ordinateur répond en canon, sur des hauteurs transposées et à des tempos plus rapides. Puis il joue simultanément plusieurs séquences à des tempos différents. Enfin il répète les mêmes hauteurs, mais encore à des tempos métronomiques différents, soit établis à l'avance, soit décidés par le pianiste.

Up-down. Des arpèges d'octaves altérées sont déclenchés par le pianiste, qui voit ses notes proliférer. Le tempo des arpèges est établi d'abord par le tempo du pianiste ; puis la note qu'il joue ; enfin par l'intensité du jeu.

Résonances. Au début et à la fin, l'ordinateur tient de longs accords. Dans la section médiane, le pianiste joue des accords muets : les cordes sont mises en résonance par les séquences jouées par l'ordinateur [7].

La seconde résidence date de 1991 et Risset l'intitule *Trois études en duo*. Le principe et le dispositif sont identiques à ceux de la première résidence. Cependant, il symbolise les opérations d'augmentation utilisées dans ces études par des personnages tirés de la mythologie grecque.

Echo. L'ordinateur fait écho au pianiste, mais il ne s'agit pas d'une simple répétition : les échos sont transposés en hauteur et en tempo, et ils interviennent avec des retards variés. Cette étude tire parti des résonances sur la table d'harmonie des notes jouées par le pianiste aussi bien que par son partenaire virtuel.

Narcisse. Ici la relation s'apparente à la réflexion par un miroir : les intervalles mélodiques sont renversés – une quarte devient une quinte et vice-versa. Le centre de symétrie est une note du clavier qui varie dans le courant de la pièce. La réflexion peut aussi intervenir avec un retard variable.

Mercure. Il s'agit d'une sorte de scherzo dans lequel le pianiste déclenche des arpèges à différentes vitesses. Le tempo des arpèges dépend soit du tempo de certains groupes de notes jouées par le pianiste, soit de la hauteur, soit de l'intensité de son jeu. Les arpèges investissent l'espace des hauteurs un peu à la façon des formes d'un kaléidoscope.⁶

Le glissement entre le mythe d'Écho et de Narcisse et les opérations musicales de canon et de renversement est clair. Toutefois, l'invocation de Mercure est moins évidente à saisir. En effet, cette divinité est le protecteur du commerce, des voleurs, des voyages et le messager des dieux. Une technique de communication longue distance, ancêtre du sémaphore que l'on trouve durant l'Antiquité grecque, transmet une information par un signal visuel de relais en relais jusqu'à son destinataire.

Clytemnestre : Héphaïstos, en lançant de l'Ida une flamme brillante, puis, de feu en feu, le message igné est venu jusqu'à nous. L'Ida l'a envoyé au rocher d'Hermès, à Lemnos ; de l'île, l'éclatante lumière a été captée au troisième relais, au mont Athos voué à Zeus, puis franchissant d'un bond vigoureux les crêtes de la mer, l'ardent flambeau voyageur s'est élancé avec joie <...> et la torche a transmis sa clarté dorée, soleil de nuit, au poste de guet du Makistos.⁷

La propagation d'une note par les arpèges générés automatiquement pourrait se comparer à la propagation de l'information de la chute de Troie de la tragédie d'Eschyle. De torche en torche, ce message court au-

⁶ Note de programme, <http://brahms.ircam.fr/works/work/11507/#program>, dernière consultation le 20/03/18.

⁷ ESCHYLE, *Agamemnon*, La Différence, Paris, 2004, p. 177.

dessus des montagnes et de la mer jusqu'à Clytemnestre. Toutefois, notre proposition sur la raison de Jean-Claude Risset d'intituler cette pièce *Mercury* reste une hypothèse. Aucun des documents que nous avons consultés n'apporte de précision sur cette question.

Par la suite, Jean-Claude Risset fusionne les travaux produits durant ces deux résidences lorsqu'il interprète ces pièces. Le nom générique *Duo pour un pianiste* fait donc référence indistinctement aux pièces de ces deux résidences.

3. LES PRINCIPAUX ALGORITHMES DES DUOS

Nous avons recensé dans cette section les principales briques algorithmiques utilisées par Jean-Claude Risset pour élaborer son augmentation instrumentale. Pour effectuer ce travail, nous avons consulté les patches Max réalisés par le compositeur. Pour ce faire, nous avons utilisé l'émulateur Basilisk II des versions 7 et 8 du système d'exploitation de Macintosh. Sous cet émulateur, nous avons utilisé Max 3 pour sauvegarder les patches en version texte. Nous avons pu alors les ouvrir et les adapter avec Max/MSP 7 sous un système d'exploitation récent et entièrement fonctionnel.

Pour alimenter les algorithmes de ces duos, Jean-Claude Risset exploite les données générées en temps réel par le pianiste. Les renseignements transmis par une note MIDI sont sa hauteur, sa vitesse, le moment de son attaque et de son relâchement. Une horloge est couplée aux transitoires des notes et elle permet de déterminer la durée des notes et la durée entre les attaques des notes. Ces données peuvent être sauvegardées pour réaliser des échos par exemple ou être consommées dans l'instant.

Risset programme principalement des algorithmes qui transforment la hauteur et la durée des notes jouées par un pianiste. Mais il parallélise aussi les traitements tout en les configurant différemment. Il peut ainsi varier la densité des notes générées par l'automate. De plus, l'action de la pédale *una corda* du piano stoppe tous les algorithmes en cours d'exécution et oblige l'automate au silence. Nous classons ces algorithmes en trois catégories avec pour critère de discrimination les traitements réalisés sur la hauteur, le temps et les traitements conditionnels.

3.1. Traitement de la hauteur

- Transposition :
`noteMidi + transpoDemiTon`
- Renversement :
`(pointDeSymétrie * 2) - noteMidi`

- Accord et arpège à partir d'une note :

Exécution parallèle de l'algorithme de transposition avec un contrôle du délai d'exécution sur chaque branche. Si les délais sont identiques, les notes seront jouées simultanément. Sinon le programme égrainera l'énoncé des notes dans le temps et produira un arpège.

- Agrandissement :
`noteMidi * coefAgrandissement`

3.2. Traitement temporel

- Agrandissement d'une durée :
`duréeNoteMidi * coefAgrandissement`
- Détermination d'un tempo après trois notes attaquées :
`120 / (dateNoteOnT3 - dateNoteOnT1)`
- Répétition d'une note :
`Tant Que répétitionPorte est ouverte Faire ...`

3.3. Traitement conditionnel

- Sensibilisation d'une réponse :
`Si vitesseNoteMidi < vitesseSeuil`
`Alors coefAgrandissement <-- 1.3`
`Sinon coefAgrandissement <-- 0.7`
- Déclenchement d'une action suivant un groupement de notes jouées :
`Si (noteMidi == noteMidiPivot) &&`
`(noteMidi-1 == noteMidiPivot-1)`
`Alors déclenchement d'une action`

Dans le vocabulaire de Jean-Claude Risset, ce traitement conditionnel est encapsulé dans l'objet : *trig2 noteMidiPivot-1 noteMidiPivot* (cf. figure 1).

4. MISE EN ŒUVRE DES OPÉRATIONS

La figure 1 représente la première partie du patch Max de la pièce *Miroirs* et elle nous indique la progression du paramétrage de l'opération de renversement. À l'initialisation du programme, le point de symétrie de l'opération reçoit la note MIDI 69. Le délai avant que l'automate joue les notes renversées est de 175 ms. La première note pivot (la note MIDI 27) est activée. À ce moment, toutes les notes jouées par le pianiste seront renversées suivant la note point de symétrie avec une latence d'exécution de 175 ms. Toutes les notes, sauf la note MIDI 27. Lorsque celle-ci sera jouée par le pianiste, un nouveau paramétrage

algorithmique sera mis en place. Le point de symétrie deviendra la note MIDI 85 et la prochaine note pivot sera la note MIDI 64. Le délai avant de jouer les notes renversées n'est pour le moment pas modifié.

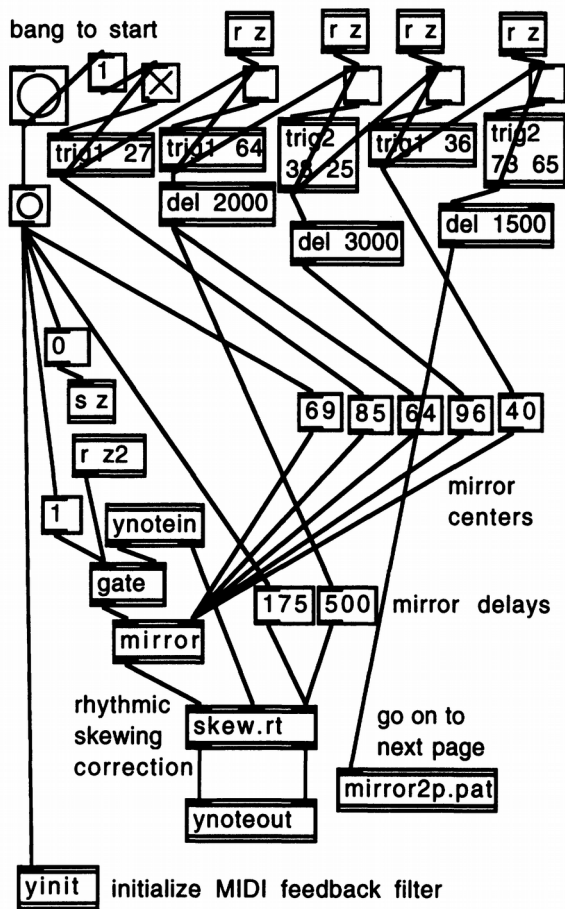


Figure 1. Première partie du patch Max de la pièce Miroirs [8].

La pulsation de cette pièce est de 120 à la noire. Jean-Claude Risset note uniquement les barres de mesures du premier système. Nous sommes à 4/4, mais il ne l'écrit pas. Ces premières barres de mesures montrent la carrure générale des prochains systèmes, mais l'interprète n'a pas à suivre cette contrainte. Le premier déclenchement est placé sur le premier temps de la quatrième mesure. La première modification des paramètres de l'algorithme est donc programmée six secondes après le début de la pièce. La deuxième est prévue dix secondes après cette première modification et ainsi de suite.

Dans *Miroirs*, le paramétrage de l'opération est fixé par le compositeur, mais il évolue durant toute la pièce à une fréquence assez soutenue comme nous venons de le voir. Ces modifications récurrentes altèrent la régularité des réponses générées par l'automate. Nous représentons cette réalisation par la figure 2. À

l'initialisation du système, le traitement reçoit des arguments opératoires puis les notes à traiter, mais certaines notes provoquent des changements de paramètres.

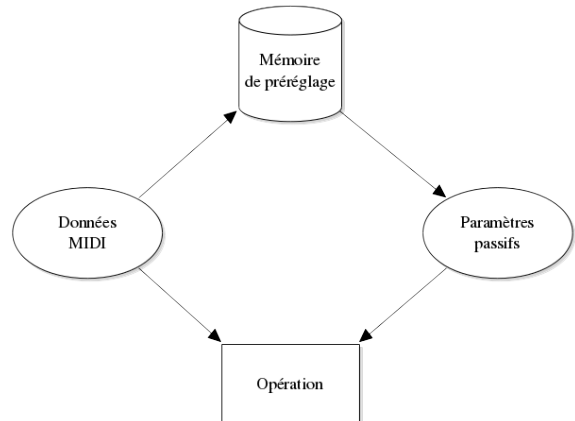


Figure 2. Automatisation passive, mais avec une évolution des paramètres du traitement.

D'autre part, pour rendre les réponses de l'automate actives et sensibles au jeu du pianiste, Jean-Claude Risset conditionne les valeurs du paramétrage d'un algorithme aux caractéristiques des notes jouées en temps réel. De même que pour un paramétrage passif, ces conditions évoluent durant le déroulement de la pièce. Par exemple, dans la pièce *Mercur*e « le tempo des arpèges dépend soit du tempo de certains groupes de notes jouées par le pianiste, soit de la hauteur, soit de l'intensité de son jeu ».

Deux types de configurations alimentent les opérations, les arguments passifs et actifs. Mais en plus, ces deux types d'arguments évoluent durant l'exécution de la pièce. La figure 3 représente l'ensemble des possibilités de paramétrage d'une opération utilisé par Jean-Claude Risset pour élaborer la réactivité de son système.

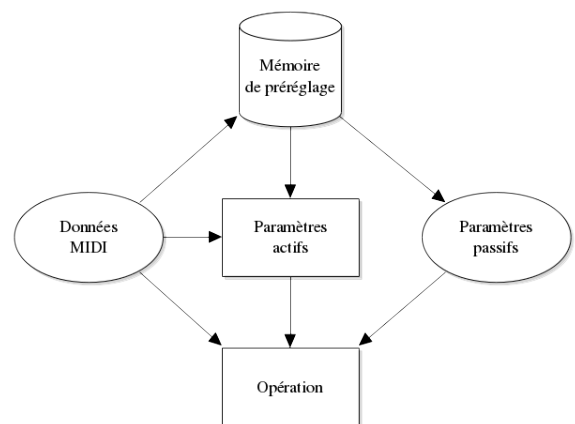


Figure 3. Automatisation active.

De plus, Jean-Claude Risset emploie des rapports toujours complexes entre les paramètres d'une opération afin de pulvériser la plupart des relations synchrones dans ses duos. Dans la première partie de *Miroirs* (figure 1), le rapport entre le premier délai et le deuxième avant que l'automate produise une réponse est de 20/7. Par ailleurs, le premier délai de réponse de l'automate n'est pas synchronisé au tempo du pianiste, noté 120 à la noire et provoque un écart entre la pulsation du pianiste et celle de l'automate. À la deuxième note pivot, le délai de réponse de l'automate (de la durée d'une noire) et le tempo du pianiste se synchronisent. Mais dans la seconde partie du programme, Risset restaure cette dichotomie entre les pulsations et il l'entretiendra jusqu'à la fin de la pièce.

Dans les différents duos, l'automate génère généralement des réponses asynchrones par rapport aux données produites par le pianiste. Néanmoins, les protagonistes de ces duos utilisent le même instrument. L'auditeur ne perçoit pas un dialogue ou le développement d'une opération compositionnelle, mais un monologue riche, plus ou moins dense. Dans le dispositif établi par Jean-Claude Risset, l'automate remplit la fonction d'augmentation instrumentale en accompagnant ou en amplifiant le jeu du pianiste. Pour varier le discours musical et briser la monotonie des réponses de son système, Risset fait fluctuer par différentes méthodes les relations qui lient l'automate au pianiste.

5. APPROPRIATION DES DUOS

Nous nous sommes approprié les principes établis par Jean-Claude Risset dans l'élaboration de ces duos pour présenter un instrument augmenté lors de l'exposition IA² organisée dans les locaux de l'université de Saint-Étienne pour animer les événements alternatifs (*Off*) de la Biennale Internationale Design 2017 de Saint-Étienne. Pour ce faire, nous disposons d'un piano Schimmel TwinTone. Ce modèle est un piano droit équipé d'un synthétiseur. Les entrées MIDI du synthétiseur ne contrôlent pas la partie mécanique du piano. Cette caractéristique technique implique que notre automate peut produire des sons uniquement synthétiques.

L'avantage de cette contrainte technique est que nous n'avons pas été confrontés au problème de réinjection des données dans la boucle établie entre les entrées/sorties MIDI de l'ordinateur et du piano ni aux distorsions rythmiques induites par des latences variables des événements pour commander le piano. Ces problèmes et leur solution sont discutés dans l'article rédigé par Jean-Claude Risset et Scott Van Duyne [8]. L'inconvénient de cette contrainte technique est la piètre mixité entre les parties acoustique et synthétique du piano. Pour homogénéiser le résultat sonore de notre installation, nous avons décidé d'utiliser uniquement le son du synthétiseur.

Notre augmentation instrumentale propose au public d'explorer six modes opératoires sensibles. Les notes les plus graves du piano permettent de changer de mode et l'action de la pédale *una corda* stoppe l'automate. Pour faciliter l'interaction entre le public et l'automate, nous affichons son état sur un écran. Nous utilisons ce retour visuel pour présenter les différents contrôles, stipuler et donner une explication du mode opératoire sélectionné et indiquer les notes jouées par l'automate (cf. figure 4).

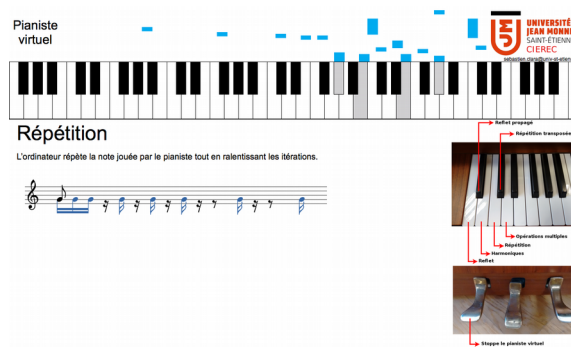


Figure 4. Retour visuel de l'état de notre automate.

Pour stabiliser l'expérience des utilisateurs, nous avons utilisé uniquement une configuration active pour conduire l'augmentation de notre instrument. Fixer le paramétrage du système participe à la compréhension globale des utilisateurs et accélère l'apprentissage du fonctionnement du dispositif. Sensibiliser les réponses de l'automate décuple l'expérience récréative de l'installation et procure une plus grande satisfaction lorsqu'on maîtrise musicalement cette augmentation. Pour les personnes qui possèdent aucune notion musicale, le retour visuel donne une clé d'écoute pour comprendre le principe d'un instrument augmenté.

De plus, si l'installation n'est pas utilisée un certain temps, elle émet un appel. Suivant le lieu d'exposition, nous avons implanté deux types de signaux. Le premier joue des extraits du répertoire occidental classique pour piano. La seconde solution enregistre le jeu des utilisateurs de l'installation et transforme ces données pour produire un appel.

Nous avons réalisé notre installation avec Max/MSP. Cependant, la majeure partie de notre système est développée en JavaScript dans des objets Max dédiés. Le fonctionnement asynchrone des tâches JavaScript nous a semblé approprié pour répondre à nos attentes. En effet, une tâche JavaScript est une fonction que l'on gère dans le temps. On peut lui spécifier un nombre d'exécutions, une condition d'arrêt particulière, la stopper manuellement, etc⁸. Pour ne pas saturer l'espace sonore par une accumulation des interventions de notre automate, nous avons fixé le nombre de tâche pouvant fonctionner en parallèle à trois.

⁸ <https://docs.cycling74.com/max5/vignettes/js/jstaskobject.html>, dernière consultation le 20/03/18.

Les principales actions de notre automate sont déclenchées lors de l’attaque ou du relâchement d’une note et suivant l’événement, notre automate réagit distinctement. Selon le mode opératoire sélectionné et ses conditions d’exécution, une réponse peut être délivrée (cf. figure 5).

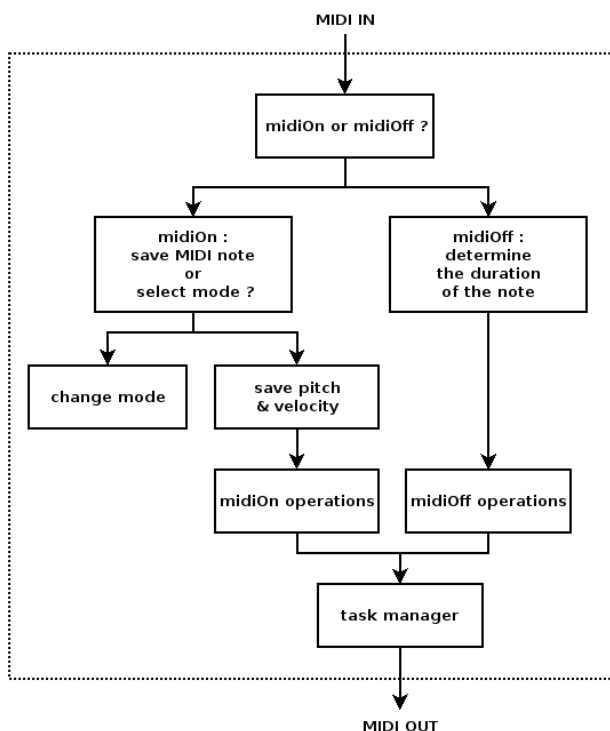


Figure 5. Schéma du fonctionnement de notre automate.

5.1. Exemple d’implémentation d’une opération

Pour traiter cet exemple d’implémentation, nous choisissons l’opération d’extension par ajout d’harmoniques. Cette opération est représentative de l’architecture de notre système. En effet, nous dédions un mode à cette opération (cf. la section 5.4). Nous l’exécutons dans ce mode lorsqu’une note est attaquée. Mais nous l’utilisons aussi dans notre dernier mode, où différentes opérations sont employées (cf. la section 5.7). Dans ce mode, nous l’exécutons au relâchement d’une note, si la durée de cette dernière est supérieure à une seconde et demi. Ces conditions d’exécution sont déterminées dans les blocs *midiOn operations* et *midiOff operations* de la figure 5.

Pour cette opération, le but de la tâche est de jouer une note de la série harmonique de la note jouée par le pianiste à chaque fois qu’elle est appelée. Par exemple, nous devons l’exécuter dix fois pour jouer les dix premières harmoniques de la note jouée par un pianiste. Une des propriétés des tâches JavaScript nous fournit le nombre de fois que la tâche a été appelée. Nous utilisons cette propriété pour déterminer le rang de l’harmonique à jouer (cf. lignes 4 et 5 de la figure 6). Nous avons

décidé que l’intensité des harmoniques diminuera suivant leur rang. Toutefois, l’amplitude de cette décroissance est corrélée à la vélocité de la note jouée par le pianiste (cf. ligne 6 de la figure 6). La tâche dispose d’un contrôle pour vérifier la vraisemblance des données à émettre (cf. ligne 8 de la figure 6). Si le déroulement de cette tâche aboutit à produire des données invalides, elle met fin à son exécution (cf. ligne 11 de la figure 6).

```

1 function overtones0pe() {
2   var note = arguments.callee.task.arguments[0][0];
3   var velocity = arguments.callee.task.arguments[0][1];
4   var coef = arguments.callee.task.iterations + 1;
5   var drift = freq2midi(midi2freq(note)*coef);
6   var amp = Math.round(velocity -
7     (arguments.callee.task.iterations*midi2data(velocity, 0.01, 3)));
8   if((drift<128)&&(amp>0)) {
9     outlet(0, [ drift, amp, midi2data(velocity, 1000, 150) ]);
10  } else {
11    arguments.callee.task.cancel ();
12  }
13 }
  
```

Figure 6. Tâche de l’opération d’extension par ajout d’harmoniques.

Nous gérons l’exécution de nos tâches avec un tableau. Nous l’initialisons pour qu’il contienne trois cellules, le nombre maximal de tâches pouvant fonctionner en parallèle. Nous l’utilisons comme une liste circulaire. Nous naviguons dans celui-ci avec un index que nous incrémentons de un à chaque appel d’une nouvelle tâche, modulo le nombre de cellules du tableau.

Avant d’affecter une nouvelle tâche à une cellule de notre tableau, nous nous assurons que la précédente est achevée. Si ce n’est pas le cas, nous l’interrompons (cf. ligne 2 de la figure 7, le test est réalisé dans la fonction *stopTask*). Nous affectons notre nouvelle tâche dans cette cellule pour qu’elle exécute l’opération d’extension par ajout d’harmoniques et nous la paramétrons avec les données MIDI fournies par le pianiste (cf. ligne 4 de la figure 7). Nous fixons une durée d’attente entre chaque itération de cette tâche. Toutefois, nous lions la valeur de cette durée à la vélocité de la note jouée par le pianiste (cf. ligne 5 de la figure 7). Enfin, nous exécutons et planifions quinze itérations de cette tâche (cf. ligne 6 de la figure 7). Cependant, l’exécution des quinze itérations de cette tâche sera menée à son terme seulement si les données qu’elle produit sont vraisemblables (cf. lignes 8 à 12 de la figure 6).

Nous pouvons exécuter une tâche JavaScript par différentes méthodes. Nous pouvons contrôler la vie d’une tâche par différents facteurs. Nous pouvons agir sur le fonctionnement des tâches en modifiant différentes propriétés. Cette variété d’usage favorise la flexibilité de notre système. Pour sensibiliser les réponses délivrées par notre automate, nous utilisons une mise à l’échelle classique entre les données MIDI fournies par le pianiste en temps réel et les paramètres de nos tâches. Dans les prochaines sections, nous

détaillons succinctement les différentes possibilités d'augmentation présentées au public.

```

1 function playOvertones(note, velocity, index) {
2   stopTask(taskRun[index]);
3
4   taskRun[index] = new Task(overtonesOpe, this, [note, velocity]);
5   taskRun[index].interval = midi2data(velocity, 1300, 25);
6   taskRun[index].repeat(15);
7 }

```

Figure 7. Gestion de la tâche de l'opération d'extension par ajout d'harmoniques.

5.2. Reflet

L'automate accompagne en miroir le pianiste. Le reflet est déterminé par le do central, point de symétrie de cette opération.



Figure 8. Opération de reflet.

5.3. Reflet propagé

L'automate accompagne en miroir le pianiste. Cependant, le reflet dérive d'octave en octave. La vitesse de la dérive est corrélée à l'intensité du jeu pianistique. Le reflet est déterminé par le do central, point de symétrie de cette opération.



Figure 9. Opération de reflet propagé.

5.4. Harmoniques

L'automate génère la série harmonique de la note jouée par le pianiste. La vitesse d'apparition des harmoniques est corrélée à l'intensité du jeu pianistique.



Figure 10. Opération d'extension par ajout d'harmoniques.

5.5. Répétition

L'automate répète la note jouée par le pianiste tout en ralentissant les itérations.



Figure 11. Opération de répétition.

5.6. Répétition transposée

L'automate répète une note plus grave que la note jouée par le pianiste tout en ralentissant les itérations. L'intervalle entre la note jouée et la note répétée est déterminé suivant l'intensité du jeu pianistique.



Figure 12. Opération de répétition transposée.

5.7. Opération multiple

Ce mode regroupe différentes opérations. Les échos sont exécutés si un intervalle particulier est joué et si le dispositif a enregistré plus de cinq notes.

Si l'intensité d'une note dépasse un certain seuil alors une répétition transposée est exécutée. Si la durée d'une note dépasse une seconde et demi alors l'ordinateur joue sa série harmonique. Si un intervalle d'octave est joué alors un double reflet (propagation dans les aigus et les graves) est exécuté. Si un intervalle montant de quarte est joué alors l'automate répète une section de notes jouée précédemment par le pianiste. L'écho est accéléré, rétrogradé et transposé symétriquement par rapport au do central.



Figure 13. Opération rétrograde et de symétrie d'un écho.

Si un intervalle de triton est joué alors l'automate répète une section de notes jouée précédemment par le pianiste. L'écho est accéléré, rétrogradé et transposé d'un demi-ton plus grave.



Figure 14. Opération rétrograde, d'accélération et de transposition d'un demi-ton plus grave d'un écho.

Si un intervalle descendant de quinte est joué alors l'automate répète une section de notes jouée précédemment par le pianiste. L'écho est juste accéléré.



Figure 15. Opération d'accélération d'un écho.

Si un intervalle de sixte mineure est joué alors l'automate répète une section de notes jouée précédemment par le pianiste. L'écho est accéléré et transposé symétriquement par rapport au do central.

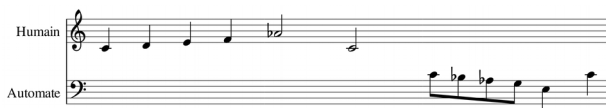


Figure 16. Opération d'accélération et de symétrie d'un écho.

6. CONCLUSION

L'objectif de notre projet de thèse est de réaliser une panoplie de captation du geste et de dispositifs de production de sons électroniques pour la musique vivante. L'étude et l'appropriation de l'augmentation instrumentale proposée par Jean-Claude Risset dans l'élaboration de ses duos pour un pianiste rentrent dans ce cadre. Suivant le même principe présenté dans cet article, nous avons programmé un arpégiateur en JavaScript dans Max/MSP. Nous avons développé aussi un certain nombre de synthétiseurs sous FAUST⁹. L'un des objectifs du langage de programmation FAUST est de pérenniser les traitements du signal et de synthèses sonores temps réel face à la volatilité des technologies et de les rendre ubiquitaires, exportables sur tous types de plates-formes, allant des extensions de différents formats à des applications pour téléphone portable.

Depuis l'introduction de la norme HTML5 et de la balise <audio>, il n'est plus nécessaire d'user d'extensions tiers pour gérer le son dans un navigateur web. De plus, la *Web Audio API* fournit aux développeurs des outils pour générer et transformer le

⁹ La librairie FAUST fournit plusieurs traitements et synthétiseurs opérationnels : <http://faust.grame.fr/library.html>.

son. En 2012, FAUST ajoute une brique à son projet pour traduire un code source DSP FAUST en fichier JavaScript. Nos développements d'augmentation instrumentale en JavaScript peuvent stimuler nos synthétiseurs FAUST dans Max/MSP, mais aussi dans un navigateur web¹⁰. Ces possibilités technologiques concourent à pérenniser notre travail. Nous pouvons le rendre accessible au plus grand nombre, sans qu'il soit restreint à l'usage d'un logiciel ou d'un système d'exploitation et cela pour des projets artistiques ou pédagogiques.

Nos prochains efforts se focaliseront sur la captation du geste afin d'irriguer de données nos différents modules d'augmentation et de synthèse. Nous validerons enfin nos différents développements par un travail artistique.

7. RÉFÉRENCES

- [1] Drummond, J. "Understanding interactive systems", Organised Sound 14/2, Cambridge University Press, 2009.
- [2] Engramelle, M.-D.-J. *Tonotechnie, ou l'art de noter les cylindres*, P. M. Delaguette, Paris, 1775.
- [3] Koetsier, T. "On the prehistory of programmable machines: musical automata, looms, calculators", *Mechanism and Machine Theory* 36, 2001.
- [4] Paradiso, J. A., O'Modhain, S. "Current Trends in Electronic Music Interfaces. Guest Editors' Introduction", *Journal of New Music Research* 32/4, 2003.
- [5] Patten, T. W. *Instruments for New Music, Sound, Technology, and Modernism*, University of California Press, Oakland, 2016.
- [6] Pottier, L. "Les musiques mixtes temps réel : pour une interprétation du son électronique en concert", colloque Soixante ans de musiques mixtes, Paris, 2012.
- [7] Risset, J.-C. "Composer le son : expériences avec l'ordinateur, 1964-1989", *Contrechamps n°11 – Musiques Électroniques, l'Age d'Homme*, 1990.
- [8] Risset, J.-C, Duyne, S. V. "Real-Time Performance Interaction with a Computer-Controlled Acoustic Piano", *Computer Music Journal* 20/1, 1996.
- [9] Roads, C., Mathews, M. "Interview with Max Mathews", *Computer Music Journal* 4/4, 1980.

¹⁰ Nous avons réalisé une transposition de cette installation pour fonctionner dans un navigateur web pour la communication de Pottier, L. « Jean-Claude Risset : autour de la synthèse sonore, de ses œuvres, de la façon de les (re)présenter graphiquement, interactivement, sur le WEB », *Rencontres internationales du Collegium Musicae, Jean-Claude Risset : interdisciplinarités*, Paris, 2018.