Institut National des Langues et Civilisations Orientales

Département Textes, Informatique, Multilinguisme

Titre du mémoire

MASTER

TRAITEMENT AUTOMATIQUE DES LANGUES

Parcours :

Ingénierie Multilingue

par

Martin DIGARD

Directeur de mémoire :

Damien NOUVEL

Encadrant :

Florent JACQUEMARD

Année universitaire 2020-2021

Table des matières

Liste des figures

Liste des tableaux

INTRODUCTION GÉNÉRALE

Ce mémoire de recherche, effectué en parallèle d'un stage à l'Inria dans le cadre du master de traitement automatique des langues de l'Inalco, contient une proposition originale ainsi que diverses contributions ayant toutes pour objectif d'améliorer **qparse**, un outil de transcription automatique de la musique sur sa capacité à transcrire la batterie. Nous ne parlerons donc pas directement de langues naturelles, mais de l'écriture automatique de partitions de musique à partir de données audios ou symboliques. Cette exercice nécessitera la manipulation d'un langage musical codifié avec une grammaire (solfège, durées, nuances, volumes) et soulèvera des problématiques concernées par les techniques du traitement automatique des langues.
L'écriture musicale offre de nombreuses possibilités pour la transcription d'un rythme donné. Le contexte musical ainsi que la lisibilité d'une partition pour un batteur entraîné conditionnent les choix d'écritures. Reconnaître la métrique principale d'un rythme, la façon de regrouper les notes par les ligatures, ou simplement décider d'un usage pour une durée parmi les différentes continuations possibles (notes pointées, liaisons, silences, etc.) constituent autant de possibilités que de difficultés.

Voici la proposition de ce mémoire ainsi que les contributions apportées lors du stage :

- Proposition principale : les systèmes (3.4) :
  Recherche de rythmes génériques en amont dans la chaîne de traitement.
  ⇒ L'objectif de fixer des choix le plus tôt possible afin de simplifier le reste des calculs en éliminant une partie d'entre eux. Ces choix concernent notamment la métrique et les règles de réécriture.
- Une description de la notation de la batterie (3.1)
- Une modélisation de la transcription de la batterie (3.2)
- Analyse MIDI-Audio (4.2)
- Théorie et tests unitaires pour le passage au polyphonique (4.4)
- Création de grammaires pondérées pour la batterie (4.4)
- Contributions sur la branche « distance » dans :
  - qparselib/notes/cluster.md
  - qparselib/src/segment/import/ :
    DrumCode hpp et cpp

Nous présenterons le contexte suivi d'un état de l'art et nous définirons de manière générale le processus de transcription automatique de la musique pour enfin étayer les méthodes utilisées pour la transcription automatique de la batterie, et nous présenterons les principales contributions apportées à l'outil qparse. Nous décrirons ensuite le corpus ainsi que les différentes expérimentations menées. Nous conclurons par une discussion sur les résultats obtenus et les pistes d'améliorations futures à explorer.

CHAPITRE 1

CONTEXTE

Sommaire

Introduction

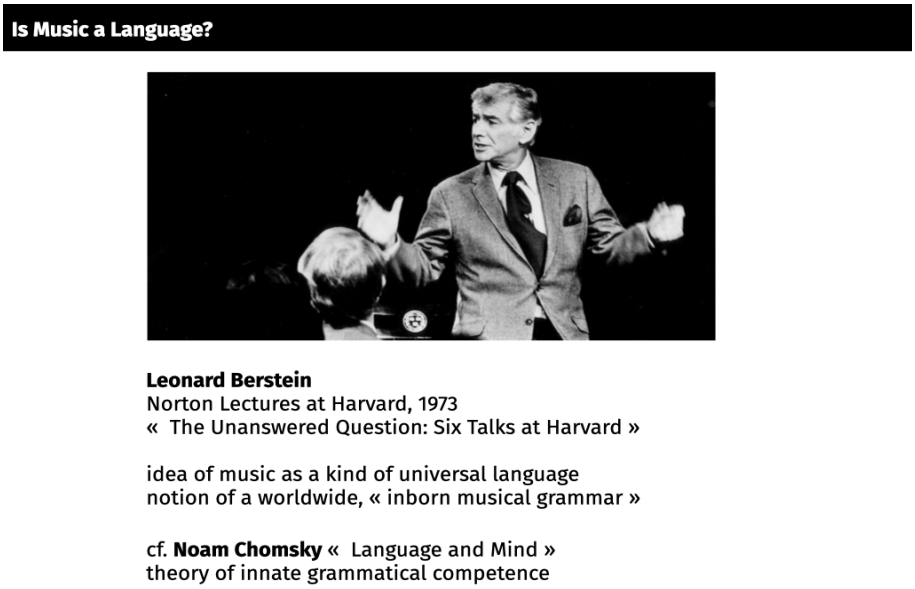
La transcription automatique de la musique (AMT) est un défi ancien [1] et difficile qui n'est toujours pas résolu. Il a engendré une pluie de sous-tâches qui ont donné naissance au domaine de la recherche d'information musicale (MIR). Actuellement, de nombreux travaux de MIR font appel au traitement automatique des langues (TAL) [1].
Dans ce chapitre, nous parlerons de l'informatique musicale, nous tenterons d'établir les liens existants entre le MIR et le TAL ainsi qu'entre les notions de langage musical et langue naturelle. Nous traiterons également de l'utilité et du problème de l'AMT et de la transcription automatique de la batterie (ADT).
Enfin, nous décrirons les représentations de la musique qui sont nécessaires à la compréhension du présent travail.

1.1 TAL et MIR

L'informatique musicale [2] est une étude du traitement de la musique [2], en particulier des représentations musicales, de la transformée de

1. NLP4MuSA, the 2nd Workshop on Natural Language Processing for Music and Spoken Audio, co-located with ISMIR 2021.
2. https://en.wikipedia.org/wiki/Music_informatics

Fourier pour la musique[3], de l'analyse de la structure de la musique et de la reconnaissance des accords. D'autres sujets de recherche en informatique musicale comprennent la modélisation informatique de la musique, l'analyse informatique de la musique, la reconnaissance optique de la musique, les éditeurs audio numériques, les moteurs de recherche de musique en ligne, la recherche d'informations musicales et les questions cognitives dans la musique.
Le MIR[4][5] apparaît vers le début des années 2000 [3]. C'est une science interdisciplinaire qui fait appel à de nombreux domaines comme la musicologie, l'analyse musicale, la psychologie, les sciences de l'information, le traitement du signal et les méthodes d'apprentissage automatisé en informatique. Cette discipline récente a notamment été soutenue par de grandes compagnies du web qui veulent développer des systèmes de recommandation de musique ou des moteurs de recherche dédiés au son et à la musique.



Aborder la musique à travers le TAL nécessite une réflexion autour de la musique en tant que langage ainsi que la possibilité de comparer ce même langage avec les langues naturelles. Quelques travaux en neurosciences ont abordé la question, notamment par observation des

3. https://interstices.info/de-fourier-a-la-reconnaissance-musicale/
4. https ://ismir.net/
5. https ://ismir2021.ismir.net/

processus cognitifs et neuronaux que les systèmes de traitement de ces deux langages avaient en commun. Dans le travail de Poulin-Charronnat *et al.* [4], la musique est reconnue comme étant un système complexe spécifique à l'être humain dont une des similitudes avec les langues naturelles est l'émergence de régularités reconnues implicitement par le système cognitif. La question de la pertinence de l'analogie entre langues naturelles et langage musical a également été soulevée à l'occasion de projets de recherche en TAL. Keller *et al.* [5] ont exploré le potentiel de ces techniques à travers les plongements de mots et le mécanisme d'attention pour la modélisation de données musicales. La question du sens d'une phrase musicale apparaît, selon eux, à la fois comme une limite et un défi majeur pour l'étude de cette analogie.

Ici, Digression sur la musicologie calculatoire vs linguistique computationnelle ?

D'autres travaux très récents, ont aussi été révélés lors de la *première conférence sur le NLP pour la musique et l'audio (NLP4MusA 2020)*. Lors de cette conférence, Jiang *et al.* [6] ont présenté leur implémentation d'un modèle de langage musical auto-attentif visant à améliorer le mécanisme d'attention par élément, déjà très largement utilisé dans les modèles de séquence modernes pour le texte et la musique.
Il semblerait que le domaine du TAL qui se rapproche le plus du MIR soit la reconnaissance de la parole. En effet, la séparation des sources ont des approches similaires dans les deux domaines. De plus, il existe un lien entre partition musicale comme manière d'écrire la musique et texte comme manière d'écrire la parole.
Similitudes :
Reconnaissance automatique de la parole :
signal $\Rightarrow$ phonèmes $\Rightarrow$ texte Transcription automatique de la musique :
signal $\Rightarrow$ MIDI $\Rightarrow$ partition Différence :
Texte (données linéaires) $\neq$ partition (données structurées hiérarchiques)

1.2 La transcription automatique de la musique

En musique, la transcription[6] est la pratique consistant à noter un morceau ou un son qui n'était auparavant pas noté et/ou pas populaire en tant que musique écrite, par exemple, une improvisation de jazz ou une bande sonore de jeu vidéo. Lorsqu'un musicien est chargé de créer une partition à partir d'un enregistrement et qu'il écrit les notes qui composent le morceau en notation musicale, on dit qu'il a créé une transcription musicale de cet enregistrement.

6. https://en.wikipedia.org/wiki/Transcription_(music)

L'objectif de la transcription automatique de la musique (AMT) [7] est de convertir la performance d'un musicien en notation musicale - un peu comme la conversion de la parole en texte dans le traitement du langage naturel. L'AMT a des intérêt multiples, notamment pour la transcription de solos ou encore pour la constitution de corpus musicologiques, ou encore pour l'interprétation de la musique et l'analyse du contenu musical [8]. Par exemple, un grand nombre de fichiers audio et vidéo musicaux sont disponibles sur le Web, et pour la plupart d'entre eux, il est difficile de trouver les partitions musicales correspondantes, qui sont nécessaires pour pratiquer la musique, faire des reprises et effectuer une analyse musicale détaillée. Les partitions de musique classique sont facilement accessibles et il y a peu de demandes de nouvelles transcriptions. D'un point de vue pratique, des demandes beaucoup plus commerciales et académiques sont attendues dans le domaine de la musique populaire [8]. Les modèles grammaticaux qui représentent la structure hiérarchique des séquences d'accords se sont avérés très utiles dans les analyses récentes de l'harmonie du jazz [9]. Comme déjà évoqué précédemment, il s'agit d'un problème ancien et difficile. C'est un « graal » de l'informatique musicale. En 1976, H. C. Longuet-Higgins [1] évoquait déjà la représentation musicale en arbre syntaxique dans le but d'écrire automatiquement des partitions à partir de données audio en se basant sur un mimétisme psychologique de l'approche humaine. De même pour les chercheurs en audio James A. Moorer, Martin Piszczalski et Bernard Galler qui, en 1977[7], ont utilisé leurs connaissances en ingénierie de l'audio et du numérique pour programmer un ordinateur afin de lui faire analyser un enregistrement musical numérique de manière à détecter les lignes mélodiques, les accords et les accents rythmiques des instruments à percussion.
La tâche de transcription automatique de la musique comprend deux activités distinctes : l'analyse d'un morceau de musique et l'impression d'une partition à partir de cette analyse.

La figure 1.1 est une proposition de Benetos *et al.* [7] qui représente l'architecture générale d'un système de transcription musicale. On y observe plusieurs sous-tâches de l'AMT :

- La séparation des sources à partir de l'audio.
- Le système de transcription :
  - Cœur du système :
    ⇒ Algorithmes de détection des multi-pitchs et de suivi des notes.
    Quatres sous-tâches optionnelles accompagnent ces algorithmes :

7. https://en.wikipedia.org/wiki/Transcription_(music)

  - identification de l'instrument ;
  - estimation de la tonalité et de l'accord ;
  - détection de l'apparition et du décalage ;
  - estimation du tempo et du rythme.
- Apprentissage sur des modèles accoustiques et musicologiques.
- *Optionnel :* Informations fournies de manière externe, soit fournie en amont (genre, instruments,...), soit par interaction avec un utilisateur (infos sur une partition incomplète).

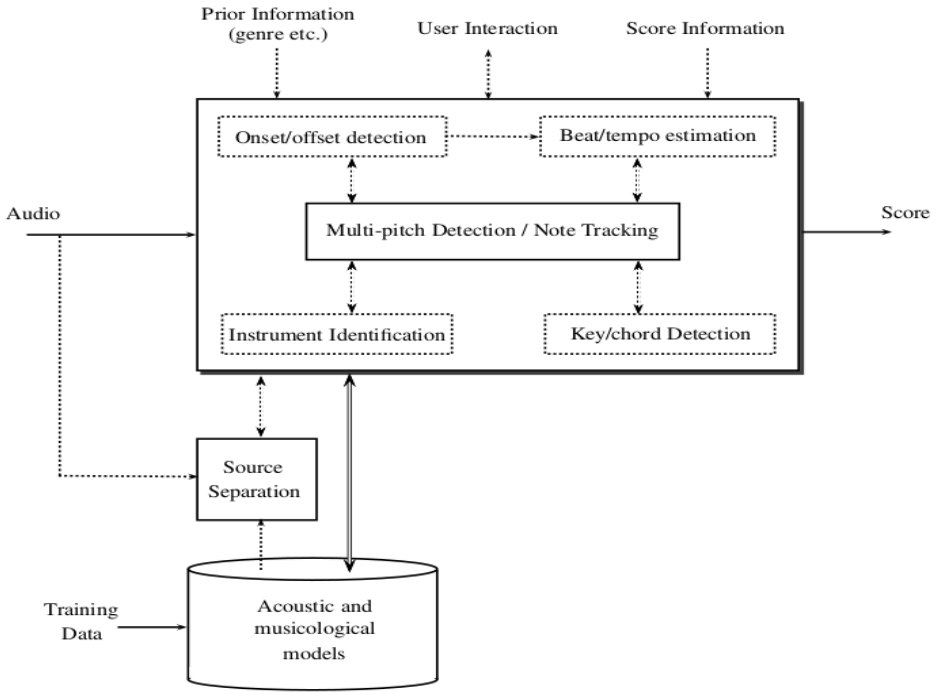


FIGURE 1.1 – Transcription automatique
Les sous-systèmes et algorithmes optionnels sont présentés à l'aide de lignes pointillées. Les doubles flèches mettent en évidence les connexions entre les systèmes qui incluent la fusion d'informations et une communication plus interactive entre les systèmes.

1.3 La transcription automatique de la batterie

La batterie est un instrument récent qui s'est longtemps passé de partition. En effet pour un batteur, la qualité de lecteur lorsqu'elle était nécessaire, résidait essentiellement dans sa capacité à lire les partitions des autres instrumentistes (par exemple, les grilles d'accords et la mélodie du thème en jazz) afin d'improviser un accompagnement approprié que personne ne pouvait écrire pour lui à sa place.
Les partitions de batterie sont arrivées par nécessité avec la pédagogie et l'émergence d'écoles de batterie partout dans le monde. Un autre facteur qui a contribué à l'expansion des partitions de batterie est l'émergence de la musique assistée par ordinateur (MAO). En effet, l'usage de boîtes à rythmes ou de séquenceurs permettant d'expérimenter soi-même l'écriture de rythmes en les écoutant mixés avec d'autres instruments sur des machines a permis aux compositeurs de s'émanciper de la création d'un batteur en lui fournissant une partition contenant les parties exactes qu'ils voulaient entendre sur leur musique.
La batterie a un statut à part dans l'univers de l'AMT puisqu'il s'agit d'instruments sans hauteur (du point de vue harmonique), d'événements sonores auxquels une durée est rarement attribuée et de notations spécifiques (symboles des têtes de notes).
Les applications de l'ADT seraient utiles dans tous les domaines musicaux contenant de la batterie dont certains manque de partitions, notamment les musiques d'improvisation (jazz, pop) [7].
Mais aussi de manière plus générale dans le domaine du MIR. Si les ordinateurs étaient capables d'analyser la partie de la batterie dans la musique enregistrée, cela permettrait une variété de tâches de traitement de la musique liées au rythme. En particulier, la détection et la classification des événements sonores de la batterie par des méthodes informatiques est considérée comme un problème de recherche important et stimulant dans le domaine plus large de la recherche d'informations musicales [10].
L'ADT est un sujet de recherche crucial pour la compréhension des aspects rythmiques de la musique, et a un impact potentiel sur des domaines plus larges tels que l'éducation musicale et la production musicale.

1.4 Les représentations de la musique

Les données audio

Le fichier WAV [8] est une instance du Resource Interchange File Format (RIFF) défini par IBM et Microsoft. Le format RIFF agit comme une "enveloppe" pour divers formats de codage audio. Bien qu'un fichier WAV

8. https ://en.wikipedia.org/wiki/WAV

puisse contenir de l'audio compressé, le format audio WAV le plus courant est l'audio non compressé au format LPCM (linear pulse-code modulation). Le LPCM est également le format de codage audio standard des CD audio, qui stockent des données audio LPCM à deux canaux échantillonnées à 44 100 Hz avec 16 bits par échantillon. Comme le LPCM n'est pas compressé et conserve tous les échantillons d'une piste audio, les utilisateurs professionnels ou les experts en audio peuvent utiliser le format WAV avec l'audio LPCM pour obtenir une qualité audio maximale.

Les données MIDI

Le MIDI[9] (Musical Instrument Digital Interface) est une norme technique qui décrit un protocole de communication, une interface numérique et des connecteurs électriques permettant de connecter une grande variété d'instruments de musique électroniques, d'ordinateurs et d'appareils audio connexes pour jouer, éditer et enregistrer de la musique.

Les données midi sont représentées sous forme de piano-roll. Chaque points sur la figure 1.2 est appelé « évènement MIDI » :

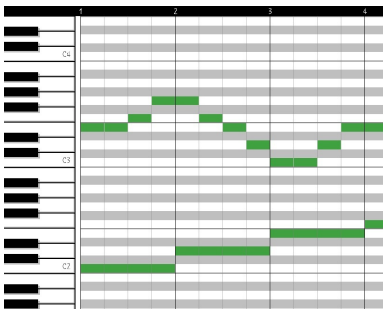


FIGURE 1.2 – Exemple évènements avec durée

Chaque évènement MIDI rassemble un ensemble d'informations sur la hauteur, la durée, le volume, etc. . . :

9. https://en.wikipedia.org/wiki/MIDI

Protocol | Event

Property	Value
Type	Note On/Off Event
On Tick	15812
Off Tick	15905
Duration	93
Note	45
Velocity	89
Channel	9

FIGURE 1.3 – Critère pour un évènement

Pour la batterie, les évènements sont considérés sans durée, nous ignorerons donc les offsets (« Off Event »), les « Off Tick » et les « Duration ». Le *channel* ne nous sera pas utile non plus.
Ici, définir Tick et channel.

Voici un exemple de piano-roll midi pour la batterie :

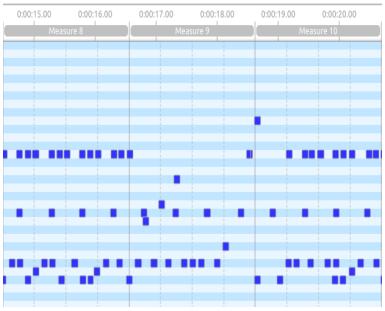


FIGURE 1.4 – Exemple évènements sans durée

On observe que toutes les durées sont identiques.

Les partitions

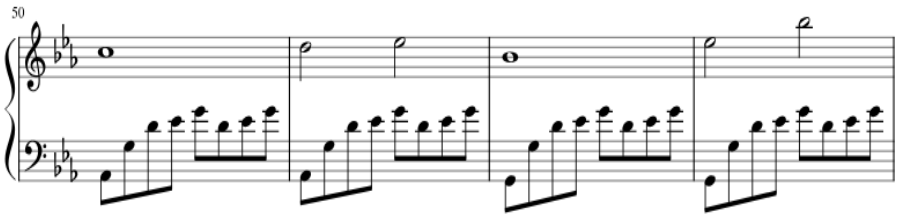


FIGURE 1.5 – Exemple de partition de piano

Une partition de musique [10] est un document qui porte la représentation systématique du langage musical sous forme écrite. Cette représentation est appelée transcription et elle sert à traduire les quatre caractéristiques du son musical :

— la hauteur ;
— la durée ;
— l'intensité ;
— le timbre.

Ainsi que de leurs combinaisons appelées à former l'ossature de l'œuvre musicale dans son déroulement temporel, à la fois :

— diachronique (succession des instants, ce qui constitue en musique la mélodie) ;
— et synchronique (simultanéité des sons, c'est-à-dire l'harmonie).

1.4.1 Le format MusicXML

MusicXML est un format de fichier basé sur XML pour représenter la notation musicale occidentale. Ce format est ouvert, entièrement documenté et peut être utilisé librement dans le cadre de l'accord de spécification finale de la communauté du W3C.
Un des avantages de ce format est qu'il peut être converti aussi bien en données MIDI qu'en partition musicale, ce qui en fait une interface homme/machine.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE score-partwise PUBLIC
    "-//Recordare//DTD MusicXML 3.1 Partwise//EN"
    "http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise version="3.1">
  <part-list>
    <score-part id="P1">
      <part-name>Music</part-name>
    </score-part>
  </part-list>
  <part id="P1">
    <measure number="1">
      <attributes>
        <divisions>1</divisions>
        <key>
          <fifths>0</fifths>
        </key>
        <time>
          <beats>4</beats>
          <beat-type>4</beat-type>
        </time>
        <clef>
          <sign>G</sign>
          <line>2</line>
        </clef>
      </attributes>
      <note>
        <pitch>
          <step>C</step>
          <octave>4</octave>
        </pitch>
        <duration>4</duration>
        <type>whole</type>
      </note>
    </measure>
  </part>
</score-partwise>
```

FIGURE 1.6 – MusicXML

10. https://fr.wikipedia.org/wiki/Partition_(musique)

Le figure 1.6 représente un do en clef de sol de la durée d'une ronde sur une mesure en 4/4.

Conclusion

Dans ce chapitre, nous avons établi que le MIR s'intéresse de plus en plus au TAL, et que, par ce biais, il y a des liens possibles entre le langage musical et les langues naturelles, le plus proche étant probablement le phénomène d'écriture des sons de l'un comme de l'autre.
Nous avons également établi que le MIR est né de l'AMT qui est un problème ancien et très difficile et qu'il serait toujours très utile de le résoudre (autant pour l'AMT que pour l'ADT).
Et enfin, nous avons décrit les représentations de la musique nécessaires à la compréhension du présent mémoire, allant du son jusqu'à l'écriture.

CHAPITRE 2

ÉTAT DE L'ART

Sommaire

Introduction

Dans ce chapitre, nous observerons les différentes avancées qui ont déjà eu lieu dans le domaine de la transcription automatique de la musique et de la batterie afin de situer notre démarche.
Nous aborderons le passage crucial du monophonique au polyphonique dans la transcription. Nous ferons un point sur les deux grandes parties de l'AMT de bout en bout : de l'audio vers le MIDI puis des données MIDI vers l'écriture d'une partition. Ensuite, nous discuterons des approches linéaires et des approches hiérarchiques.

2.1 Monophonique et polyphonique

Les premiers travaux ont été faits sur l'identification des instruments monophoniques [1] [7]. Actuellement, le problème de l'estimation automatique de la hauteur des signaux monophoniques peut être considéré comme résolu, mais dans la plupart des contextes musicaux, les instruments sont polyphoniques. L'estimation des hauteurs multiples (détection multi-pitchs ou F0 multiples) est le problème central de la création d'un système de transcription de musique polyphonique. Il s'agit de la détection de notes qui peuvent apparaître simultanément et être produites par

1. Instruments produisant une note à la fois, ou plusieurs notes de même durée (monophonie par accord).

plusieurs instruments différents. Ce défi est donc majeur pour la batterie puisque c'est un instrument qui est lui-même constitué de plusieurs instruments (caisse-claire, grosse-caisse, cymbales, toms, etc...). Le fort degré de chevauchement entre les durées ainsi qu'entre les fréquences complique l'identification des instruments polyphoniques. Cette tâche est étroitement liée à la séparation des sources et concerne aussi la séparation des voix. Les performances des systèmes actuels ne sont pas encore suffisantes pour permettre la création d'un système automatisé capable de transcrire de la musique polyphonique sans restrictions sur le degré de polyphonie ou le type d'instrument. Cette question reste donc encore ouverte.

2.2 Audio vers MIDI

Jusqu'à aujourd'hui, les recherches se sont majoritairement concentrées sur le traitement du signal vers la génération du MIDI [11]. Cette partie englobe plusieurs sous-tâches dont la détection multi-pitchs, la détection des onset et des offset, l'estimation du tempo, la quantification du rythme, la classification des genres musicaux, etc...
En ADT [10], plusieurs stratégies de répartition pré/post-processing sont possibles pour la détection multi-pitchs. Entamer la détection dès le pré-processing, en supprimant les features non-pertinentes pendant la séparation des sources afin d'obtenir une meilleure détection des instruments de la batterie, est une démarche intuitive : supprimer la structure harmonique pour atténuer l'influence des instruments à hauteurs sur la détection grosse-caisse et caisse-claire en est un exemple. Mais certaines études montrent que des expériences similaires ont donné des résultats non-concluants et que la suppression des instruments à hauteurs peut avoir des effets néfastes sur les performances de l'ADT. En outre, les systèmes d'ADT basés sur des RNN ou des NMF font la séparation des sources pendant l'optimisation, ce qui réduit la nécessité de la faire pendant le pré-processing.
Pour la reconnaissance des instruments, une approche possible [12] est de mettre un modèle probabiliste dans l'étape de la classification des évènements afin de classer les différents sons de la batterie. Cette méthode permet de se passer de samples audio isolés en modélisant la progression temporelle des features avec un HMM. Les features sont transformés en représentations statistiques indépendantes. L'approche AdaMa [13] est une autre approche de la même catégorie ; elle commence par une estimation initiale des sons de la batterie qui sont itérativement raffinés pour correspondre à (pour matcher) l'enregistrement visé.

2.3 MIDI vers partition

Le plus souvent, lorsque les articles abordent la transcription automatique de bout en bout (de l'audio à la partition), l'appellation « score » (*partition*) désigne un ouput au format Music XML, ou simplement MIDI. Par exemple, dans [8], la chaîne de traitement va jusqu'à la génération d'une séquence MIDI quantifiée qui est importée dans MuseScore pour en extraire manuellement un fichier MusicXML contenant plusieurs voix.
Seuls quelques travaux récents s'intéressent de près à la création d'outils permettant la génération de partition. Le problème de la conversion d'une séquence d'évènements musicaux symboliques en une partition musicale structurée est traité notamment dans [14]. Ce travail, qui vise à résoudre en une fois la quantification du rythme et la production de partition, s'appuie tout au long du processus sur des grammaires génératives qui fournissent un modèle hiérarchique *a priori* des partitions. Les expériences ont des résultats prometteurs, mais il faut relever qu'elle ont été menées avec un ensemble de données composé d'extraits monophoniques ; il reste donc à traiter le passage au polyphonique en couplant le problème de la séparation des voix avec la quantification du rythme.
L'approche de [14] est fondée sur la conviction que la complexité de la structure musicale dépasse les modèles linéaires.

2.4 Approche linéaire et approche hiérarchique

Plusieurs travaux ont d'abord privilégié l'approche stochastique. Par exemple, Shibata *et al.* [8] ont utilisé le modèle de Markov caché (HMM) [2] pour la reconnaissance de la métrique. Les auteurs utilisent d'abord deux réseaux de neurones profonds, l'un pour la reconnaissance des pitchs et l'autre pour la reconnaissance de la vélocité. Pour la dernière couche, la probabilité est obtenue par une fonction sigmoïde. Ils construisent ensuite plusieurs HMM métriques étendus pour la musique polyphonique correspondant à des métriques possibles, puis ils calculent la probalitité maximale pour chaque modèle afin d'obtenir la métrique la plus probable.

2. https://fr.wikipedia.org/wiki/Modèle_de_Markov_caché
https://en.wikipedia.org/wiki/Hidden_Markov_model

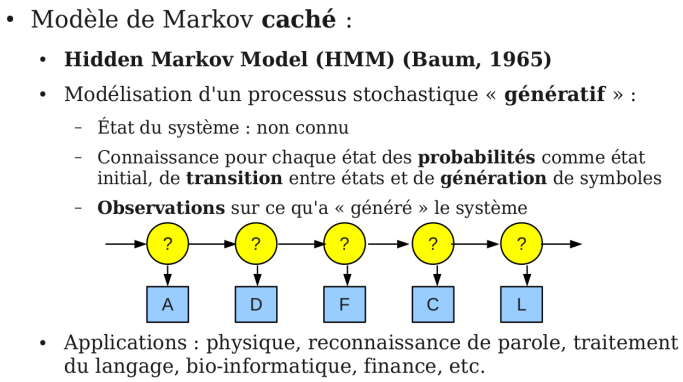


FIGURE 2.1 – HMM

Source : Cours de Damien Nouvel [3]

L'évaluation finale des résultats de [8] montre qu'il faut rediriger l'attention vers les valeurs des notes, la séparation des voix et d'autres éléments délicats de la partition musicale qui sont significatifs pour l'exécution de la musique. Or, même si la quantification du rythme se fait le plus souvent par la manipulation de données linéaires allant notamment des *real time units* (secondes) vers les musical *time units* (temps, métrique,...), de nombreux travaux suggèrent d'utiliser une approche hiérarchique puisque le langage musical est lui-même structuré.
En effet, l'usage d'arbres syntaxiques est idéale pour représenter le langage musical. Une méthodologie simple pour la description et l'affichage des structures musicales est présentée dans [15]. Les RT y sont évoqués comme permettant une cohésion complète de la notation musicale traditionnelle avec des notations plus complexes. Jacquemard *et al.* [16] propose aussi une représentation formelle du rythme, inspirée de modèles théoriques antérieurs et dont l'objectif est la réécriture de termes. Ils démontrent aussi l'application des arbres de rythmes pour les équivalences rythmiques dans [17]. La réécriture d'arbres, dans un contexte de composition assistée par ordinateur, par exemple, pourrait permettre de suggérer à un utilisateur diverses notations possibles pour une valeur rythmique, avec des complexités différentes.
La nécessité d'une approche hiérarchique pour la production automatique de partition est évoquée dans [14]. Les modèles de grammaire qui y sont exposés sont différents de modèles markoviens linéaires de précédents travaux.

3. https://damien.nouvels.net/fr/enseignement

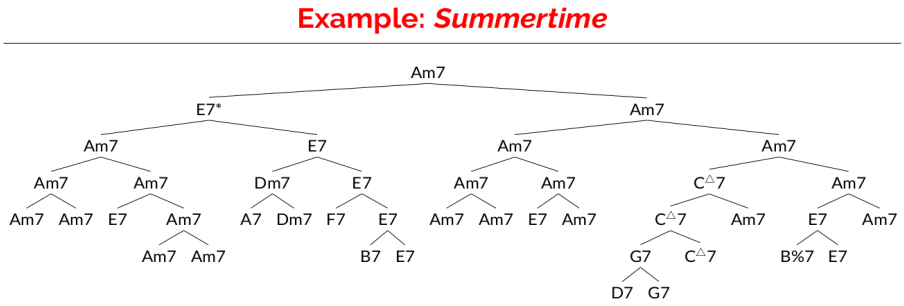


FIGURE 2.2 – arbre_jazz
Représentation arborescente d'une grille harmonique [9]

Conclusion

La plupart des travaux déjà existants sur l'ADT ont été énumérés par Wu *et al.* [10] qui, pour mieux comprendre la pratique des systèmes d'ADT, se concentrent sur les méthodes basées sur la factorisation matricielle non négative et celles utilisant des réseaux neuronaux récurrents. La majorité de ces recherches se concentre sur des méthodes de calcul pour la détection d'événements sonores de batterie à partir de signaux acoustiques ou sur la séparation entre les évènements sonores de batterie avec ceux des autres instruments dans un orchestre ou un groupe de musique [18], ainsi que sur l'extraction de caractéristiques de bas niveau telles que la classe d'instrument et le moment de l'apparition du son. Très peu d'entre eux ont abordé la tâche de générer des partitions de batterie et, même quand le sujet est abordé, l'output final n'est souvent qu'un fichier MIDI ou MusicXML et non une partition écrite.
Il n'existe pas de formalisation de la notation de la batterie ni de réelle génération de partition finale, dont les enjeux principaux seraient :
1) le passage du monophonique au polyphonique, comprenant la distinction entre les sons simultanés et les flas ou autres ornements ;
2) les choix d'écritures spécifiques à la batterie concernant la séparation des voix et les continuations.

CHAPITRE 3

MÉTHODES

Sommaire

Introduction

Dans ce chapitre, nous expliquerons en détail les méthodes que nous avons employées pour l'ADT.
Pour commencer, nous exposerons une description de la notation de la batterie ainsi qu'une modélisation de celle-ci pour la représentation des données rythmiques en arbres syntaxiques. Nous poursuiverons avec une présentation de qparse [1], un outil de transcription qui est développé par Florent Jacquemard (Inria) au sein du laboratoire Cedric au CNAM.
Enfin, nous présenterons les systèmes.

3.1 La notation de la batterie

Une figure de note [19] de musique combine plusieurs critères [2] :

— Une tête de note :
Sa position sur la portée indique la hauteur de la note. La tête de note peut aussi indiquer une durée.

1. https ://qparse.gitlabpages.inria.fr/
2. https://fr.wikipedia.org/wiki/Note_de_musique

— Une hampe :
Indicatrice d'appartenance à une voix en fonction de sa direction et indicatrice d'une durée représentée par sa présence ou non (blanche $\neq$ ronde)
— Un crochet : La durée d'une note est divisée par deux à chaque crochet ajouté à la hampe d'une figure de note.

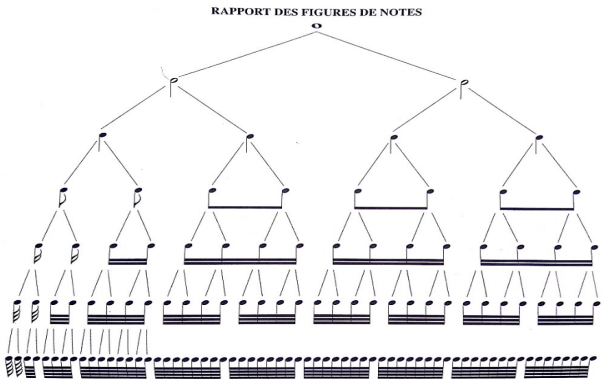


FIGURE 3.1 – Rapport des figures de notes
[19]

La figure 3.1 montre les rapports de durée entre les figures de notes. Plus les durées sont longues, plus elles sont marquées par la tête de note (la note carrée fait deux fois la durée d'une ronde) ou la présence ou non de la hampe. À partir de la noire (3ème lignes en partant du haut), on ajoute un crochet à la hampe d'une figure de notes pour diviser sa durée par 2. Les notes à crochet (croche , double-croche, triple...) peuvent être reliées ou non par des ligatures (Voir les 4 dernière lignes de la figure 3.1).

Les hauteurs et les têtes de notes

Pour la transcription, nous proposons une notation inspirée du recueil de pièces pour batterie de J.-F. Juskowiak [20] et des méthodes de batterie Agostini [21], car nous trouvons la position des éléments cohérente et intuitive.
En effet, les hauteurs sur la portée représentent :

— La hauteur physique des instruments :
La caisse claire est centrale sur la portée et sur la batterie (au niveau de la ceinture, elle conditionne l'écart entre les pédales et aussi la position de tous les instruments basiques d'une batterie).
Tout ce qui en-dessous de la caisse-claire sur la portée est en dessous de la caisse-claire sur la batterie (pédales, tom basse) ;
Tout ce qui est au-dessus de la caisse-claire sur la portée, l'est

aussi sur la batterie.

— La hauteur des instruments en terme de fréquences :
Sauf pour le charley au pied et si l'on sépare en trois groupes (grosse-caisse, toms et cymbales), de bas en haut, les instruments vont du plus grave au plus aigu.

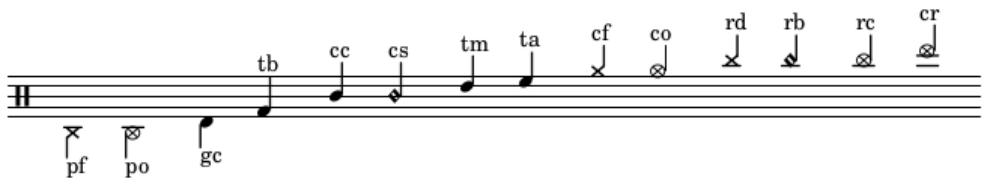


FIGURE 3.2 – Hauteur et têtes de notes

Les noms des instruments correspondant aux codes des notes de la figure 3.2 sont dans le tableau 3.1.

Les durées

Comme nous venons de la voir, la majorité des instruments de la batterie sont représentés par les têtes des notes. Par conséquent, les symboles rythmiques concernant la tête de note ne pourront pas être utilisés. Cela est valable aussi pour la présence ou non de la hampe puisque ce phénomène n'existe qu'avec les têtes de notes de type cercle-vide (opposition blanche-ronde). L'usage des blanches existe dans certaines partitions de batterie [22] mais cela reste dans des cas très rares. Certains logiciels permettent de faire des blanches avec des symboles spécifiques à la batterie ou aux percussions mais leur lecture reste peu aisée et leur utilisation pour la batterie est rarissime.
La durée d'une note peut être allongée par divers symboles :

— Le point ;
— La liaison.

Ces symboles ne seront utiles que pour l'écriture des ouvertures de charley. Le charley est le seul instrument de la batterie dont la durée est quantitifiée (les cymbales attrapées à la main peuvent l'être aussi mais cela est très rare.)

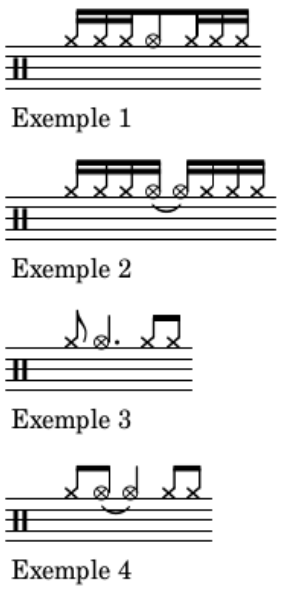


FIGURE 3.3 – Point et liaison

L'écriture de la batterie doit faire ressortir la pulsation. La première chose à prendre en compte pour analyser la figure 3.3 est donc la nécessité de regrouper les notes par temps à l'aide des ligatures.
Exemple 1 : ouverture de charley quantifiée mais pas notes pas regroupées par temps.
Exemple 2 : bieeen !
Exemple 3 et exemple 4 : les deux exemples sont valables mais le deuxième est le plus souvent utilisé car plus intuitif (regroupement par temps).
En cas de nécessité de rallonger la durée d'une note pour la batterie, on privilégiera la liaison.

Les silences

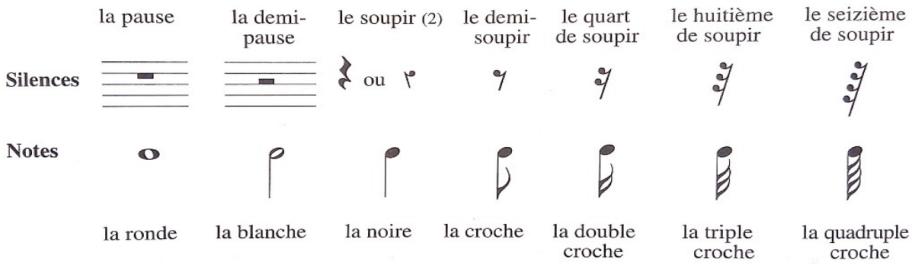


FIGURE 3.4 – Les silences

Les silences sont parfois utilisés pour quantifier les ouvertures de charley. Les fermetures du charley sont notées soit par un silence (correspondant à une fermeture de la pédale), soit par un écrasement de l'ouverture par un autre coup de charley fermé, au pied ou à la main. Physiquement, le charley est fermé par une pression du pied sur la pédale de charley. Dans les fichiers MIDI, cette pression est traduite par un charley joué au pied. Mais dans une vraie partition, cette écriture ne traduirait pas ce que le batteur doit penser.

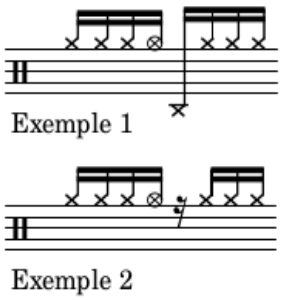


FIGURE 3.5 – Silence joué

L'exemple 1 de la figure 3.5 montre ce qui est écrit dans les données MIDI et l'exemple 2 montre ce que le batteur doit penser en lisant la partition. Il faut aussi prendre en compte l'écriture surchargée que l'exemple 1 donnerait avec une partition comprenant plusieurs voix et plusieurs instruments jouant simultanément.

Les équivalences rythmiques

Pour les instruments mélodiques, la liaison et le point sont les deux seules possibilités en cas d'équivalence rythmique pour des notes dont la durée de l'une à l'autre est ininterrompue. Mais pour la batterie, à part pour les ouvertures de charley (voir section 3.1), les durées des notes n'ont pas d'importance. L'usage des silences pour combler la distance rythmique entre deux notes devient donc possible.
Cela pris en compte, et étant donné que les indications de durée dans les têtes de notes sont peu recommandées (voir section 3.1), l'écriture à l'aide de silences sera privilégiée comme indication de durée sauf dans les cas où cela reste impossible. Ce choix à pour but de n'avoir qu'une manière d'écrire toutes les notes, que leurs têtes de notes soit modifiées ou non.
Sur la figure 3.6, théoriquement, il faudra choisir la notation de la deuxième mesure mais dans certains contextes, pour des raisons de lisibilité ou de surcharge, la version sans les silences de la troisième mesure pourra être choisie.

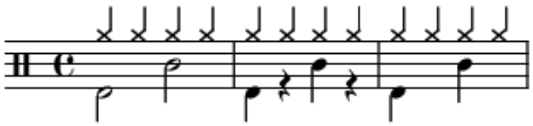


FIGURE 3.6 – Équivalence

Les voix

Les voix[3] désignent les différentes parties mélodiques constituant une composition musicale et destinées à être interprétées, simultanément ou successivement, par un ou plusieurs musiciens. En batterie, une voix est l'ensemble des instruments qui, à eux seuls, constituent une phrase rythmique et sont regroupés à l'aide des ligatures. Plusieurs écritures étant possibles pour un même rythme, on peut regrouper les instruments de la batterie par voix. Sur une portée de batterie, il existe le plus souvent 1 ou 2 voix. Sur la figure 3.7, il faudra faire un choix entre les exemples 1, 2 et 3 qui sont trois façons d'écrire le même rythme.

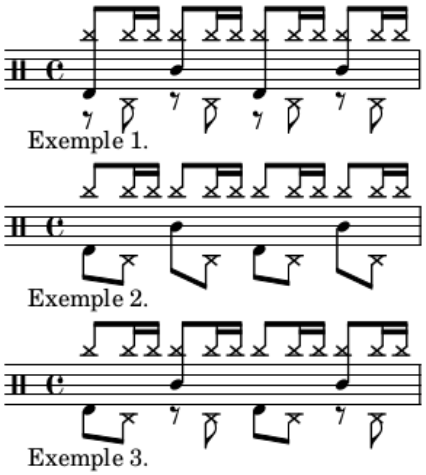


FIGURE 3.7 – Séparation des voix

Ce choix se fera en fonction des instruments joués, de la nature plus ou moins systèmatique de leurs phrasés, et des associations logiques entre les instruments dans la distribution des rythmes sur la batterie (voir la section 3.4).

3. https://fr.wikipedia.org/wiki/Voix_(polyphonie)

Les accentuations et les ghost-notes

« Certaines notes dans une phrase musicale doivent, ainsi que les différentes syllabes d'un mot, être accentuées avec plus ou moins de force, porter une inflexion particulière. » [19]

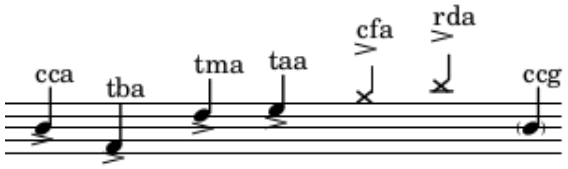


FIGURE 3.8 – Les accents et les ghost-notes

La figure 3.8 ne prend en compte que les accents que nous avons estimés nécessaires (voir la section 3.2). Les accents sont marqués par le symbole « > ». Il est positionné au-dessus des notes représentant des cymbales et en-dessous des notes représentant des toms ou la caisse-claire. Ce choix a été fait pour la partition de la figure 4.2 car elle est plus lisible ainsi, mais ces choix devront être adaptés en fonction des différents systèmes reconnus (voir la section 3.4). Par exemple, pour les systèmes jazz, les ligatures pour les toms et la caisse-claire seront dirigés vers le bas, il faudra donc mettre les symboles d'accentuation correspondants au-dessus des têtes de notes.
La dernière note de la figure 3.8 montre un exemple de ghost-notes. Le parenthésage a été choisi car il peut être utilisé sur n'importe quelle note sans changer la tête de note.
Pour les codes, on prend le code de la note et on ajoute un « a » pour un accent et un « g » pour une ghost-note. Toutes les notes de la figure 3.8 sont exposées en situation réelle dans la figure 3.9.

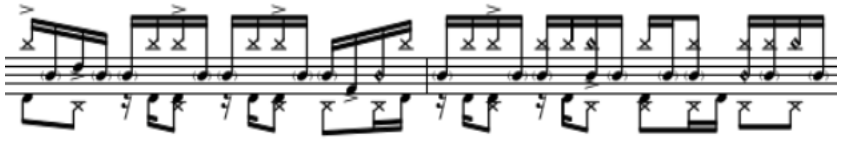


FIGURE 3.9 – Exemple pour les accentuations et les ghost-notes

3.2 Modélisation pour la transcription

Les pitchs

Codes	Instruments	Pitchs
cf	charley-main-fermé	22, 42
co	charley-main-ouvert	26
pf	charley-pied-fermé	44
rd	ride	51
rb	ride-cloche (bell)	53
rc	ride-crash	59
cr	crash	55
cc	caisse-claire	38, 40
cs	cross-stick	37
ta	tom-alto	48, 50
tm	tom-medium	45, 47
tb	tom-basse	43, 58
gc	grosse-caisse	36

TABLE 3.1 – Pitchs et instruments

Il existe, pour de nombreux instruments de la batterie, plusieurs samples audio associés à des pitchs. Pour cette première version, nous avons choisi de n'avoir qu'un code-instrument pour différentes variantes d'un instrument, c'est pourquoi certain code-instrument se voit attribuer plusieurs pitchs dans le tableau 3.1.
Malgré le large panel de pitchs disponible, il semblerait qu'aucun pitch ne désigne le charley ouvert joué au pied. Pourtant, dans la batterie moderne, plusieurs rythmes ne peuvent fournir le son du charley ouvert qu'avec le pied car les mains ne sont pas disponibles pour le jouer. Cela doit en partie être dû à l'utilisation des boîte à rythmes en MAO qui ne nécessitent pas de faire des choix conditionnés par les limitations humaines (2 pieds, 2 mains, et beaucoup plus d'instruments...)

La vélocité

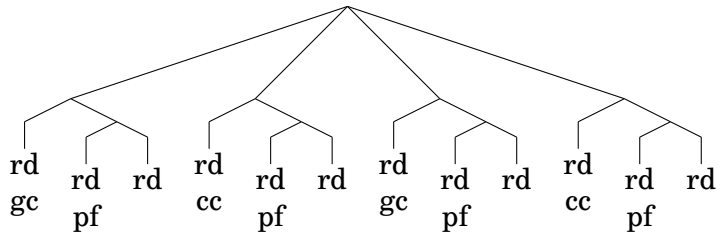
La partition de la figure 4.2 a été transcrite manuellement avec lilypond par analyse des fichiers MIDI et audio correspondants.
Cette transcription nous a mené aux observations suivantes :

- Vélocité inférieure à 40 : ghost-note ;
- Vélocité supérieure à 90 : accent ;
- Pas d'intention d'accent ni de ghost-note pour une vélocité entre 40 et 89 ;

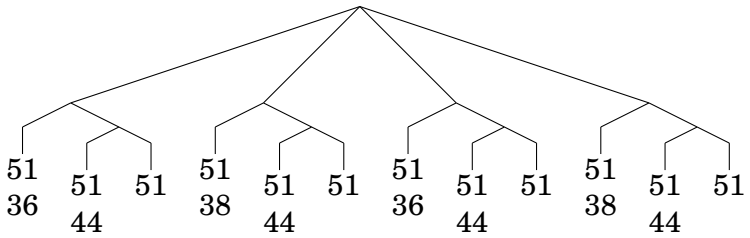
— Les accents et les ghosts-notes ne sont significatifs ni pour les instruments joués au pied, ni pour les cymbales crash.
En effet, certaines vélocités en dessous de 40 étant détectées et inscrites dans les données MIDI sont dues au mouvement du talon du batteur qui bat la pulsation sans particulièrement jouer le charley. Ce mouvement est perçu par le capteur de la batterie électronique mais le charley n'est pas joué.
— Au final, nous avons relevé les ghost-notes et les accents pour la caisse-claire ainsi que les accents pour les toms et les cymbales rythmiques (charley et ride).

Les arbres de rythmes

Les arbres de rythmes représentent un rythme unique dont les possibilités de notation sur une partition sont théoriquement multiples. Voici une représentation de la figure 3.7 en arbre de rythmes avec les codes de chaque instrument :



Ci-dessous, le même arbre dont les codes des instruments sont remplacés par leurs données MIDI respectives :



Chacun des trois exemples de la figure 3.7 est représenté par un des deux arbres syntaxiques ci-dessus.

3.3 Qparse

La librairie Qparse [4] implémente la quantification des rythmes basée sur des algorithmes d'analyse syntaxique pour les automates arborescents pondérés. En prenant en entrée une performance musicale symbolique (séquence de notes avec dates et durées en temps réel, typiquement un fichier MIDI), et une grammaire hors-contexte pondérée décrivant un langage de rythmes préférés, il produit une partition musicale. Plusieurs formats de sortie sont possibles, dont XML MEI. Les principaux contributeurs sont :

— Florent Jacquemard (Inria) : développeur principal.
— Francesco Foscarin (PhD, CNAM) : construction de grammaire automatique à partir de corpus ; Evaluation.
— Clement Poncelet (Salzburg U.) : integration de la librairie Midifile pour les input MIDI.
— Philippe Rigaux (CNAM) : production de partition au format MEI et de modèle intermédiaire de partition en sortie.
— Masahiko Sakai (Nagoya U.) : mesure de la distance input/output pour la quantification et CMake framework ; évaluation.

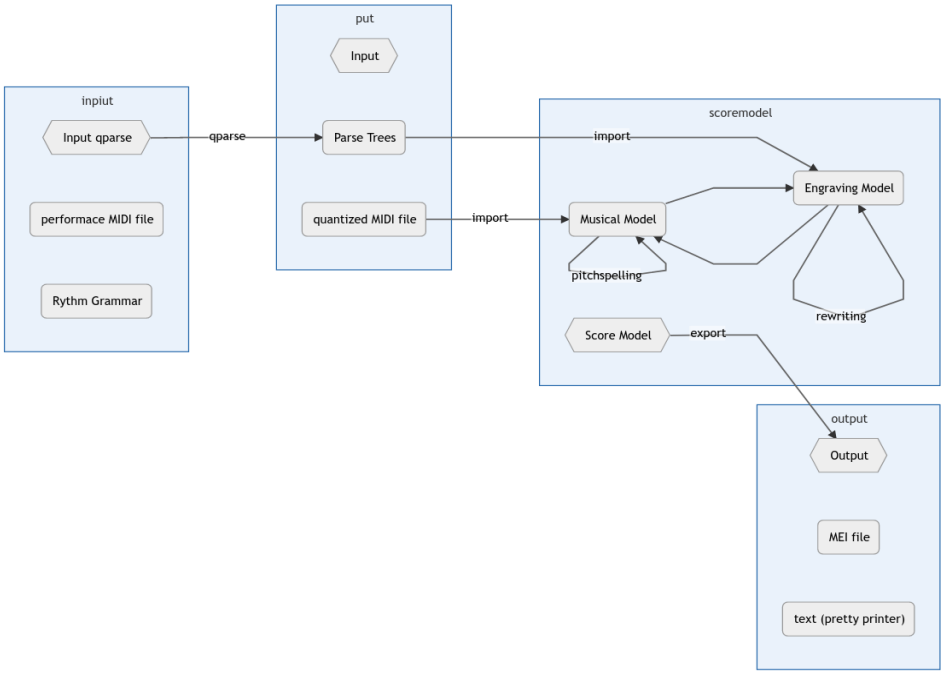


FIGURE 3.10 – Présentation de Qparse

4. https://qparse.gitlabpages.inria.fr

Explication des différentes étapes de la figure 3.10 [5] :

- **Input Qparse** :
  Un fichier MIDI (séquence d'événements datés (piano roll) accompagné d'un fichier contenant une grammaire pondérée) ;
- **Arbre de parsing** :
  Les données MIDI sont quantifiées, les notes de dates proches sont alignées et les relations entre les notes sont identifiées (accords, fla, etc...) ; un arbre de parsing global est créé ;
- **Score Model** :
  - Les instruments sont identifiés dans scoremodel/import/tableImporterDrum.cpp ;
  - Réécriture 1 :
    séparation des voix ⇒ un arbre par voix ⇒ représentation intermédiaire (RI) ;
  - Réécriture 2 :
    simplification de l'écriture de chaque voix dans la RI ;
- **Output** :
  export de la partition. Plusieurs formats sont possibles (xml, mei, lilypond,...).

Plusieurs enjeux :

- Problème du MIDI avec Qparse :
  ON-OFF en entrée ⇒ 1 seul symbole en sortie.
- Minimiser la distance entre le midi et la représentation en arbre.
- Un des problèmes de Qparse était qu'il était limité au monophonique.
  Quelles sont les limites du monophonique ?
  - Impossibilité de traiter plusieurs voix et de reconnaître les accords.

3.4 Les systèmes

Un système est la combinaison d'un ou de plusieurs éléments qui jouent un rythme en boucle (motif) et d'un autre élément qui joue un texte rythmique variable mais en respectant les règles propres au système (gamme).

Définitions

Système : motif + gamme/texte
Motif : rythmes coordonnés joués avec 2 ou 3 membres en boucle (répartis

5. https://gitlab.inria.fr/qparse/qparselib/-/tree/distance/src/scoremodel

sur 1 ou 2 voix)
Texte : rythme irrégulier joué avec un seul membre sur le motif (réparti sur 1 voix).
Gamme : la gamme d'un système considère l'ensemble des combinaisons que le batteur pourrait rencontrer en interprétant un texte rythmique à l'aide du système.

Un ensemble de systèmes comprenant leur métrique et leurs règles spécifiques de réécriture sera nécessaire. Les systèmes devront être distribués dans 4 grandes catégories :

Systèmes	Métriques	Subdivisions	Possibles	nb voix
binaires	simple	doubles-croches	triolets, sextolets	2
jazz	simple	triolets	croches et doubles-croches	2
ternaires	complexe	croches	duolets, quartelets	2
afros-cubains	simple	croches	-	3

TABLE 3.2 – Sytèmes

Nous exposerons 3 systèmes afin d'illustrer les propos de cette section :
- 4/4 binaire
- 4/4 jazz
- 4/4 afro-cubain

Objectif des systèmes

Les systèmes devront être matchés sur l'input MIDI afin de :
- définir une métrique ;
- choisir une grammaire appropriée ;
- fournir les règles de réécriture (séparation des voix et simplification.

La partie *motif* des systèmes sera utilisée pour la **définition des métriques**. Le *motif* et la gammes des systèmes seront utilisés pour la **séparation des voix**. Les règles de **simplification** (les combinaisons de réécritures) seront extraites des voix séparées des systèmes.

Détection d'indication de mesure

La détection de la métrique est importante, non seulement pour connaître le nombre de temps par mesure ainsi que le nombre de subdivisions pour chacun de ces temps, mais aussi pour savoir comment écrire l'unité de temps et ses subdivisions.

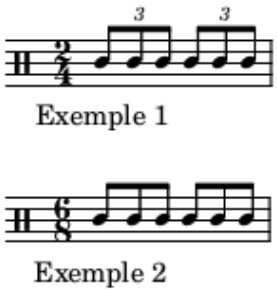


FIGURE 3.11 – Métrique

La figure 3.11 montre deux indications de mesure différentes. L'une (exemple 1) est *simple* (2 temps binaires sur lesquels sont joués des triolets), l'autre (exemple 2) est *complexe* (2 temps ternaires). Le jazz est traditionnellement écrit en binaire avec ou sans triolet (même si cette musique est dite ternaire alors que le rock ternaire sera plutôt écrit comme dans l'exemple 2).

Choix d'une grammaire

Il faut prendre en compte l'existence potentielle de plusieurs grammaires dédiées chacunes à un type de contenu MIDI. Le choix d'une grammaire pondérée doit être fait avant le parsing puisque Qparse prend en entrée un fichier MIDI et un fichier wta (grammaire). C'est pour cette raison que la métrique doit être définie avant le choix de la grammaire.
Pour les expériences effectuées avec le Groove MIDI Data Set, le style et l'indication de mesure sont récupérables par les noms des fichiers MIDI, mais il faudra par la suite les trouver automatiquement sans autres indications que les données MIDI elles-mêmes. Par conséquent, les motifs des systèmes devront être recherchés sur l'input *(fichiers MIDI)* avant le lancement du parsing, afin de déterminer la métrique en amont. Cette tâche devra probablement être effectuée en Machine Learning.

Séparation des voix

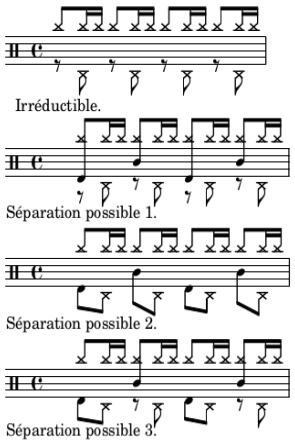


FIGURE 3.12 – Motif 4-4 binaire

Ici, le système est construit sur un modèle rock en 4/4 : after-beat sur les 2 et 4 avec un choix de répartition des cymbales type fast-jazz. Le système est constitué par défaut du motif rd/pf/cc (voir 3.1) et d'un texte joué à la grosse-caisse. La première ligne de la figure 3.12 est appelée « Irréductible » car il n'y a pas d'autre choix pertinent pour la répartition de la ride et du charley au pied. La troisième séparation proposée est privilégiée car elle répartit selon 2 voix, une voix pour les mains (rd + cc) et une voix pour les pieds (pf + gc). Ce choix paraît plus équilibré car deux instruments sont utilisés par voix et plus logique pour le lecteur puisque les mains sont en haut et les pieds en bas.

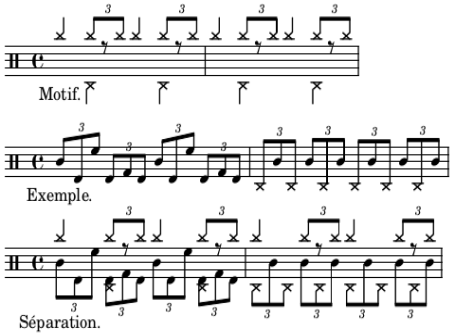


FIGURE 3.13 – Motif 4-4 jazz

Dans la plupart des méthodes, le charley n'est pas écrit car il est considéré comme évident en jazz traditionnel. Ce qui facilite grandement l'écriture : la ride et les crash sur la voix du haut et le reste sur la voix du bas. Ici, le parti pris est de tout écrire. Dans l'exemple ci-dessus, les mesures 1 et

2 combinées avec le *motif* de la première ligne, sont des cas typiques de la batterie jazz. Tout mettre sur la voix haute serait surchargé. De plus, la grosse caisse entre très souvent dans le flot des combinaisons de toms et de caisse claire et son écriture séparée serait inutilement compliquée et peu intuitive pour le lecteur. Le choix de séparation sera donc de laisser les cymbales en haut et toms, caisse-claire, grosse-caisse et pédale de charley en bas.

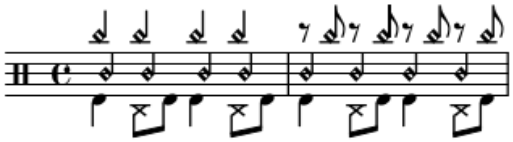


FIGURE 3.14 – Système 4-4 afro-latin

La figure 3.14 montre un exemple minimaliste de système afro-latin [22]. Ce système doit être écrit sur trois voix car la voix centrale est souvent plus complexe qu'ici (que des noirs) et la mélanger avec le haut ou le bas serait surchargé et peu lisible.

Simplification de l'écriture

Les gammes qui accompagnent les motifs d'un système étayent toutes les combinaisons d'un système. Elles sont générées manuellement à partir de la simplification de chaque voix d'un système selon les principes de la section 3.1 (aller à la section 4.3 pour voir un exemple).

Conclusion

Nous avons formalisé une notation de la batterie, modélisé cette notation pour la transcription de données MIDI en partition, nous avons décrit Qparse.
Enfin, nous avons exposé une approche de type dictionnaire (les « systèmes ») pour détecter une métrique, choisir une grammaire pondérée appropriée et énoncer des règles de séparation des voix et de simplification de l'écriture.

CHAPITRE 4

EXPÉRIMENTATIONS

Sommaire

Introduction

Dans ce chapitre, nous présenterons le jeu de données et les analyses audio-MIDI. Nous ferons ensuite l'expérimentation théorique d'un *système* implémentable qui devra être utilisé comme base de connaissances pour augmenter la rapidité et la qualité en sortie de Qparse. Enfin, nous présenterons les différentes contributions de développement.

4.1 Le jeu de données

Nous avons utilisé le Groove MIDI Dataset [1] [23] (GMD) qui est un jeu de données mis à disposition par Google sous la licence Creative Commons Attribution 4.0 International (CC BY 4.0).
Le GMD est composé de 13,6 heures de batterie sous forme de fichiers MIDI et audio alignés. Il contient 1150 fichiers MIDI et plus de 22 000 mesures de batterie dans les styles les plus courants et avec différentes qualités de jeu. Tout le contenu a été joué par des humains sur la batterie électronique Roland TD-11 (figure 4.1).

1. https://magenta.tensorflow.org/datasets/groove

FIGURE 4.1 – Batterie électronique
Source : https://www.youtube.com/watch?v=BX1V_IE0g2c

Autres critères spécifiques au GMD :

- Toutes les performances ont été jouées au métronome et à un tempo choisi par le batteur.
- 80% de la durée du GMD a été joué par des batteurs professionnels qui ont pu improviser dans un large éventail de styles. Les données sont donc diversifiées en termes de styles et de qualités de jeu (professionnel ou amateur).
- Les batteurs avaient pour instruction de jouer des séquences de plusieurs minutes ainsi que des fills [2]
- Chaque performance est annotée d'un style (fourni par le batteur), d'une métrique et d'un tempo ainsi que d'une identification anonyme du batteur.
- Il a été demandé à 4 batteurs d'enregistrer le même groupe de 10 rythmes dans leurs styles respectifs. Ils sont dans les dossiers eval-session du GMD.
- Les sorties audio synthétisées ont été alignées à 2 ms près sur leur fichier MIDI.

Format des données

Le Roland TD-11 divise les données enregistrées en plusieurs pistes distinctes :

- une pour le tempo et l'indication de mesure ;
- une pour les changements de contrôle (position de la pédale de charley) ;
- une pour les notes.

Les changements de contrôle sont placés sur le canal 0 et les notes sur le canal 9 (qui est le canal canonique pour la batterie).
Pour simplifier le traitement de ces données, ces trois pistes ont été fusionnées en une seule piste qui a été mise sur le canal 9.

2. Un *fill* est une séquence de relance dont la durée dépasse rarement 2 mesures. Il est souvent joué à la fin d'un cycle pour annoncer le suivant.

« Control Changes The TD-11 also records control changes specifying the position of the hi-hat pedal on each hit. We have preserved this information under control 4. »
(https://magenta.tensorflow.org/datasets/groove)
⇒ ??? Je ne comprends pas encore comment trouver ce type d'informations dans les fichiers MIDI.
L'utilisation de pretty_midi devient urgente !

4.2 Analyse MIDI-Audio

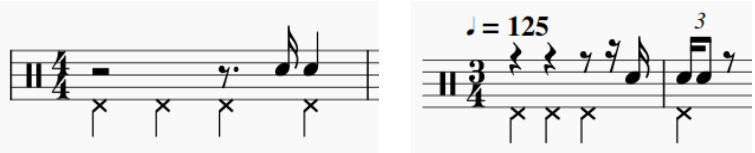
Ces analyses ont été faites dans le cadre de transcriptions manuelles à partir de fichiers MIDI et Audio du GMD.

Comparaisons de transcriptions

Pour les comparaisons de transcriptions, les transcriptions manuelles (TM) ont été éditées à l'aide de Lilypond [3] ou MuseScore [4] et les transcriptions automatiques (TA) ont toutes été générées manuellement avec MuseScore.

Exemple d'analyse 1

Transcription manuelle ⇒ Transcription automatique

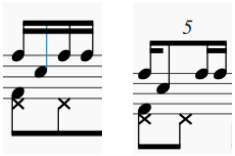


— Erreur d'indication de mesure (3/4 au lieu de 4/4) ;
— Les silences de la mesure 1 de la TA sont inutilement surchargés ;
— La noire du temps 4 de la mesure 1 de la TM est devenue les deux premières notes (une double-croche et une croche) d'un triolet sur le temps 1 de la mesure 2 de la TA.

3. http://lilypond.org/
4. https://musescore.com/

Exemple d'analyse 2

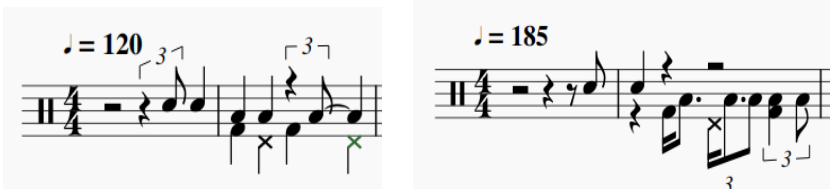
Transcription manuelle ⇒ Transcription automatique



— Les doubles croches ont été interprétées en quintolet
— La deuxième double-croche est devenue une croche.

Exemple d'analyse 3

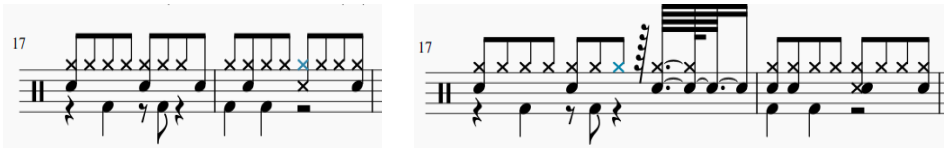
Transcription manuelle ⇒ Transcription automatique



— Les grosses-caisses, les charleys et les caisses-claires ont été décalés d'un temps vers la droite.
— Les toms basses des temps 1 et 2 de la mesure 2 de la TM ont été décalés d'une double croche vers la droite dans la TA.
— La première caisse-claire de la mesure 1 devient binaire dans la TA alors qu'elle appartenait à un triolet dans la TM.
— Le triolet de tom-basse du temps 4 de la mesure 2 de la TA n'existe pas la TM.

Exemple d'analyse 4

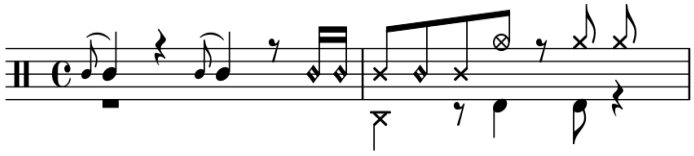
Transcription manuelle ⇒ Transcription automatique



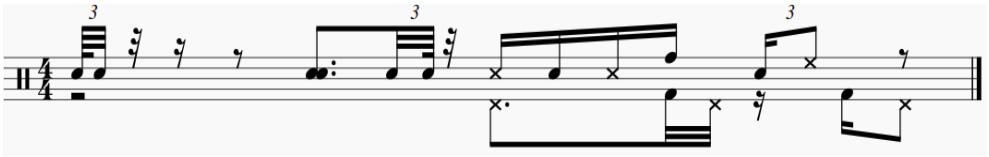
Sur le temps 4 de la mesure 1, la deuxième croche a été transcrite d'une manière excessivement complexe !

Exemple avec des flas

Transcription manuelle



Transcription automatique



— Le premier fla est reconnu comme étant un triolet contenant une quadruple croche suivie d'une triple croche au lieu d'une seule note ornementée.
— Le deuxième fla est reconnu comme étant un accord.
— Les deux double en l'air sur le temps 4 de la TM sont mal quantifiée dans la TA.
— La TA ne reconnaît qu'une mesure quand la TM en transcrit deux. En effet, la TA a divisé par deux la durée des notes afin de les faire tenir dans une mesure à 4 temps dont les unités de temps sont les noires. Par exemple, le soupir du temps 2 de la TM devient un demi-soupir sur le contre-temps du temps 1 dans la TA. Ou encore, la noire (pf, voir le tableau 3.1) sur le temps 1 de la mesure 2 de la TM suivie d'un demi-soupir devient une croche pointée sur le temps 3 de la TA.
— Autre problème : certaines têtes de notes sont mal attribuées. Par exemple, le charley ouvert en l'air sur le temps 2 de la mesure 2 de la TM devrait avoir le même symbole sur la TA. Idem pour les cross-sticks.

Transcription de partition

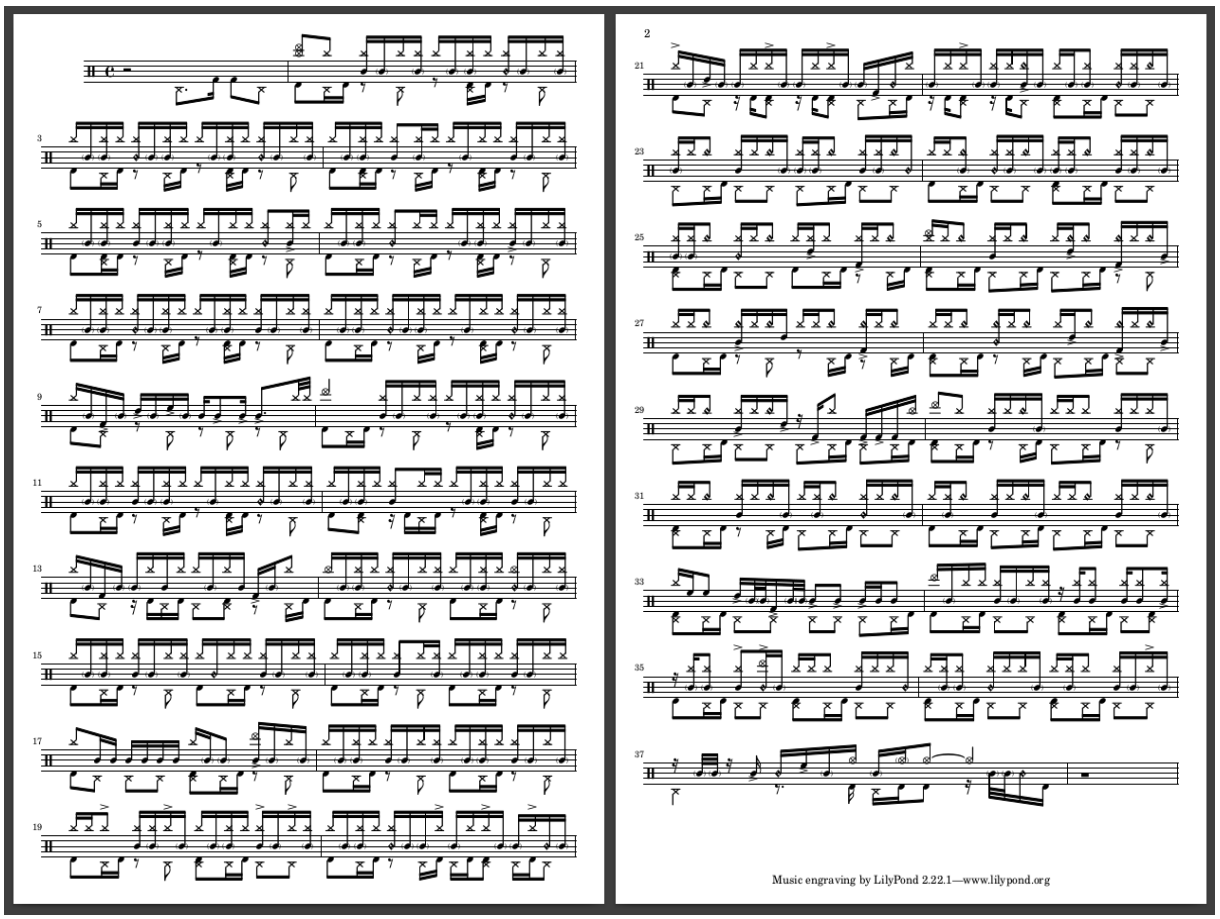


FIGURE 4.2 – Partition de référence

La figure 4.2 est la transcription manuelle des fichiers *004_jazz-funk_116_beat_4-4.mid* et *004_jazz-funk_116_beat_4-4.wav* du GMD.
Cette transcription a été entièrement faite avec Lilypond (voir le code lilypond sur le git https://github.com/MartinDigard/Stage_M2_Inria) Il s'agit d'une partition d'un 4/4 binaire dont le fichier MIDI est annoncé dans le GMD de style «jazz-funk» probablement en raison de la ride de type shabada rapide (le ternaire devient binaire avec la vitesse) combiné avec l'after-beat de type rock (caisse-claire sur les deux et quatre).
La transcription des données audio et MIDI contenues dans ces fichiers a permis une analyse plus approndie des critères à relever pour chaque évènement MIDI et de la manière de les considérer dans un objectif de transcription en partition lisible pour un musicien (Voir la section 3.2).

4.3 Expérimentation théorique d'un système

Cette expérimentation théorique, basée sur la partition de référence de la figure 4.2, montre le procédé de création d'un *système* et des règles qui en découlent (métrique, choix de grammaire, règles de séparation des voix et de simplification de l'écriture). Le *système* devra ensuite être implémenté pour appliquer des tests qui seront effectués, dans un premier temps, sur la partition de référence.

Motifs et gammes

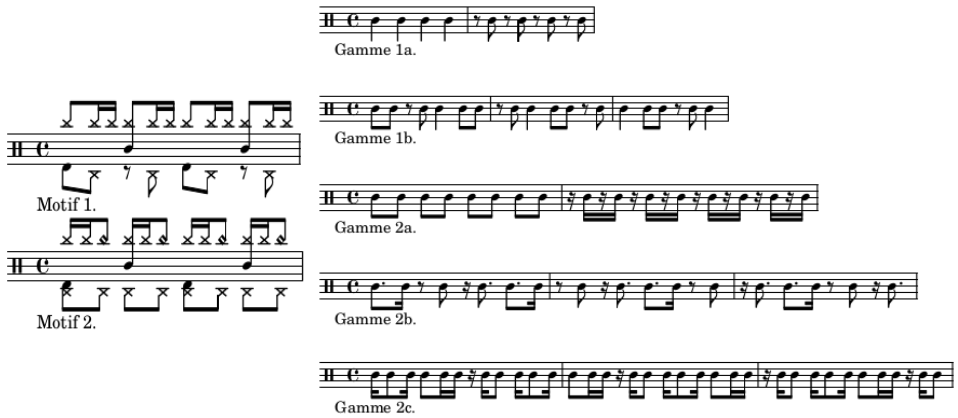


FIGURE 4.3 – Motifs et gammes

Motifs

À partir de la partition de référence, les deux motifs de la figure 4.3 peuvent être systématisés. Le motif 1 est joué du début jusqu'à la mesure 18 avec des variations et des fills et le motif 2 est joué de la mesures 23 à la mesure 28 avec des variations. Ces deux motifs sont très classiques et pourront être détectés dans de nombreuses performances.

Gammes

Les gammes de la figure 4.3 étayent toutes les combinaisons d'un motif en 4/4 binaires jusqu'aux doubles croches.
Les lignes 1 et 2 traitent les croches. La ligne 1 a 2 mesures dont la première ne contient que des noires et la deuxième que des croches en l'air. Ces deux possibilités sont combinées de manière circulaire dans les 3 mesures de la deuxième ligne.
Les lignes 3, 4 et 5 traitent les doubles-croches. La ligne 3 a 2 mesures

dont la première ne contient que des croches et la deuxième que des doubles-croches en l'air. Ces deux possibilités sont combinées de manière circulaire dans les lignes 4 et 5 qui contiennent chacunes 3 mesures.

Systèmes — motifs et gammes combinés

Pour la suite de l'expérimentation théorique, nous utiliserons le motif 1 de la figure 4.3.

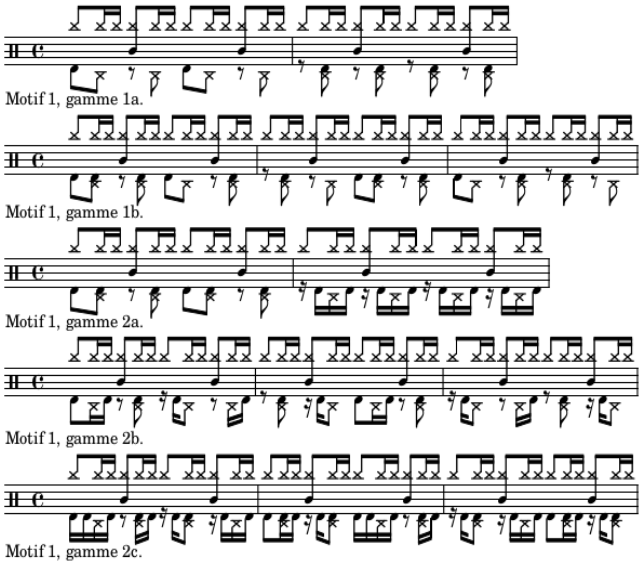


FIGURE 4.4 – Partition d'un système en 4/4 binaire

Représentation du système en arbres de rythmes

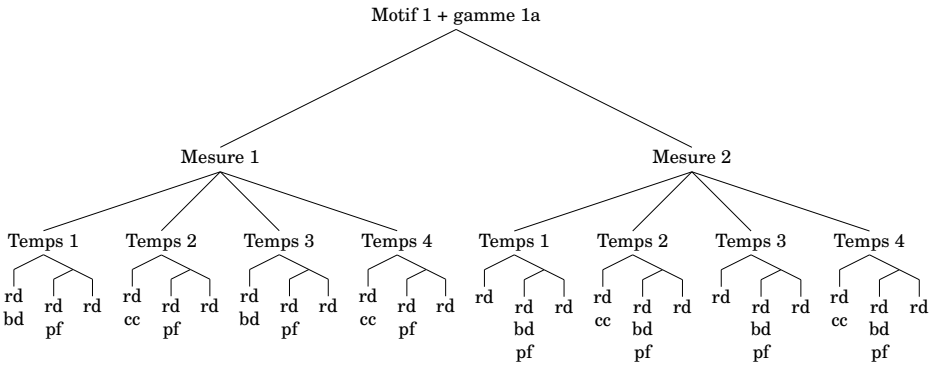


FIGURE 4.5 – Arbre de rythme — système

L'arbre de la figure 4.5 servira de base pour le suite de l'expérimentation. Comme indiqué à la racine de l'arbre, il représente la première ligne de la figure 4.4. Même si cet arbre représente parfaitement le rythme concerné, il manque des indications de notation telles que les voix spécifiques à chaque partie du rythme ainsi que les choix d'écriture pour les distances qui séparent les notes de chaque voix entre elles en termes de durée.

Réécriture — séparation des voix et simplification

La séparation des voix

Ainsi l'arbre syntaxique de départ est divisé en autant d'instruments qui le constituent et les voix seront regroupées en suivant les régles du système.

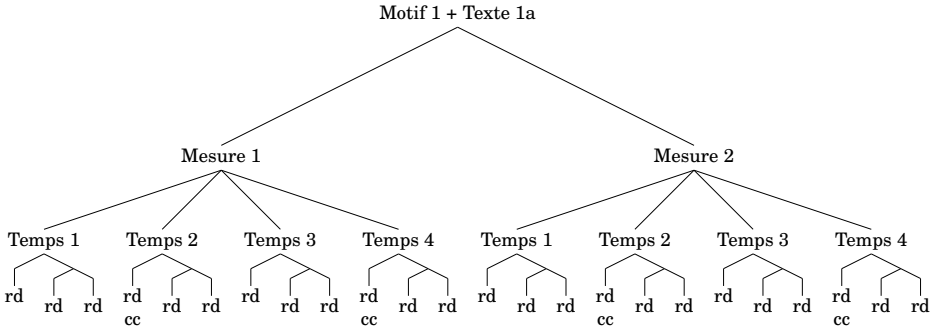


FIGURE 4.6 – Arbre de rythme — voix haute

La voix haute regroupe la ride et la caisse-claire sur les ligatures du haut.

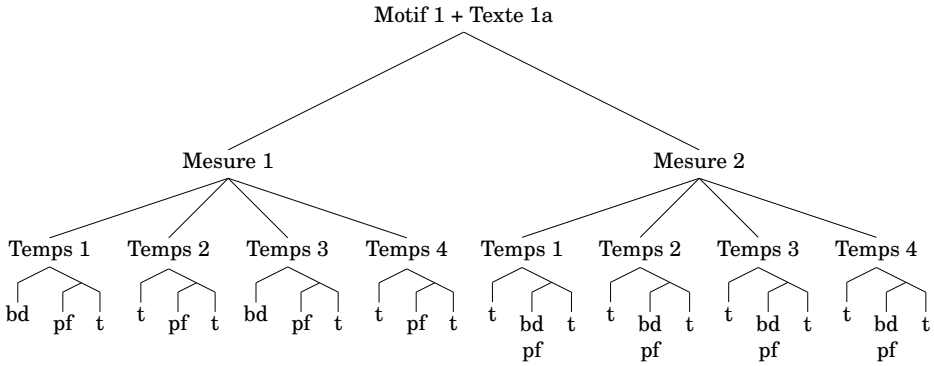


FIGURE 4.7 – Arbre de rythme — voix basse

La voix basse regroupe la grosse-caisse et le charley au pied sur les ligatures du bas.

Les règles de simplifications

L'objectif des simplifications est de réécrire les écarts de durées qui séparent les notes d'une manière appropriée pour la batterie et qui soit la plus simple possible.
Pour la notation soit appropriée à la batterie, il faut que les ligatures relient les notes d'un temps entre elles (rendre la pulse visuelle).

Pour les figures ci-dessous :
- x = une note ;
- r = un silence ;
- t = une continuation (point ou liaison)

Simplifier l'écriture de chaque voix (*Règles établis par le système*)

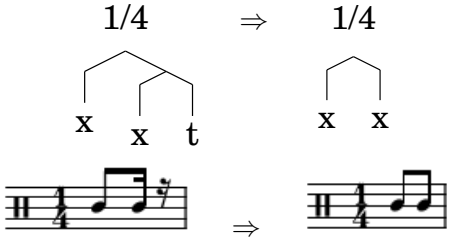


FIGURE 4.8

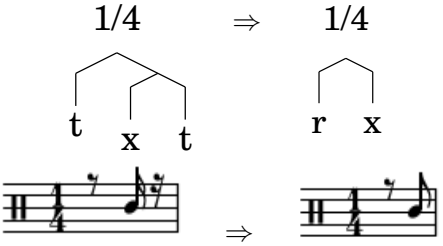


FIGURE 4.9

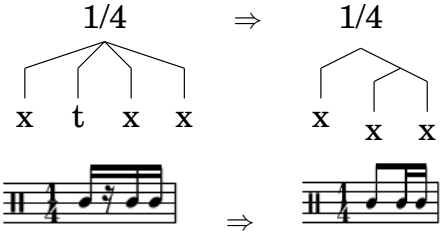


FIGURE 4.10

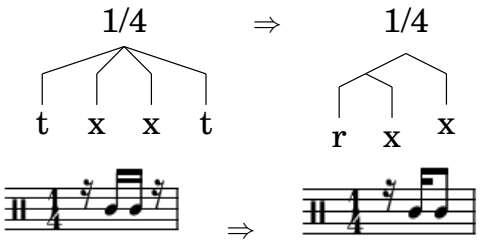


FIGURE 4.11

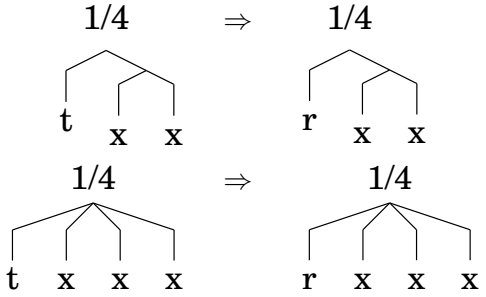


FIGURE 4.12

Objectif de cette expérimentation théorique

La méthode des *systèmes* étant basée sur une approche dictionnaire, cette expérimentation théorique a pour but d'orienter la recherche d'autres systèmes par observation du jeu de données et de montrer comment les construire pour agrandir la base de connaissance de Qparse pour l'ADT.

4.4 Développement

DrumCode

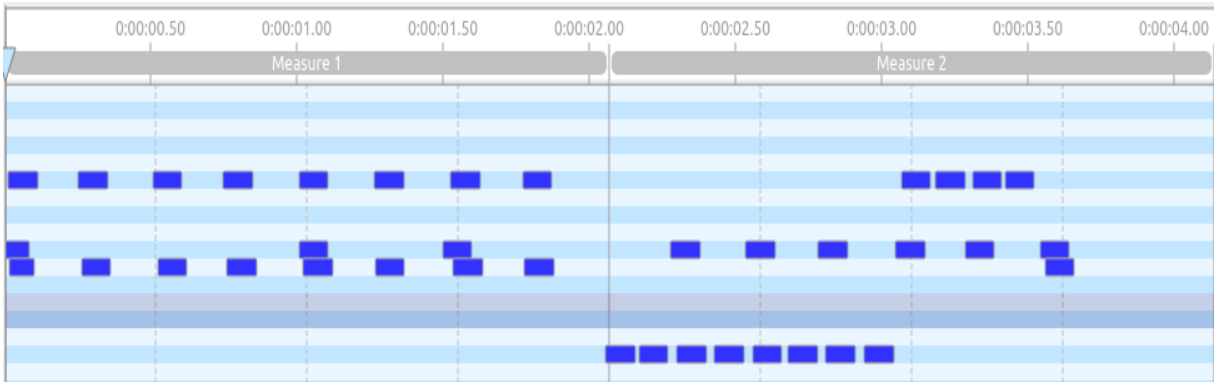
Adaptation de la modélisation pour la transcription en cpp.

Tests unitaires sur les Jams

Les Jams permettent de passer du monophonique au polyphonique.

Parsing

Tests effectués avec le fichier midi suivant :



Un premier test convaincant est effectué avec la grammaire suivante :

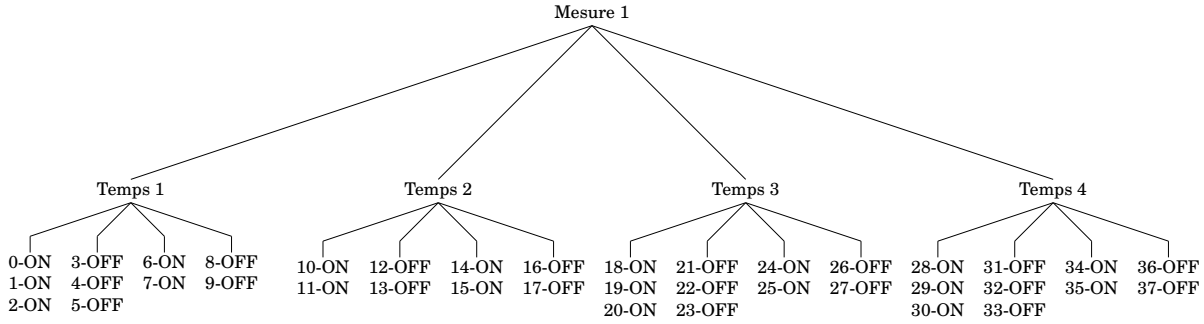
// bar level
0 -> C0 1
0 -> E1 1
0 -> U4(1, 1, 1, 1) 1

// half bar level
9 -> C0 1
9 -> E1 1

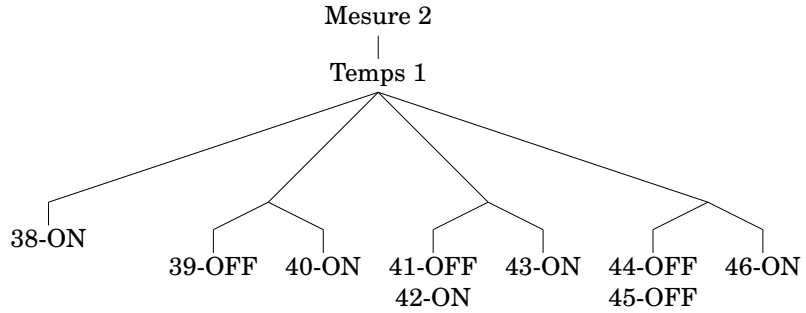
// beat level
1 -> C0 1
1 -> E1 1
1 -> T2(2, 2) 1
1 -> T4(4, 4, 4, 4) 1

```
// croche level
2 -> C0 1
2 -> E1 1

// double level
4 -> C0 1
4 -> E1 1
4 -> E2 1
4 -> T2(6, 6) 1

// triple level
6 -> E1 1
```

Cette grammaire sépare les ligatures par temps au niveau de la mesure. Puis, au niveau du temps, elle autorise les divisions par deux (croches) et par quatre (doubles-croches). Tous les poids sont réglés sur 1. L'arbre de parsing en résultant est considéré comme « convaincant » car il découpe correctement les mesures et les temps.



Les temps de la première mesure du fichier MIDI sont bien quantifié mais ceux de la deuxième mesure présentent quelques défauts de quantification visibles dès le premier temps.



Les Onsets sont correctement triés au niveau des doubles croches mais certaines doubles croches sont inutilement subdivisées en triples croches (les 2ème, 3ème et 4ème doubles croches sur le premier temps ci-dessus).

2ème exemple :
Après une augmentation du poids des triples croches dans la grammaire (monté de 1 à 5)et une baisse de tous les autres poids (descendu de 1 à 0.5), et mis à part le troisième temps de la 2ème mesure, tous les Onsets sont bien triés et aucuns ne sont subdivisés.

Conclusion

Conclusion de ce chapitre.

CHAPITRE 5

DISCUSSION

Sommaire

Introduction

Dans ce chapitre, nous discuterons sur la pertinence de l'ensemble des choix qui ont été faits. Nous ferons un bilan des différentes avancées qui ont été faites ou non et nous tenterons d'en expliquer la ou les raisons.

5.1 Bilan sur les travaux réalisés

Résultats

Évaluation

5.2 Travaux futures

Écrire des règles de réécriture spécifique aux charley avec un système approprié. Le jeu de système

— implémenter un pattern...
⇒ manque de temps ?

— La partie résultat est manquante car :
⇒ Sujet très difficile ;
⇒ Matcher les motifs peut être fait ultérieurement ;
Mais ce travail aurait été indispensable pour obtenir une quantité de résultats qui justifieraient une évaluation automatique permettant de faire des graphiques.

— L'évaluation fut entièrement manuelle car :
  ⇒ Très dure automatiquement : il faut comparer 2 partitions (réf VS output)
— Le ternaire jazz (voir expérience 2)
— Reconnaissance d'un motif sur le MIDI
  Reconnaître un motif (système) sur une mesure de l'input (un fichier midi représentant des données audios)
  ⇒ Motif (système) reconnu : true ou false
  Si true :
  - Choisir la grammaire correspondante ;
  - Parser le MIDI ;
  - Appliquer les règles de réécritures (Séparation des voix et simplification)
— Nous travaillerons aussi sur la détection de répétitions sur plusieurs mesures afin de pouvoir corriger des erreurs sur une des mesures qui aurait dû être identique aux autres mais qui présente des différences.
— dans quelle catégorie mettre le shuffle ?

Conclusion

Sujet passionnant mais difficile. Obtenir la totalité des critères pour le mémoire n'aurait pas pu être fait sans bâcler. Une base solide spécifique à la batterie a été générée. Elle sera un bon point de départ pour les travaux futurs dont plusieurs propositions sont énoncés dans le présent document.

CONCLUSION GÉNÉRALE

Dans ce mémoire, nous avons traité de la problématique de la transcription automatique de la batterie. Son objectif était de transcrire, à partir de leur représentation symbolique MIDI, des performances de batteur de différents niveaux et dans différents styles en partitions écrites.
Nous avons avancé sur le parsing des données MIDI établissant un processus de regroupement des évènements MIDI qui nous a permis de faire la transition du monophonique vers le polyphonique. Une des données importante de ce processus était de différencier les nature des notes d'un *accord*, notamment de distinguer lorsque 2 notes constituent un *accord* ou un *fla*.
Nous avons établis des *grammaires pondérées* pour le parsing qui correspondent respectivement à des métriques spécifiques. Celles-ci étant sélectionnables en amont du parsing, soit par indication des noms des fichiers MIDI, soit par reconnaissance de la métrique avec une approche dictionnaire de patterns prédéfinis [1] qu'il serait pertinent de mettre en œuvre en machine learning.
Nous avons démontré que l'usage des *systèmes* élimine un grand nombre de calcul lors de la réécriture. Pour la séparation des voix grâce au motif d'un système et pour la simplification grâce aux gammes du motif d'un système. Nous avons aussi montré comment, dans des travaux futurs, un système dont le motif serait reconnu en amont dans un fichier MIDI pourrait prédéfinir le choix d'une grammaire par la reconnaissance d'une métrique et ainsi améliorer le parsing et accélérer les choix ultérieurs dans la chaîne de traitement en terme de réécriture.
Il sera également intéressant d'étudier comment l'utilisation de LM peut améliorer les résultats de l'AM, voir [2], et ouvrir la voie à la génération entièrement automatisée de partitions de batterie et au problème général de l'AMT de bout en bout.[7]

1. *Motifs* dans les *systèmes* de la présente proposition.

BIBLIOGRAPHIE

[1] H. C. Longuet-Higgins. Perception of melodies. 1976. – Cité pages 9 et 12.

[2] Meinard Müller. *Fundamentals of Music Processing*. 01 2015. – Cité page 9.

[3] Caroline Traube. Quelle place pour la science au sein de la musicologie aujourd'hui ? *Circuit*, 24(2) :41–49, 2014. – Cité page 10.

[4] Bénédicte Poulin-Charronnat and Pierre Perruchet. Les interactions entre les traitements de la musique et du langage. *La Lettre des Neurosciences*, 58 :24–26, 2018. – Cité page 11.

[5] Mikaela Keller, Kamil Akesbi, Lorenzo Moreira, and Louis Bigo. Techniques de traitement automatique du langage naturel appliquées aux représentations symboliques musicales. In *JIM 2021 - Journées d'Informatique Musicale*, Virtual, France, July 2021. – Cité page 11.

[6] Junyan Jiang, Gus Xia, and Taylor Berg-Kirkpatrick. Discovering music relations with sequential attention. In *NLP4MUSA*, 2020. – Cité page 11.

[7] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri. Automatic music transcription : Challenges and future directions. *Journal of Intelligent Information Systems*, 41, 12 2013. – Cité pages 12, 14, 19 et 57.

[8] Kentaro Shibata, Eita Nakamura, and Kazuyoshi Yoshii. Non-local musical statistics as guides for audio-to-score piano transcription. *Information Sciences*, 566 :262–280, 2021. – Cité pages 12, 21 et 22.

[9] Daniel Harasim, Christoph Finkensiep, Petter Ericson, Timothy J O'Donnell, and Martin Rohrmeier. The jazz harmony treebank. – Cité pages 12 et 23.

[10] Chih-Wei Wu, Christian Dittmar, Carl Southall, Richard Vogl, Gerhard Widmer, Jason Hockman, Meinard Müller, and Alexander Lerch. A review of automatic drum transcription. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(9) :1457–1483, 2018. – Cité pages 14, 20 et 23.

[11] Moshekwa Malatji. Automatic music transcription for two instruments based variable q-transform and deep learning methods, 10 2020. – Cité page 20.

[12] Antti J. Eronen. Musical instrument recognition using ica-based transform of features and discriminatively trained hmms. *Seventh International Symposium on Signal Processing and Its Applications, 2003. Proceedings.*, 2 :133–136 vol.2, 2003. – Cité page 20.

[13] Hiroshi G. Okuno Kazuyoshi Yoshii, Masataka Goto. Automatic drum sound description for real-world music using template adaptation and matching methods. *International Conference on Music Information Retrieval (ISMIR)*, pages 184–191, 2004. – Cité page 20.

[14] Francesco Foscarin, Florent Jacquemard, Philippe Rigaux, and Masahiko Sakai. A Parse-based Framework for Coupled Rhythm Quantization and Score Structuring. In *MCM 2019 - Mathematics and Computation in Music*, volume Lecture Notes in Computer Science of *Proceedings of the Seventh International Conference on Mathematics and Computation in Music (MCM 2019)*, Madrid, Spain, June 2019. Springer. – Cité pages 21 et 22.

[15] C. Agon, K. Haddad, and G. Assayag. Representation and rendering of rhythm structures. In *Proceedings of the First International Symposium on Cyber Worlds (CW'02)*, CW '02, page 109, USA, 2002. IEEE Computer Society. – Cité page 22.

[16] Florent Jacquemard, Pierre Donat-Bouillud, and Jean Bresson. A Term Rewriting Based Structural Theory of Rhythm Notation. Research report, ANR-13-JS02-0004-01 - EFFICACe, March 2015. – Cité page 22.

[17] Florent Jacquemard, Adrien Ycart, and Masahiko Sakai. Generating equivalent rhythmic notations based on rhythm tree languages. In *Third International Conference on Technologies for Music Notation and Representation (TENOR)*, Coroña, Spain, May 2017. Helena Lopez Palma and Mike Solomon. – Cité page 22.

[18] R. Marxer and J. Janer. Study of regularizations and constraints in nmf-based drums monaural separation. In *International Conference on Digital Audio Effects Conference (DAFx-13)*, Maynooth, Ireland, 02/09/2013 2013. – Cité page 23.

[19] A. Danhauser. *Théorie de la musique*. Edition Henry Lemoine, 41 rue Bayen - 75017 Paris, Édition revue et augmentée - 1996 edition, 1996. – Cité pages 25, 26 et 31.

[20] J.-F. Juskowiak. *Rythmiques binaires 2*. Alphonse Leduc, Editions Musicales, 175, rue Saint-Honoré, 75040 Paris, 1989. – Cité page 26.

[21] Dante Agostini. *Méthode de batterie, Vol. 3*. Dante Agostini, 21, rue Jean Anouilh, 77330 Ozoir-la-Ferrière, 1977. – Cité page 26.

[22] O. Lacau J.-F. Juskowiak. *Systèmes drums n. 2*. MusicCom publications, Editions Joseph BÉHAR, 61, rue du Bois des Joncs Marins - 94120 Fontenay-sous-Bois, 2000. – Cité pages 27 et 39.

[23] Jon Gillick, Adam Roberts, Jesse Engel, Douglas Eck, and David Bamman. Learning to groove with inverse sequence transformations. In *International Conference on Machine Learning (ICML)*, 2019. – Cité page 41.