

Option Electronique et Signal pour la Musique : nano-projets

S. Reynal

Printemps 2021

Contents

1	Production demandée et évaluation	2
2	Sujets autour de la synthèse sonore ou de traitements audio	2
2.1	Synthétiseur de Karplus-Strong (dit "par modèle physique") .	3
2.2	Synthétiseur FM	3
2.3	Synthétiseur par méthode additive	3
2.4	Transfert de timbre par Deep Learning	4
2.5	Synthétiseur granulaire	4
2.6	Synthèse vocale : sons voisés par filtrage formantique	5
2.7	Algorithme de stretch et pitch-shift	5
2.8	Modification de la voix chantée par la méthode PSOLA (Pitch Synchronous OverLap and Add)	5
2.9	Application audio temps réel pour Android	5
2.10	Application audio temps réel sur STM32 : réverbération artificielle	6
2.11	On-line Audio Effect Generator (webaudio)	6
3	Sujets autour de la MIR	6
3.1	Convertisseur audio/MIDI (extraction de mélodie)	6
3.2	Classification en genres musicaux par Deep Learning	7
3.3	Séparation de sources d'un morceau de musique par la méthode de "NMF informée"	7
3.4	Séparation automatisée de la voix et de l'accompagnement	7
3.5	Visualisation automatique de la structure d'un morceau de musique	7
3.6	Algorithme Shazam	8
4	Sujets art et sciences : danse, musicologie, composition automatique, ...	8
4.1	Suivi musical de cible pour la danse avec Max/MSP	8

4.2	Musique et mathématiques	9
4.3	Composition automatique de musique avec WaveNet (production de signaux audio par DL)	9
4.4	Composition automatique de musique avec OpenMusic (production de partitions)	9
4.5	Systèmes de gammes en orient et en occident	9
5	Sujets expérimentaux autour de l'acoustique	10
5.1	Mesure et amélioration des caractéristiques acoustiques d'une salle - application à la chambre sourde et/ou à la cabine d'écoute	10
5.2	Mesure des caractéristiques réverbérantes d'une salle - application à la réverbération artificielle	10
5.3	Mesure du diagramme de rayonnement d'un instrument de musique	10
5.4	Techniques de prise de son stéréophonique	10
6	Sujets autour de l'électronique audio : conception de circuits	11
6.1	Conception d'un amplificateur classe A pour casque audio	11
6.2	Conception d'un préamplificateur pour microphone électrodynamique	11
6.3	Etude des systèmes de liaison filaire entre Console FOH et scène : sym, asym, boites de DI	11

1 Production demandée et évaluation

Il faut produire :

- un rapport écrit (étude théorique, éventuellement état de l'art, présentation critiques des résultats),
- une réalisation pratique : implémentation d'un algorithme etc, soit sous matlab, soit en programmation (Java ou C), soit sous PureData ou Max/MSP, ou bien une réalisation électronique le cas échéant.

2 Sujets autour de la synthèse sonore ou de traitements audio

Pour tous les sujets autour de la synthèse sonore, je conseille la lecture préalable des documents suivants sur la synthèse sonore [[HEZ2013](#)] et [[DAV2008](#)]. Je demande toujours une petite étude bibliographique (état de l'art, quelques articles, éventuellement un ou deux *très bons* sites web, donc pas wikipedia...), une étude théorique qui figurera dans le rapport, et l'implémentation de la

méthode, soit sous matlab, soit en programmation (Java ou C), soit sous PureData ou Max/MSP.

2.1 Synthétiseur de Karplus-Strong (dit "par modèle physique")

Il s'agit d'une méthode de synthèse sonore par modèle physique, c'est-à-dire utilisant les équations de la physique pour modéliser la production du son par l'instrument. La méthode KS utilise les ingrédients suivants : propagation dans des guides d'onde (simulés sous forme de ligne à retard z^{-n}), filtrage passe-bas et enfin, boucle de (contre-)réaction. C'est une méthode qui fonctionne très bien avec les instruments à corde pincée (guitare, clavecin), ainsi que les percussions.

On pourra étudier l'algorithme détaillé ici :

http://ccrma.stanford.edu/~jos/pasp/Karplus_Strong_Algorithm.html

et/ou suivre le sujet détaillé ici :

http://www-reynal.ensea.fr/docs/sigmus/projet_karplus.pdf

Travail à réaliser : réaliser une étude théorique de la synthèse KS, puis implémenter la méthode et proposer quelques exemples de synthèse pour la guitare ou pour des percussions.

2.2 Synthétiseur FM

Mettre en oeuvre la technique de synthèse sonore par modulation de fréquence, inventée en 1967 par John Chowning, et popularisée dans les 80's grâce au célèbre synthétiseur DX7 de Yamaha (Depeche Mode, The Cure, Tear for fears, etc.). Le principe fondamental consiste à générer un son riche en harmonique en modulant un signal sinusoïdal par un autre oscillateur sinusoïdal de fréquence différente. Il existe de nombreux raffinements : modulations multiples et en cascade, auto-modulation...

Biblio spécifique : l'article historique de J. Chowning [CHOW].

Travail à réaliser : réaliser une étude théorique de la synthèse FM (principe de la modulation FM, compréhension du spectre généré à l'aide des fonctions de Bessel, topologie des oscillateurs), implémenter la méthode et proposer quelques exemples de synthèse (sons naturels et sons artificiels).

2.3 Synthétiseur par méthode additive

Synthèse sonore d'un instrument par la technique additive utilisée dans de nombreux synthétiseurs des années 80. Le principe consiste à construire un signal musical complexe en additionnant plusieurs partiels à des fréquences et amplitudes judicieusement choisies, puis en modulant les partiels en amplitude (enveloppe ADSR). Par exemple, si l'on se restreint à une enveloppe

”decay”, on aurait une construction du type

$$x(t) = \sum_i a_i \sin \omega_i t e^{-t/\tau_i}$$

où τ_i est le temps de ”decay” associé au i ème partiel, et qui rend compte de l’absorption différente par les matériaux des aigus et des graves. Les fréquences ω_i ne sont pas nécessairement dans un rapport harmonique.

On pourra suivre le sujet détaillé suivant : http://www-reynal.ensea.fr/docs/sigmus/projet_synthese_additive.pdf

Travail à réaliser : Implémenter un algorithme de synthèse additive. Il doit être facile d’éditer les paramètres. On pourra choisir un instrument, analyser ses partiels et affiner les paramètres de synthèse en fonction de cette analyse.

2.4 Transfert de timbre par Deep Learning

Magenta DDSF est une bibliothèque de fonctions de traitement du signal audio (synthétiseurs, filtres, non-linéarités, ...) adaptées à l’apprentissage machine par Deep Learning. Les différents modules DDSF peuvent être utilisés pour générer et manipuler du signal audio à partir d’un réseau de neurone. Une des application est le transfert de timbre, qui transforme par exemple une voix (enregistrée) en un son de violon à partir d’un modèle pré-entraîné.

Lien vers le site du projet : <https://github.com/magenta/ddsp>

2.5 Synthétiseur granulaire

La synthèse granulaire consiste à combiner des ”grains” de signaux existants (souvent enregistrés) pour produire un nouveau signal complexe. Les grains sont des échantillons sonores de l’ordre de la milliseconde (10 à 100 ms) dont on contrôle globalement la densité, la hauteur, la longueur, l’enveloppe, etc. le plus souvent sous la forme d’intervalles dans lesquels le programme de synthèse choisit des valeurs de manière aléatoire. Le son obtenu est une sorte de nuage, composé de l’ensemble des grains.

Doc initiale ici : <http://www.soundonsound.com/techniques/granular-synthesis>

Site de ressource sur la SG : <http://www.granularsynthesis.com>

Un exemple de SG appliquée à la voix : <https://www.youtube.com/watch?v=puXWMZ6bUCA>

Travail à réaliser : Réaliser l’implémentation d’un synthétiseur granulaire, soit sous Matlab, sous PureData (ou son équivalent MAX/MSP), soit en Java.

2.6 Synthèse vocale : sons voisés par filtrage formantique

Ce projet a pour but de réaliser un synthétiseur de voyelles en Matlab. Le synthétiseur s'appuie sur la modélisation source-filtre de la production vocale. La source est générée par un synthétiseur paramétrique et le filtre est une concaténation de filtres passe-bande paramétriques. Après une étape de recherche bibliographique sur les thèmes de la synthèse source-filtre et de la phonétique, il est proposé de réaliser une étude comparative des différents modèles programmés et un dictionnaire de voyelles.

Le sujet détaillé est disponible ici : http://www-reynal.ensea.fr/docs/sigmus/projet_synthe_formantique.pdf

2.7 Algorithme de stretch et pitch-shift

Il s'agit de développer un algorithme permettant de modifier un fichier audio, soit en durée (stretch : titre 'étiré' sans modification de la hauteur tonale), soit en hauteur tonale (pitch-shift : titre plus aigu ou plus grave, sans modification de la durée) ; i s'agit ici, modestement, d'exploiter le code des algorithmes existants et d'être capable, à la fin du projet, d'expliquer rigoureusement comment il fonctionne.

Je vous propose d'utiliser la technique du [vocoder](#) de phase. Plusieurs exemples de code existent déjà, dont par exemple une implémentation en python : <https://github.com/haoyu987/phasevocoder>.

2.8 Modification de la voix chantée par la méthode PSOLA (Pitch Synchronous OverLap and Add)

C'est une technique de traitement du signal numérique utilisée pour le traitement de la parole. Cette technique permet, entre autres, d'effectuer des modifications de la prosodie (évolution de la fréquence instantanée du signal) et de la durée d'un signal de parole. Le but du projet est de réaliser une implémentation Matlab de l'algorithme PSOLA. Le programme sera capable de transposer et de rallonger/raccourcir un signal de parole. La programmation est guidée et certaines fonctions sont fournies.

Le sujet ainsi que les documents nécessaires sont disponibles ici : http://www-reynal.ensea.fr/docs/sigmus/projet_PSOLA.pdf et http://www-reynal.ensea.fr/docs/sigmus/projet_PSOLA.zip.

2.9 Application audio temps réel pour Android

The goal of this project is to implement a real-time audio signal processing application on iOS and/or Android. You will follow the usual steps of such a project:

- Preliminary step: choice of an audio effect and bibliography.

- S1. Prototyping the audio processing algorithm with Matlab or Python.
- S2. Platform-agnostic implementation of the audio processing module in C++, testing in a simple non-real-time command-line-interface (CLI) program.
- Integration of the module in iOS and/or Android applications.

The main objective here is to work on real-time audio processing algorithms in C++. Therefore, other parts of the code (command line or graphical interface, plugging to the system's audio render loop etc.) is provided. However, it is possible to modify everything at will.

Sujet détaillé ici : <https://gitlab.com/AudioScientist/real-time-audio-processing-on-m>

2.10 Application audio temps réel sur STM32 : réverbération artificielle

Votre carte Discovery STM32F746 contient tout ce qu'il faut pour réaliser du traitement du signal audio en temps réel : une paire de microphones, une entrée et une sortie au format jack, et un CODEC audio. Ce sujet vous propose d'implémenter un effet de réverbération audio par convolution.

Voir une introduction à la réverbération par convolution : <http://www-reynal.ensea.fr/docs/sigmus/reverb-convol.pdf>

2.11 On-line Audio Effect Generator (webaudio)

Il s'agit de concevoir un outil graphique (en ligne) pour la construction d'une chaîne d'effets audio pour un site web dédié à la musique ; l'outil permet de déposer des blocs d'effet (compresseur, filtre, reverb) sur une page web vierge pour construire le code javascript correspondant.

- voir [l'initiation](#) à l'API WebAudio
- voir le [site](#) de la WebAudio API (tutoriels, doc, exemples)
- voir la [documentation](#) sur l'API "Canvas" permettant de dessiner (en javascript) dans une page web (utile — mais pas indispensable — pour dessiner l'interface graphique)
- [tutoriel](#) pour s'initier à l'API "Canvas"

3 Sujets autour de la MIR

3.1 Convertisseur audio/MIDI (extraction de mélodie)

Il s'agit d'étudier un signal audio en utilisant l'une de ces représentations, le chromagramme, puis d'implémenter un algorithme permettant d'extraire

la mélodie d'un morceau et de la convertir en codes MIDI. Le sujet détaillé question par question est disponible ici :

http://www-reynal.ensea.fr/teaching/sigmus/tp/tp_chroma.pdf

On pourra commencer par lire les notes de cours correspondantes : [HPAP2013]

3.2 Classification en genres musicaux par Deep Learning

Le but de ce projet est de réaliser une classification en genre musical à partir du signal audio. La classification sera réalisée à l'aide d'un réseau de neurones profonds constitué de couches convolutives. L'apprentissage et l'évaluation de ce réseau de neurones seront réalisés sur la base de données Free Music Archive. Le système fera une classification monolabel (un seul genre prédit par morceau) sur 8 classes de genres.

Voir le sujet détaillé ici : http://www-reynal.ensea.fr/docs/sigmus/Projet_ENSEA_classification_en_genres_musicaux.pdf

3.3 Séparation de sources d'un morceau de musique par la méthode de "NMF informée"

Il s'agit de séparer les différentes sources qui composent un signal de musique dans le cas où l'on dispose uniquement d'un signal monophonique. C'est un problème qui a de nombreuses applications telles que la transcription automatique des morceaux de musique, la reconnaissance d'instruments, ou l'estimation de la mélodie.

Sujet détaillé : http://www-reynal.ensea.fr/docs/sigmus/projet_separation_sources_nmf.pdf et les divers scripts matlab pour démarrer http://www-reynal.ensea.fr/docs/sigmus/projet_separation_sources_nmf.zip

3.4 Séparation automatisée de la voix et de l'accompagnement

La séparation de la voix chantée est une étape essentielle pour plusieurs applications telles que l'identification de chanteur, la transcription de la mélodie, la recherche par fredonnement.

Sujet détaillé : http://www-reynal.ensea.fr/docs/sigmus/projet_separation_voix.pdf et divers documents matlab http://www-reynal.ensea.fr/docs/sigmus/projet_separation_voix.zip

3.5 Visualisation automatique de la structure d'un morceau de musique

Les morceaux de musique sont structurés à différentes échelles de temps (battements, temps, accords, mesure, phrase, section etc.). L'estimation de la structure consiste à décrire l'organisation des éléments qui constituent un

morceau de musique à un niveau macro-temporel. Ce problème a de nombreuses applications, dont l'indexation dans de grandes bases de données, la classification des musiques par genre ou la génération de résumés audio, et a attiré l'attention de nombreux chercheurs ces dernières années.

Sujet détaillé : http://www-reynal.ensea.fr/docs/sigmus/projet_structure.pdf et les documents matlab utiles pour démarrer http://www-reynal.ensea.fr/docs/sigmus/projet_structure.zip

3.6 Algorithme Shazam

L'objectif est de reproduire l'algorithme d'identification audio par technique de fingerprint de Shazam tel que publié [ici](#) ; il faudra construire :

- une base de morceaux faites de ses morceaux personnel (votre collection musicale)
- une fonction A permettant l'extraction des fingerprints d'un titre donné et le stockage de ceux-ci dans un fichier ;
- une fonction B de batch appelant la fonction A sur tout les morceaux de la base, et créant un fichier unique stockant tous les triplets [fingerprint, morceau d'origine, temps d'origine] ;
- une fonction C de reconnaissance audio, celle-ci enregistrera le signal audio du micro (fonction matlab audiorecorder), appellera la fonction A pour obtenir les fingerprints et appellera la fonction D de reconnaissance ;
- une fonction D qui recherchera le meilleur appariement entre fingerprint d'entrée et ceux stockés dans la base.

4 Sujets art et sciences : danse, musicologie, composition automatique, ...

4.1 Suivi musical de cible pour la danse avec Max/MSP

Max/MSP est un langage graphique dédié au traitement audio pour la musique (similaire à puredata) ; ici il s'agira de concevoir un "patch" sous Max/MSP permettant, à partir d'une captation caméra (webcam ou autre), de modifier un paramètre sonore en temps réel en fonction des déplacement d'un danseur (ou toute autre cible mobile lors d'un spectacle ou d'une performance) ; on utilisera la bibliothèque JITTER/video intégrée dans Max/MSP (disponible sur le Mac situé dans la petite salle au fond de la 261) [Max/MSP/puredata]

- [démystifier](#) Max/MSP !
- une [video](#) de demo de jitter

4.2 Musique et mathématiques

On propose d'analyser au travers de quelques articles de mathématiciens/musicologues, les apports de la théorie des groupes dans : i) l'analyse de la construction d'oeuvres en musique baroque ou classique ; ii) la composition en musique contemporaine chez Boulez.

Travail à réaliser : une étude bibliographique (choisir un ou deux articles au maximum, et approfondir leur étude), puis implémenter quelques exemples illustrant une ou l'autre méthode exposée (sous Matlab, PureData, MAX/MSP ou en Java selon votre goût). Les articles sont disponibles sur le site de l'option, cf bibliographie attenante : [APAP101], [ANDR], [FISH], [APAP112], [TAQ] et [WEI].

4.3 Composition automatique de musique avec WaveNet (production de signaux audio par DL)

WaveNet est un modèle génératif basé sur du Deep Learning, permettant de générer de l'audio échantillon par échantillon. WaveNet permet notamment de générer de la parole, mais aussi de synthétiser des signaux audio musicaux.

Ce sujet vous proposer de vous initier au fonctionnement de WaveNet. Article de référence : <https://arxiv.org/pdf/1609.03499.pdf>

4.4 Composition automatique de musique avec OpenMusic (production de partitions)

Il s'agit d'exploiter l'environnement OpenMusic, un logiciel de programmation graphique (similaire à puredata) dédié à la composition musicale, qui permet de générer des morceaux de musique à partir de règles de programmation (transpositions, figures rythmiques, etc.).

Prise en main du logiciel OpenMusic : <http://repmus.ircam.fr/openmusic/home>

4.5 Systèmes de gammes en orient et en occident

Réaliser un programme sous matlab ou PureData/MaxMSP permettant de construire et de jouer (à l'aide d'une interface utilisateur simple) différents systèmes de gammes (Bach, Pythagore, dodécaphonique, arabe, Werkmeister, gammes indo-pakistanaïses, ...).

Bibliographie : s'inspirer du sujet de TP "gammes" disponible sur le site http://www-reynal.ensea.fr/docs/sigmus/tp_gammes.pdf

Vous y trouverez également un Notebook Mathematica d'introduction : http://www-reynal.ensea.fr/docs/sigmus/mathematica_gammes.nb

5 Sujets expérimentaux autour de l'acoustique

5.1 Mesure et amélioration des caractéristiques acoustiques d'une salle - application à la chambre sourde et/ou à la cabine d'écoute

Il s'agit de s'initier à l'acoustique des salles, et notamment à ce qui "fait une bonne cabine d'écoute". Il faudra appréhender la notion de modes de résonance, et ce qui influence leur position spectrale et leur coefficient de qualité. Dans un premier temps, on construira un dispositif expérimental de mesure des modes acoustiques d'une salle. Puis on cherchera le meilleur positionnement des éléments acoustique d'amortissement pour éviter les modes de résonance et améliorer la réponse de la salle.

5.2 Mesure des caractéristiques réverbérantes d'une salle - application à la réverbération artificielle

Il s'agit ici aussi de s'initier à l'acoustique des salles, mais plutôt sous l'angle de la spatialisation. Il faudra comprendre la notion de modes de résonance, et en quoi ils influencent la réponse temporelle de la salle (la "réverbération"). Dans un premier temps, on construira un dispositif expérimental de mesure de la réponse impulsionnelle d'une salle (il existe plusieurs techniques utilisant la déconvolution). Puis on concevra un algorithme permettant d'émuler cette réverbération sur machine et on comparera avec la réponse réelle de la salle (amphi, salle de TD, tout peut faire l'affaire).

5.3 Mesure du diagramme de rayonnement d'un instrument de musique

Il s'agit de réaliser un dispositif expérimental capable de mesurer le diagramme de rayonnement d'un instrument (guitare, clarinette,...) dans la salle semi-anéchoïde et de comparer les résultats obtenu avec un modèle simplifié de diagramme de directivité.

5.4 Techniques de prise de son stéréophonique

Réaliser une étude bibliographique de la prise de son stéréophonique (couples de microphone, positionnement, techniques particulières liées à tel ou tel instrument), puis mettre cette technique en oeuvre sur un instrument au choix (chambre sourde, ou salle du piano).

Matériel : microphones électrostatiques LEM, carte son, câbles XLR-XLR, pieds, acquisition sous matlab ou Audacity.

WARNING : ce sujet ne peut être choisi que par un seul binôme (en raison du matériel disponible en un seul exemplaire !).

Bibliographie : "la prise de son stéréophonique" (disponible au centre de documentation).

6 Sujets autour de l'électronique audio : conception de circuits

6.1 Conception d'un amplificateur classe A pour casque audio

Il s'agit d'un sujet pour fans d'électronique analogique, consistant à réaliser une baie d'amplification pour casque à 4 voix. Cette baie sera branchée sur la console du studio de la chambre sourde. Il s'agit de choisir un circuit intégré approprié pour ce type d'application en amplification classe A (exemple : TDA2030), de concevoir le circuit (prévoir une EQ grave/aigu et un potentiomètre de niveau, et la possibilité de passer en mono ou stereo), et finalement de concevoir puis de réaliser le circuit (breadboard ou wrapping).

6.2 Conception d'un préamplificateur pour microphone électrodynamique

Il s'agit d'un sujet pour fans d'électronique analogique, consistant à réaliser un préamplificateur (AOp ou transistors ou CL...) pour microphone électrodynamique (non polarisé). Il s'agira de réfléchir à un cahier des charges simples (quel critères ? SNR ? consommation ?), de choisir le schéma le plus approprié pour ce type d'application, et finalement de concevoir puis de réaliser le circuit (breadboard ou wrapping).

6.3 Etude des systèmes de liaison filaire entre Console FOH et scène : sym, asym, boites de DI

Il s'agit de mettre en oeuvre et de mesurer les performances de plusieurs types de liaison filaire entre la console de mixage et la scène lors d'un live : liaison asymétrique, liaison symétrique, boîte de D.I., ... Quelles sont leurs performances respectives ? Leur SNR ? etc. L'accent est vraiment sur la mesure.

On pourra ensuite réaliser un petit circuit symétriseur/désymétriseur à ampli-op, à titre d'illustration.

References

- [HEZ2013] T. Hezard, Transparents de cours sur les différentes techniques de synthèse sonore : http://www-reynal.ensea.fr/docs/sigmus/AMEA_2012-2013_poly_Synthese.pdf

- [CHOW] John M. Chowning, The Synthesis of complex audio spectra by means of frequency modulation. http://www-reynal.ensea.fr/docs/sigmus/projet_Article_Chowning_Synthese_FM.pdf
- [HPAP2013] H. Papadopoulos, Transparents de cours sur les représentations temps-fréquences adaptées aux signaux musicaux (chromagramme, etc.). http://www-reynal.ensea.fr/docs/sigmus/lecture_pitch_chroma_amea_2013_6.pdf
- [DAV2008] B. David, Polycopié de cours d'introduction à la modélisation des instruments et à la synthèse sonore. http://www-reynal.ensea.fr/docs/sigmus/acmus_ensea02.pdf
- [APAP101] A. Papadopoulos, Mathématique et musique chez J.S. Bach, http://www-reynal.ensea.fr/docs/sigmus/math_100_101_Papadopoulos.pdf.
- [ANDR] M. Andreatta, Quelques aspects théoriques d'une approche algébrique en musique, http://www-reynal.ensea.fr/docs/sigmus/math_112_Andreatta.pdf.
- [FISH] M. Fisher, Leonhard Euler et la musique, http://www-reynal.ensea.fr/docs/sigmus/math_112_Fischer.pdf.
- [APAP112] A. Papadopoulos, Consonance musicale et complexité mathématique, http://www-reynal.ensea.fr/docs/sigmus/math_112_Papadopoulos.pdf.
- [TAQ] R. Taquard, Quelques exemples d'utilisation des mathématiques dans la théorie et la composition musicales, http://www-reynal.ensea.fr/docs/sigmus/math_112_Tacquard.pdf.
- [WEI] N. Weiss, Quelques propriétés de la technique de Boulez de multiplication des blocs sonores, http://www-reynal.ensea.fr/docs/sigmus/math_114_Weiss.pdf.