

Faculté Polytechnique

Conception et réalisation d'un générateur musical intelligent, modulaire et embarqué

Travail de fin d'études

Réalisé au sein du
Input Devices and Music Interaction Laboratory de l'Université McGill

Rapport de TFE

Robin VANDEBROUCK



IDMIL

Sous la supervision de
Pr. Ir. Thierry DUTOIT, Université de Mons, Belgique
Pr. Ir. Marcelo WANDERLEY, Université McGill, Canada

Année Académique 2021 - 2022

Remerciements

Tout d'abord, je tiens à remercier Monsieur Thierry Dutoit, sans qui tout ce travail, et le contexte qui l'accompagne, n'auraient pas été possibles.

Je remercie également Monsieur Marcelo Wanderley pour m'avoir accueilli en toute sympathie et encadré à Montréal.

Je remercie tous les aimables membres du *Input Devices and Music Interaction Laboratory* de m'avoir accompagné et aidé.

Je remercie aussi Brady Boettcher avec qui collaborer a été un vrai plaisir.

Je remercie en outre Maxime Maton pour son aide précieuse.

Enfin, je remercie la faculté polytechnique pour ces cinq années d'études et cette fin de parcours à l'étranger.

Résumé

Le travail présenté dans ce rapport consiste en la création d'un générateur intelligent, modulaire et embarqué. Ce travail s'inscrit dans la suite d'un projet préexistant du *Input Devices and Music Interaction Laboratory* nommé Probatio. Celui-ci est une boîte à outils de prototypage pour la création de nouveaux instruments de musique numériques. Pour cela, il se compose de diverses bases sur lesquelles un utilisateur peut placer arbitrairement des blocs, eux-mêmes constitués de diverses méthodes d'interactions (bouton, potard, percussion, etc.). Les deux objectifs maîtres du Probatio étant la proposition de nouvelles approches de création et la réduction du temps de prototypage, ce travail propose une nouvelle approche encore jamais implémentée dans le cadre du Probatio : une génération intelligente. Cette génération représente une solution aux deux problématiques. Ainsi, le générateur propose un nouvel outil de test et d'improvisation, mais surtout permet la suppression de la nécessité d'installation d'un instrument « classique » dans une situation où l'utilisateur désirerait adjoindre un appareil musical numérique à un autre instrument existant. En effet, une installation d'instrument est dans la majorité des cas longue et chronophage, ce qui n'est pas l'idéal pour un premier pas dans le prototypage. Ce nouveau bloc de génération pour le Probatio permettra ainsi de la création d'un flux continu de commandes audio de type MIDI pouvant être utilisé dans n'importe lequel des outils de synthèse audio. Cette synthèse peut alors remplacer le son de l'instrument que l'utilisateur souhaite améliorer. Pour sa construction, le générateur se base sur l'utilisation de Music Transformer [11] et, par un processus de parallélisation, rend possible la génération en continu, la lecture et le contrôle des paramètres de lecture.

Ce générateur se veut également modulaire et embarqué de manière à pouvoir être utilisé dans le plus de contextes possibles. A cette fin, un Proof-of-Concept a pu être réalisé avec un Raspberry Pi. Ce POC a notamment permis de mettre en évidence la possibilité d'une adaptation du code pour une génération en ligne. Cette configuration a également mis en exergue l'intérêt qu'elle avait de faire tomber toutes les contraintes de performance embarquée de la carte, à condition de disposer d'une connexion Internet. Un patch Pure-Data a également pu être conçu de façon à gérer de manière complètement locale (sur une carte de type Raspberry) les interactions avec les blocs du Probatio.

Pour tester la fonctionnalité de l'implémentation, des sessions d'enregistrement ont pu être réalisées avec et sans le Probatio. Certaines de ces performances ont également été utilisées pour créer un sondage permettant d'évaluer la pertinence du travail. Combiné à des analyses de performance, il en ressort que le générateur est tout à fait fonctionnel et pertinent, notamment de par sa robustesse et la qualité qu'il apporte à la lecture en série. De cette façon, cet outil répond de manière qualitative aux questions de recherche du Probatio, et fournit de nouvelles fonctionnalités pour les réseaux génératrices de fichiers.

Table des matières

Introduction	1
1 Contextualisation & État de l'art	3
1.1 Le monde de la musique aujourd'hui	3
1.2 Instruments de musique numériques	6
1.2.1 Définition et potentiel	6
1.2.2 De nouvelles problématiques	7
1.2.3 Utilisation et types de performance	7
1.2.4 Exemples d'instruments de musique numériques	8
1.2.5 Récapitulatif	10
1.3 Intelligence Artificielle et musique	10
1.3.1 L'Intelligence Artificielle : les fondamentaux	11
1.3.2 Deep Music Generation	13
1.3.3 Les problématiques de l'évaluation	17
2 Outils	20
2.1 MIDI	20
2.2 Google Magenta	22
2.3 Synthèse audio	23
2.4 Libmapper	24
3 Le Probatio	25
4 Apports personnels	27
5 Développement du générateur musical	30
5.1 Approche globale	30
5.2 Présentation du réseau de neurones exploité	32
5.2.1 Les transformers	32
5.2.2 Music Transformer	33
5.3 Implémentation du système	34
5.4 Systèmes embarqués	35
5.4.1 Première approche : le tout embarqué	36
5.4.2 Deuxième approche : génération en ligne	36
5.4.3 Troisième approche : Ajout d'une unité locale de traitement de signal	38

6 Résultats	41
6.1 Analyse de la productivité du générateur	42
6.2 Appréciation humaine et test qualitatif	43
6.3 Analyse quantitative du générateur	47
6.4 Analyse critique	48
7 Perspectives d'amélioration	50
Conclusion	51
Annexes	
A Détails sur Music Transformer	56
A.1 Représentation des données	56
A.2 Self-attention dans les transformers	56
A.3 Self-attention et position relative	57
B Lignes chronologiques des différentes générations musicales	58
C Qu'est-ce que la musicalité ?	62
D Formulaire du test de perception humaine	64

Table des figures

1.1	Recettes de l'industrie mondiale de la musique enregistrées 1999 - 2021 (en milliards de dollars US).	4
1.2	Localisation des conférences NIME par continent.	5
1.3	Tendance des 10 termes uniques les plus courants dans le corps du texte des articles NIME publiés.	5
1.4	Le cadre de Rasmussen appliqué à l'interaction musicale.	8
1.5	La Slapbox	9
1.6	Le T-Stick	10
1.7	Types de réseaux de neurones : non profond à gauche et profond à droite.	11
1.8	Le perceptron.	12
1.9	Exemples de fonctions d'activation	12
1.10	Processus d'entraînement et ajustement des poids.	13
1.11	Processus de génération musicale.	14
1.12	Chronologie de la composition de performances expressives. Les couleurs représentent les différentes architectures de modèles.	16
1.13	Flux de travail général de la méthode proposée dans [28].	19
2.1	Connectiques MIDI DIN femelles	21
2.2	Connectique MIDI DIN male	21
2.3	Magenta Studio	22
2.4	Exemple de fichier Pure Data implémenté (ring modulator).	23
3.1	Exemples de plusieurs blocs du projet Probatio.	25
3.2	Exemple de construction de Probatio.	26
4.1	Remplacement du boîtier externe du M5.	29
4.2	Nouveau bloc de connexion du M5.	29
5.1	Illustration du processus d'influence du passé sur la génération à travers des lignes d'attention.	34
5.2	Photo du POC : système embarqué et génération en ligne.	37
5.3	Exemple d'utilisation de l'objet [mapper].	38
5.4	Exemple blocs Pure Data : la batterie.	39
6.1	Évolution de la quantité de musique générée et consommée en fonction du temps.	42
6.2	Informations déduites des résultats du test d'appréciation.	46
6.3	Informations déduites des résultats du test d'intérêt/musicalité.	47
6.4	Évaluation de la compatibilité des sets générés et de la base de données MAESTRO.	48

B.1	Ligne du temps : génération de partition monophonique.	58
B.2	Ligne du temps : génération de partition polyphonique.	59
B.3	Ligne du temps : génération de partition multi-track.	59
B.4	Ligne du temps : harmonisation de mélodie.	59
B.5	Ligne du temps : génération de caractéristiques de performance.	60
B.6	Ligne du temps : génération de performance expressive.	60
B.7	Ligne du temps : génération de synthèse audio.	61
B.8	Ligne du temps : génération de synthèse de voix humaine.	61
D.1	Formulaire du test de perception, page 1.	65
D.2	Formulaire du test de perception, page 2.	66

Introduction

Depuis la nuit des temps, l'Homme est animé d'une volonté créatrice qui lui a permis d'évoluer, de s'exprimer, d'exister. La nature de cette volonté a toujours su se montrer variée, à la fois dans ses formes et ses accomplissements. De la technique à l'artistique, de la fabrication des premiers outils aux peintures rupestres, le champ de la créativité n'a jamais connu comme frontière que les limites de l'imagination.

Aujourd'hui encore cette volonté perdure et s'est même vue décuplée par tout ce qu'offrait, en termes de possibilités, le domaine des nouvelles technologies. Le *Input Devices and Music Interaction Laboratory* (IDMIL) de l'Université McGill s'inscrit lui aussi dans cette vague créatrice et s'intéresse à la conception de nouvelles interfaces pour l'expression musicale. Ce domaine spécifique d'activité est par ailleurs très intéressant, car venant justement à l'intersection des mondes de la technique et de l'artistique. Il pourrait même être ajouté que dans de tels processus créatifs, ce sont ces interactions entre domaines d'activité différents qui donne, à l'un comme à l'autre, un réel sens.

Comme il a pu être développé au cours de ces années d'études, l'Ingénierie est une science tentaculaire présente à travers le monde et dans tout domaine. La réalisation de ce travail de fin d'études en est une preuve. Ainsi dans le cadre du développement de ce TFE, le domaine de la musique est mis sur le devant de la scène. De manière plus précise, ce travail s'inscrit dans la suite d'un projet existant de l'IDMIL. Celui-ci porte le nom de Probatio et peut résumer son approche comme la fourniture d'une boîte à outils pour le prototypage de nouveaux instruments de musique numériques. Pour cela, Probatio met à disposition un ensemble de bases et de structures dans lesquels il est possible de placer arbitrairement des blocs de différentes natures. Ceux-ci représentent alors différentes manières d'interagir avec un instrument, que ce soit en le tapant, en utilisant des boutons, des éléments tournants, etc. L'idée derrière tout cela est d'offrir un premier pas plus aisément dans le prototypage musical. Bien évidemment, comme dans la plupart des cas, ce domaine spécifique des instruments de musique numériques, et d'autant plus sa branche dédiée au prototypage, possède ses problématiques propres dont il faut avoir conscience pour effectuer un travail de recherche correct dans cet univers.

Parmi ces problématiques, le travail rapporté dans ce document prend le parti de s'attaquer à un tout nouveau champ applicatif pour le Probatio : la génération intelligente. L'idée est ainsi de fournir une toute nouvelle option, une nouvelle voie de possibilités, à cette boîte à outils. En réalité, cette nouvelle fonctionnalité se pose comme une pièce clé permettant, certes d'être utilisée comme un support à l'improvisation et aux tests, mais également de lutter contre une

grande source de perte de temps dans le domaine de la musique : l'installation. En effet, dans le cadre d'une utilisation combinée du Probatio et d'un instrument plus « classique », le temps d'installation explose. Or dans une démarche de premier pas dans le prototypage, ce type de perte de temps peut être extrêmement incommodante, voir rédhibitoire. La volonté insufflée dans l'implémentation de ce projet est donc de pouvoir substitué l'instrument « classique » par le générateur, et de cette manière ne pas avoir à encaisser le coût horaire de son installation.

Ajouté à cela, le générateur se veut intelligent pour pouvoir proposer une structure musicale intéressante et agréable à écouter. Il se veut également modulaire et embarqué de façon à pouvoir être utilisé avec simplicité et dans un champ de contextes variés. Le concept de ce générateur est plus précisément l'obtention d'un flux continu de commandes audio de type MIDI en sortie. L'aboutissement de cette démarche est ainsi de pouvoir piloter n'importe quelle synthèse audio à partir de ce flux généré. Ajouté à tout cela, le générateur se veut contrôlable par un utilisateur et robuste aux cas applicatifs moins généreux en performance.

Bien évidemment pour arriver à ce résultat, il est important de comprendre un nombre conséquent de concepts variés. Dans cette démarche, ce rapport propose de guider le lecteur intéressé à travers différentes sections permettant d'appréhender le fonctionnement du générateur. Le premier des chapitres représente ainsi le premier pas dans la compréhension du monde applicatif du générateur. Il consiste en une contextualisation et une présentation de l'état de l'art du domaine des instruments de musique numériques et de la génération musicale par Intelligence Artificielle. Le deuxième chapitre, lui, porte sur les outils nécessaires au bon fonctionnement du générateur. Troisièmement vient la présentation du projet dans lequel le générateur s'inscrit, le Probatio. Une fois passé, ces trois première partie et le contexte étant plus clair, la quatrième partie a pour tâche de présenter les apports personnels liés au travail développé dans ce rapport. Enfin, viennent les chapitres dédiés au développement du générateur et les perspectives d'avenir qui lui sont liés. Ce rapport pourra alors se clôturer sur une conclusion récapitulative.

Chapitre 1

Contextualisation & État de l'art

Pour une grande partie des personnes extérieures au monde de la musique, la perception de ce secteur reste cloisonnée aux productions musicales audibles sur les grandes chaînes de radio et autres plateformes de streaming musical. Mais dans la réalité, ce monde est bien plus vaste et cache tout un univers de technologie, de techniques et d'ingénierie se mêlant aux questions artistiques. Si certains pensent que la création d'instruments de musique est une tâche d'un autre temps, ces derniers sont dans le faux ! De nos jours, de nombreuses recherches sont consacrées à la création de nouveaux dispositifs musicaux. Le passage au numérique a par ailleurs lancé tout un nouveau pan de ces possibilités. C'est dans cette démarche que s'inscrit le *Input Devices and Music Interaction Laboratory* (IDMIL), laboratoire de recherche en technologie musicale de l'Université McGill, dans lequel a été développé le projet traité par ce rapport.

Ainsi et pour une meilleure compréhension de l'environnement dans lequel a évolué ce projet, ce premier chapitre a pour objectif de retracer un état de l'art des différentes facettes de ce milieu. Il se décompose en diverses sections. La première a pour rôle de présenter au lecteur l'état réel et actuel du monde musical, et tout particulièrement celui de la recherche et création de nouvelles interfaces pour l'expression musicale. Dans un deuxième temps, ce chapitre aborde la question des instruments de musique numériques. La manière dont ils sont perçus, conçus ou utilisés sont autant de questionnements qui sont abordés dans cette section. Enfin, la dernière section est, elle, consacrée au secteur plus spécifique de l'intelligence artificielle et traite également de sa place dans la création musicale.

1.1 Le monde de la musique aujourd'hui

De nos jours, l'industrie de la musique représente une part non négligeable de l'économie mondiale. Et ce n'est pas prêt de changer ! Chiffres à l'appui, la valeur totale de l'industrie du disque en 2021 serait de 25,9 milliards de dollars US avec une croissance mondiale moyenne de 18,5 pourcent (selon les données de l'IFPI, la Fédération Internationale de l'Industrie Phonographique)[14]. De plus, comme peut le montrer le graphe ci-dessous (fig. 1.1), cette tendance à l'évolution est loin d'être une exception, et se marque maintenant depuis plusieurs années, pour atteindre ainsi de véritables records ces derniers temps. A ceci, il peut également être ajouté la popularité actuelle des genres musicaux plus électroniques. En effet, suivant les données du CNM (Centre National de la Musique) [23] les genres musicaux s'exportent le mieux

en 2021 sont la dance/electro avec 40,8%, le rap avec 25,8% et la variété pop avec 19,7%. Or ce type de musique plus électronique est particulièrement friand d'innovations technologiques. Que ce soit à travers de nouvelles interfaces musicales physiques ou dématérialisées, de nouveaux synthétiseurs, ou encore de nouveaux instruments et effets, le marché de la production musicale est extrêmement vivant. Il représente ainsi un véritable vivier de développement technologique et un marché économique tout aussi vigoureux.

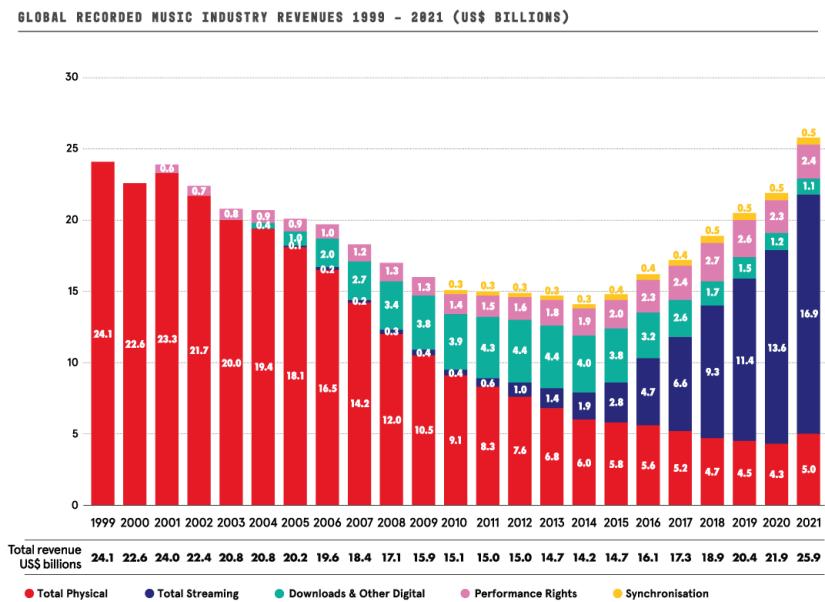


FIGURE 1.1 – Recettes de l’industrie mondiale de la musique enregistrées 1999 - 2021 (en milliards de dollars US).

Une autre trace de l’intérêt pour le développement de ce type de technologies est l’existence de rassemblements de spécialistes de ce domaine spécifique de recherche. Parmi ceux-ci, un des plus importants et reconnus est le NIME, la conférence internationale sur les nouvelles interfaces pour l’expression musicale. Cette dernière fêtait par ailleurs son vingtième anniversaire l’année passée, s’intéresser à ses statistiques peut donc être très intéressant ; et c’est précisément ce qu’ont pu faire S. Fasciani et J. Goode [7]. Évidemment, ce rapport n’a pas pour but de rentrer dans tous les détails de ce propos, et c’est pourquoi voici quelques éléments clés issues de cet article :

- NIME, ce sont des conférences mondialement reconnues prenant place à travers le monde comme le montre la figure 1.2 ;
- NIME, c’est un total de 1867 articles publiés avec un nombre de pages parues en globale hausse ;
- NIME, c’est un rassemblement de recherches explorant tout un éventail de sous-thèmes de l’expression musicale comme celui des capteurs, du mappage, de l’interprétation de gestes et encore bien d’autres (comme peut l’illustrer la figure 1.3¹).

1. La figure 1.3 est une illustration dans laquelle il est possible d’observer l’évolution des 10 termes uniques

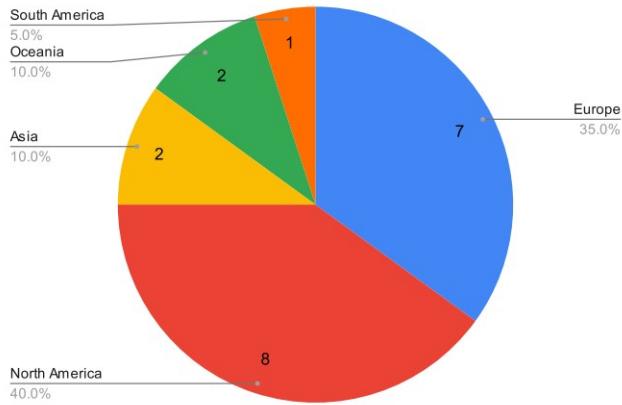


FIGURE 1.2 – Localisation des conférences NIME par continent.

- Enfin, NIME, c'est une communauté qui s'est développée et consolidée au cours de ses vingt ans d'existence.

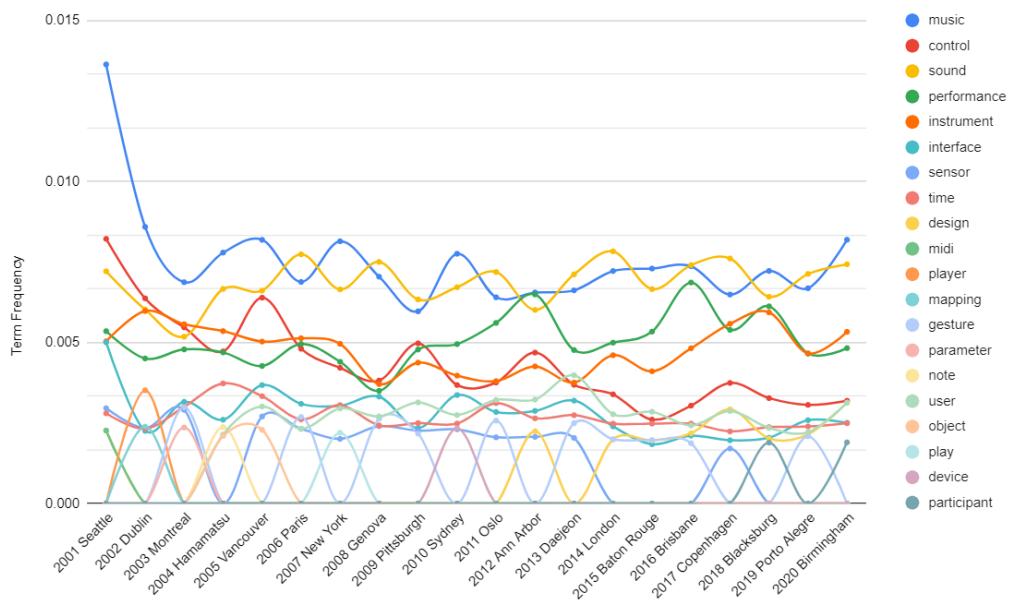


FIGURE 1.3 – Tendance des 10 termes uniques les plus courants dans le corps du texte des articles NIME publiés.

En bref, que ce soit par pure question économique ou par intérêt scientifique, il serait tout à fait erroné de considérer le monde du développement de nouveaux dispositifs musicaux comme une petite niche technologique sans grands enjeux. Il s'agit bien d'un domaine riche d'intérêt et où, de plus, l'ingénieur a tout à fait sa place.

les plus courants dans le corps du texte des articles publiés.

1.2 Instruments de musique numériques

La création d'instruments de musique n'est pas quelque chose de neuf. Néanmoins, comme souligné précédemment, le passage au numérique a ouvert un grand nombre de portes en termes de conception et a profondément bouleversé l'approche de la musique. Ainsi, il est intéressant et important de bien comprendre ce qu'est un instrument de musique numérique et ce qu'il permet de faire. En effet, la grande majorité des innovations récentes n'existent que grâce à cette numérisation. C'est pourquoi cette section a pour tenant et aboutissant la compréhension de cet aspect numérique.

1.2.1 Définition et potentiel

Une des grandes caractéristiques de l'instrument de musique numérique est sa capacité à se détacher des lois physiques. Il n'est plus question ici d'absolument se concentrer sur une source physique de vibrations, que ce soit par cordes, membranes ou circulation d'air, dont il est nécessaire de s'accommoder pour être capable de produire du son. De fait, la conception d'instrument par le passé se développait avant tout autour de la manière de moduler la solution physique sélectionnée et s'interrogeait sur la manière de rendre cette action de modulation accessible à un utilisateur. Cette logique fonctionnelle a ainsi mené à la création d'un grand nombre d'instruments. Néanmoins, ces créations ne placent souvent pas l'utilisateur au centre de la réflexion, car cette obligation de considération des lois physiques est et reste une contrainte conséquente de design, que ce soit à travers les possibilités musicales que peuvent offrir l'instrument, ou à travers son ergonomie. De cette manière, la suppression de cette contrainte rend le numérique très intéressant.

Dans la continuité, un des exemples grand public les plus parlants est celui de la guitare. En effet, la version électrique de cet instrument, bien que très proche en termes de conception de sa version acoustique, permet un grand nombre de nouvelles possibilités musicales. Avec un « simple » traitement du signal électrique de la guitare, il est ainsi possible de modifier de manière significative la sortie audio de l'instrument. Mais une fois passé cet aspect de traitement du signal électrique, il est également possible de numériser ce signal. A ce stade, de nouvelles possibilités musicales se dévoilent encore et permettent à la guitare de toucher à de grands concepts comme celui de la MAO² (Musique Assistée par Ordinateur). Ainsi, à travers cet exemple, l'instrument de musique numérique se présente en quelques sortes comme une évolution logique de la production musicale.

Une autre grande force de l'instrument numérique est la possibilité de travailler avec des séquences musicales. Un instrument « classique », c'est à dire produisant directement ses mélodies grâce à son design et sans l'aide de signaux électriques, crée par nature de la musique en réaction directe à son utilisation. Si l'utilisateur s'arrête, l'instrument de musique s'arrêtera également. La production est directe et spécifique aux paramètres de sa création. Si cette caractéristique n'est pas une mauvaise chose en elle-même, elle n'en demeure pas moins

2. La Musique assistée par Ordinateur désigne tout contexte dans lequel l'outil informatique est amené à intervenir dans un processus musical. Ces contextes peuvent aussi bien être des tâches de composition, comme d'interprétation. L'ordinateur est ainsi devenu un support très polyvalent et présent dans le domaine de la musique. La MAO est aujourd'hui un outil commun et se retrouve ainsi dans la très grande majorité des productions musicales contemporaines.

limitante. A contrario, les séquences musicales peuvent être manipulées, car elles permettent un enregistrement temporaire ou définitif d'une utilisation d'instrument. Dans cette optique, il est alors aisément de créer des boucles (comme il peut être fait en concert grâce à une pédale par exemple) ou d'éditer ultérieurement une séquence jouée. La musique produite par un instrument n'est alors plus spécifique à son moment de production et offre un ensemble de possibilités en séances d'enregistrement comme en live (de très beaux exemples de cela sont notamment les représentations d'artistes multi-instrumentistes jonglant d'instrument en instrument à l'aide de samples et de boucles).

Tout ceci étant dit, fondamentalement qu'est-ce qu'un instrument de musique numérique ? De manière concise, il s'agit donc d'une composition technologique typiquement faite :

1. d'une interface (ce avec quoi l'utilisateur interagit) utilisant certain type de technologie de capteurs ;
2. d'algorithmes de synthèse audio en temps réel fonctionnant sur un ordinateur ;
3. de connexions, définies artificiellement, entre les divers signaux d'entrée (provenant de l'utilisation de l'instrument par l'interprète) et les paramètres de synthèse (c'est à dire, définir que cette action engendre cette répercussion).

1.2.2 De nouvelles problématiques

Comme toutes nouveautés, cette technologie vient avec son lot de nouvelles problématiques également. Parmi celles-ci, on peut compter les questions d'expressivité, de retour sur actions (où on retrouve les solutions haptiques), de reprogrammabilité, de mappage dynamique et bien d'autres encore. Ce qu'il faut bien comprendre, c'est que contrairement à ce que l'on pourrait imaginer, le monde de la musique est extrêmement exigeant. Cette citation issue d'un papier de J. Malloch et M. Wanderley [17], portant sur la définition des instruments de musique numériques, laisse d'ailleurs très bien entendre ces difficultés :

In the grand scheme of things, there are three levels of design : standard spec, military spec and artist spec. Most significantly, I learned that the third, artist spec, was the hardest (and most important). If you could nail it, then everything else was easy.
(Buxton, 1997, p. 10)

Tout ceci étant dit, il est important de garder à l'esprit qu'utiliser un instrument est riche en contraintes. Un bon exemple est le temps de réaction qui se doit d'être très court en musique. Il peut également être assez difficile de faire percevoir le plein potentiel d'un instrument aux utilisateurs, notamment à travers toutes les possibilités de mappage. En effet, comme les connexions entre entrées et synthèse sont artificielles, tout est possible ; ce qui est fantastique dans le principe, mais est une très rapide source de confusion pour un utilisateur lambda. Ainsi, le juste milieu est souvent difficile à trouver dans les options données à l'utilisation, mais également dans la manière de les faire percevoir.

1.2.3 Utilisation et types de performance

Enfin, avant de conclure cette partie portant sur les instruments de musique numériques, un dernier point reste à aborder : leur utilisation. Et ce point n'est pas des moindres puisque

les objectifs d'utilisation conditionnent énormément le processus de conception. La figure ci-dessous (fig. 1.4) est issue de travaux précédents réalisés dans l'IDMIL [18] et est une adaptation du travail de J. Rasmussen au cas de la musique[19]. Cette représentation est un cadre théorique utile pour donner un sens aux diverses possibilités d'interaction dans la musique.

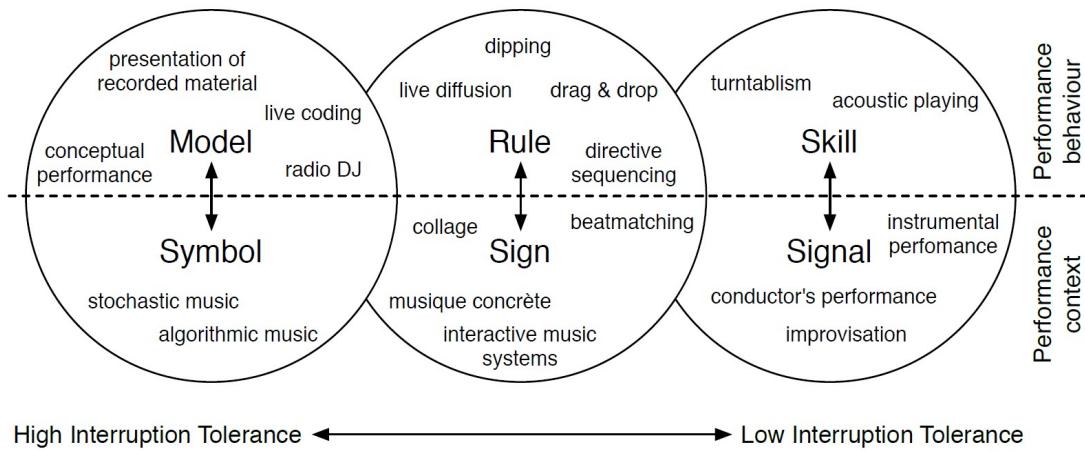


FIGURE 1.4 – Le cadre de Rasmussen appliqué à l'interaction musicale.

Cette figure est composée des trois comportements de performance identifiés. Ceux-ci sont retrouvables dans la moitié supérieure du graphe sous les appellations : basés sur des compétences (skill-based), des règles (rule-based) ou des modèles (model-based). Les contextes de performance sont représentés dans la moitié inférieure. À droite, les contextes exigent un couplage temporel étroit entre l'interprète et l'instrument, avec une faible tolérance aux interruptions. À gauche, les contextes ont un couplage beaucoup plus lâche. Globalement, au niveau des compétences, les interprètes interagissent avec leurs instruments dans une relation temporelle très étroite, mais il est alors nécessaire que l'interprète ait accumulé suffisamment d'expérience et de temps de pratique avec l'instrument. C'est le niveau qui se rapproche le plus, en termes d'expérience et d'exigences, de l'utilisation d'instruments acoustiques. Au niveau des règles, les interprètes interagissent avec les instruments/systèmes de manière plus détachée, en choisissant parmi un ensemble d'actions déjà apprises. L'utilisateur n'a alors le contrôle que d'un sous-ensemble de commandes mises à disposition, et une partie du processus est traitée par le système de manière indépendante. Enfin, au niveau des modèles, comme son nom l'indique, l'utilisateur travaille sur des modèles/systèmes prédéfinis qu'il va adapter à sa situation spécifique, de manière à construire une solution suivant ses désirs. Bref, un instrument de musique numérique peut prendre énormément de visages en fonction de l'application que l'on souhaite en faire. Il est donc primordial de correctement définir ses objectifs avant la conception.

1.2.4 Exemples d'instruments de musique numériques

Comme des exemples sont la plupart du temps plus parlants que de longs discours, voici deux projets d'instruments de musique numériques créés au sein de l'IDMIL. Le premier présenté ici est la Slapbox, un projet très récent et dont la dernière version a été produite il y

a quelques mois à peine. Le second à l'inverse est un projet dont les débuts datent maintenant d'il y a plus de dix ans et porte le nom de T-Stick.

La Slapbox

La Slapbox [22] (voir figure 1.5) est un instrument de percussion numérique autonome qui offre à la fois l'excitation et la modulation sur les mêmes surfaces d'interaction. Construit à l'aide de cartes Bela³, l'instrument traite les interactions gestuelles et produit des sons à l'aide de son propre moteur de synthèse. Il peut être pris en main et joué sans dispositifs externes ni configuration, ce qui est une caractéristique importante pour soutenir la longévité des instruments numériques. Les pads de batterie situés sur le dessus, le dos et les côtés de la boîte sont capables de détecter une pression continue ainsi que la position, et peuvent détecter les coups avec des performances similaires à celles des contrôleurs de batterie MIDI. Au total, l'instrument compte 4 pads de position et de pression pour une utilisation de type percussion, 2 petits pads de pression et un élément strié imitant un güiro. En outre, deux boutons de modulation situés sur le panneau supérieur permettent de contrôler la hauteur du son produit.



FIGURE 1.5 – La Slapbox

Le T-Stick

Le T-Stick [25] (voir figure 1.6) est un instrument de musique numérique conçu par Joseph Malloch et D. Andrew Stewart. En développement depuis 2006, il a une histoire relativement longue pour un DMI (Digital Musical Instrument), a de multiples versions, plusieurs interprètes experts, et même un répertoire associé. Plus de 20 copies ont été construites sans intention d'utilisation commerciale. Néanmoins, il a été adopté par des interprètes et compositeurs experts dans le cadre de leur pratique musicale. Il a fait l'objet de dizaines d'apparitions publiques dans des pays tels que le Canada, les États-Unis, le Brésil, l'Italie, la Norvège et le Portugal. Pour ce qui est de son fonctionnement, de manière concise, le T-Stick est un instrument en forme de tube droit qui peut détecter où et dans quelle mesure il est touché, tapé, tordu, incliné, pressé et secoué ; le tout déclenchant différentes conséquences sur la synthèse du son.

3. Bela est une plateforme informatique embarquée permettant de créer des projets interactifs très réactifs. Basé sur la famille d'ordinateurs embarqués open-source BeagleBone, Bela combine la puissance de traitement d'un ordinateur embarqué avec la précision du timing et la connectivité d'un microcontrôleur.



FIGURE 1.6 – Le T-Stick

1.2.5 Récapitulatif

En conclusion, les instruments de musique numériques représentent le nouveau visage de la musique et permettent l'ouverture d'un grand champ de possibilités. Néanmoins, leur capacité fondamentale à se détacher des règles physiques de production du son et à ainsi pouvoir « tout faire » peut très vite devenir une source de confusion pour un utilisateur. C'est pourquoi la réflexion autour de la création d'instruments doit être complète et claire vis-à-vis de ses objectifs. Mais ce qui est certain, c'est que ces nouvelles voies musicales sont captivantes et permettent des choses jusqu'alors inimaginables, comme de l'interprétation haut-niveau. Vous désirez émettre de la musique en fonction du mouvement d'un danseur ? Vous pouvez parfaitement concevoir un instrument spécifique. La seule vraie limite est votre imagination !

1.3 Intelligence Artificielle et musique

Contrairement à l'idée que se font beaucoup de personnes, l'Intelligence Artificielle n'est pas une technologie si récente. Si cette dernière est réapparue au centre des attentions, c'est aussi énormément grâce à l'évolution du matériel informatique et à la puissance de calcul l'accompagnant. Ceci faisant (et accompagnée de quelques autres facteurs extérieurs comme les notions de Big Data), l'IA est revenue sur le devant de la scène et connaît depuis quelques années un engouement sans précédent, aussi bien venant du monde scientifique que du grand public. Dans une telle atmosphère d'intérêt, l'IA a pu énormément évoluer et dans un grand nombre de domaines différents. Il n'est ainsi pas étonnant de retrouver certaines de ces avancées également dans le monde de la musique. Que ce soit sous une vue fantasmée, digne des meilleurs romans de science fiction, ou de manière plus terre à terre, de nombreux chercheurs ont émis le désir de créer des générateurs musicaux utilisant cette précieuse IA. Ce travail s'inscrit également dans cette voie. Néanmoins, l'IA est par excellence une technologie faisant naître de nombreuses idées préconçues dans l'esprit du public. C'est pourquoi dans une voie d'exploration

scientifique de la question, une remise en contexte est nécessaire. Cette section reviendra tout d'abord sur les bases techniques de l'intelligence artificielle. Par la suite, l'attention sera portée sur le cas plus spécifique de la génération musicale et finalement sur les problématiques d'évaluation qui accompagne la question de la génération musicale.

1.3.1 L'Intelligence Artificielle : les fondamentaux

Comme peut le laisser deviner son nom, l'intelligence artificielle trouve ses fondements dans l'utilisation de réseaux de neurones. De cette utilisation, on tire deux types de réseaux : les profonds et les non profonds. L'élément permettant de les distinguer n'est en réalité que le nombre de couches qu'il est possible d'y trouver. En prenant l'image ci-dessous (fig. 1.7), il est possible de comprendre que, plus précisément, c'est le nombre de couches de neurones cachés qui fait la différence, c'est à dire les neurones situés entre les couches d'entrée et de sortie. Bien évidemment, les deux types de réseaux n'offrent pas les mêmes possibilités. C'est pourquoi, les réseaux profonds sont au centre de la plupart des recherches modernes.

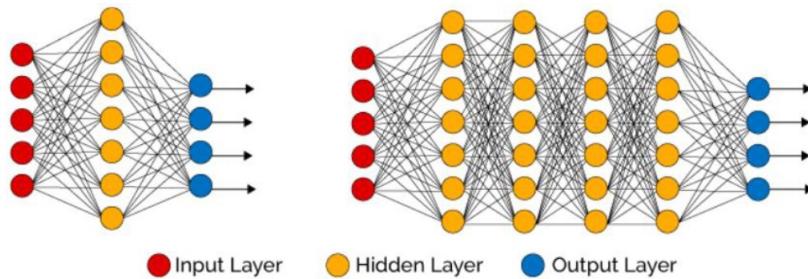


FIGURE 1.7 – Types de réseaux de neurones : non profond à gauche et profond à droite.

Cette différence dans les applications possibles repose sur ce que représente réellement un « neurone » en IA. Pour cela, l'idéal est de repartir du cas fondamental : le perceptron. Cette unité est le schéma le plus simple à concevoir et peut être vue comme un classificateur élémentaire. Partant de cette illustration (fig. 1.8), le perceptron peut être considéré comme une unité prenant en entrée un certain nombre de variables. Ces variables recevront chacune un certain poids qui les multipliera, leur accordant en quelques sortes une importance relative. Le perceptron possède également une autre valeur appelée biais qui est additionné aux variables multipliées par leur poids. L'équation d'un perceptron peut ainsi se noter (avec d , le nombre de variables d'entrée, B le biais et X , le vecteur de variables d'entrée) :

$$z = \sum_{i=1}^d (W_i * X_i) + B \quad (1.1)$$

Néanmoins, le perceptron ne s'arrête pas là, car comme indiqué sur la figure 1.8, ce premier résultat est ensuite placé dans une fonction dite d'activation. Cette étape a pour but de placer une utilité, un sens sur la valeur z obtenue précédemment. De manière plus générale, la fonction d'activation choisie dépend du cas applicatif dans lequel le développeur se trouve. Voici par exemple deux fonctions d'activation très répandues (fig. 1.9) :

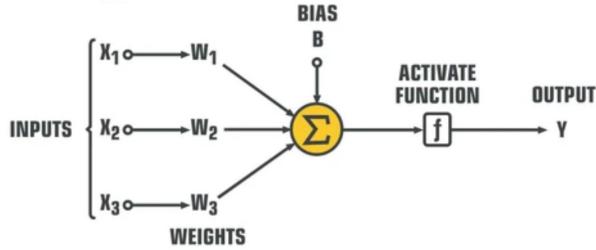


FIGURE 1.8 – Le perceptron.

La fonction sigmoïde (à gauche) est par exemple très indiquée pour créer des probabilités comme sortie. Imaginons que l'objectif du perceptron soit de reconnaître un type de fruits en fonction de paramètres d'entrée. Le perceptron à travers l'équation 1.1 va dessiner une frontière de décision (là où $z = 0$) influencée par les poids et le biais. La sortie de cette équation indiquera alors où se trouve le fruit lié aux données mises en entrée par rapport à cette frontière ; cette position identifiant par la même occasion le type de ce fruit. La fonction sigmoïde, elle, indiquera alors à quel point cette prédiction est probable. Ainsi au plus loin de la frontière (représentée par la valeur 0 en abscisse) sont les données du fruit, au plus la prédiction a de probabilités d'être correcte. En d'autres termes, si les données sont loin des frontières de décision, c'est qu'il y a moins de confusion/d'hésitations, donc la décision a plus de chance d'être exacte. Dans un autre registre, la fonction ReLU (à droite) est typiquement une des plus utilisées dans le domaine de l'IA, et tout particulièrement dans les réseaux caractérisés de convolutionnels. Comme il peut être remarqué, cette fonction est appréciée, car elle permet la mise à zéro de toutes valeurs négatives. Cette caractéristique est souvent intéressante, ce qui explique en partie sa forte présence dans le domaine du Deep Learning. Quoiqu'il en soit, la fonction d'activation décrit en partie le fonctionnement d'un neurone et sa valeur de sortie constitue la véritable sortie du perceptron.

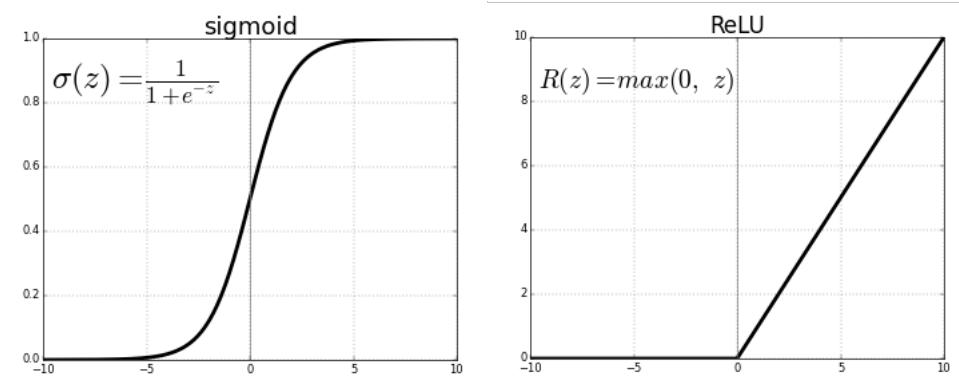


FIGURE 1.9 – Exemples de fonctions d'activation

Tout ceci ne représente qu'une infime partie du potentiel de l'intelligence artificielle, mais le cas du perceptron permet d'appréhender la situation. En effet, au plus il est possible d'utiliser la sortie de neurones comme l'entrée d'autres, au plus il est possible de créer d'interactions,

au plus les connexions peuvent devenir complexes et au plus les réseaux ont de potentiels d'action.

Toutefois, les bases de l'IA ne s'arrêtent pas là, car il manque encore une de ses composantes les plus importantes : sa capacité d'amélioration. En effet, les poids qui ont été mentionnés plus tôt ont la capacité de modifier leur valeur durant la période d'entraînement. C'est cet affinage des poids qui donne réellement sa force à un réseau de neurones. Pour comprendre ce processus la figure 1.10 ci-dessous est un bon appui.

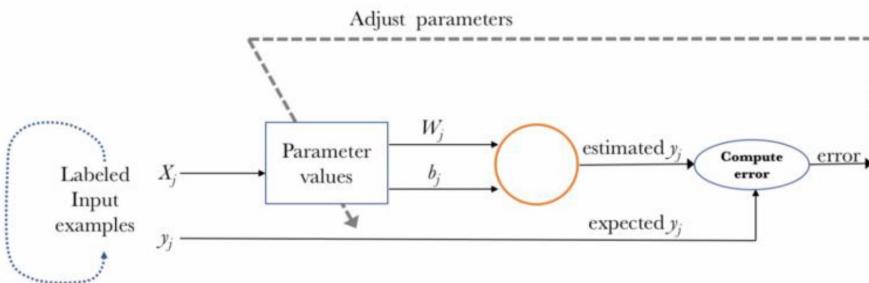


FIGURE 1.10 – Processus d'entraînement et ajustement des poids.

L'idée derrière ce processus est de présenter un nombre de fois défini des données connues à l'entrée d'un réseau. Pour chacune de ces données, le réseau va alors créer une estimation. Celle-ci s'apparente au résultat attendu. Toutefois, comme les données présentées au réseau sont des données connues, l'utilisateur connaît également le résultat réel et attendu pour ces données d'entrée. Ainsi, chacune des sorties estimées du réseau est comparée au résultat attendu pour les données qui lui sont relatives. Cette comparaison donne alors une perte en fonction de l'erreur entre le produit et l'attendu, et cette perte est à son tour utilisée en vue d'ajuster les poids du réseau. On appelle cette remontée d'informations dans le réseau, nécessaire à l'affinage des poids, la backpropagation. Dans les faits, cette remontée est un peu plus complexe, car le réseau emploie également des stratégies vis-à-vis de l'affinage en fonction des pertes calculées. Ces stratégies s'apparentent à des méthodes d'optimisation et on y retrouve des notions telles que la descente du gradient. L'entraînement en lui-même mène également à d'autres problèmes d'optimisation comme celui du surentraînement, où le modèle devient trop spécifique à ses données d'entraînement. De cette manière, chaque réseau possède son architecture, sa manière de s'entraîner, ses données d'entraînement et sous-tend à un ensemble de questions d'optimisation que cette section, présentant les bases de l'IA, n'a pas vocation à aborder.

1.3.2 Deep Music Generation

Maintenant que le décor a été planté, cette section va approfondir le domaine plus spécifique de la génération de musique par réseaux profonds. Comme expliqué dans la partie précédente, les réseaux profonds ont de très nombreuses applications de par leur nature. De manière à faciliter la navigation dans toutes ces possibilités, les réseaux profonds sont souvent rassemblés en sous-familles en fonction de leur approche, de leur manière d'utiliser leurs couches ou encore de leur façon de réaliser leurs connexions. Pour citer quelques exemples, on peut retrouver la

famille des GAN (Generative Adversarial Network), des auto encodeurs, des transformateurs (plus couramment appelés « transformers ») et bien d'autres encore. Néanmoins, un type d'application ne désigne pas spécialement un type de réseau. C'est pourquoi, il est souvent intéressant de s'intéresser à ce qui a déjà été fait dans le domaine, de manière à apprendre des autres. Le domaine de la musique ne fait pas exception à cette règle, et il est ainsi possible de retrouver au fil des années bien des approches différentes de la question de la génération musicale. Ainsi, cette partie de l'état de l'art a pour objectif de se renseigner sur ce qui a déjà pu être fait par le passé, et par la même occasion déterminer de quelle(s) approche(s) il est plus stratégique de se rapprocher.

Processus de production musicale

Avant toute chose, il faut comprendre que le processus de production musicale se compose de trois parties distinctes possédant chacune ses problématiques. La première est la génération de partitions, c'est à dire la définition du script sur lequel se baserait un musicien pour jouer. Typiquement, on y retrouve la sélection des notes et des rythmes. La partie suivante est la génération de performances. Cette phase a, comme son nom l'indique, pour objectif l'ajout de caractéristiques de performance aux partitions. Dans le cas d'un musicien, cela s'apparente à l'interprétation du joueur et des intentions qu'il met dans son jeu. Enfin, vient la partie de la génération audio qui a pour but la conversion des partitions ayant reçues des caractéristiques de performance en audio en leur attribuant un timbre (ou simplement la génération directe de musique au format audio). A l'image de la figure 1.11 issue de [15], chacune de ces parties sont nécessaires. De manière à produire des séquences musicales, il est ainsi obligatoire de se préoccuper de chacune d'elles.

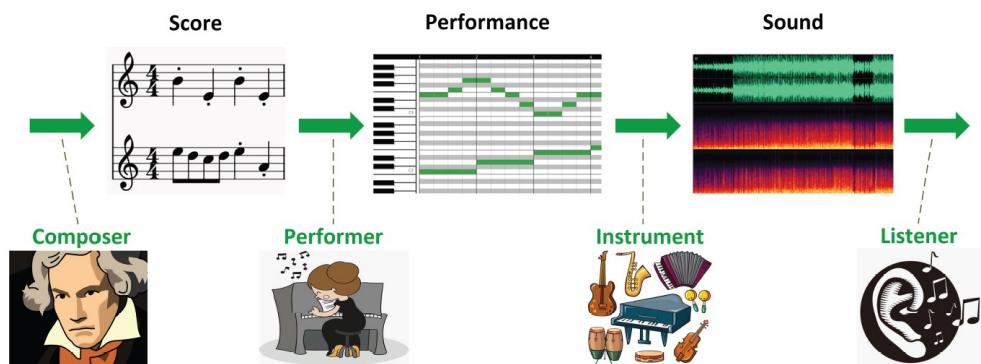


FIGURE 1.11 – Processus de génération musicale.

Positionnement du problème (IA & Probatio)

De manière à orienter d'avantage cet état de l'art, cette section se consacre avant tout sur les types de génération qui sont touchés par le générateur implémenté dans ce rapport⁴. Ainsi par choix, la partie de génération audio sera traitée de manière classique. Dans le cadre applicatif de ce rapport, l'utilisation de techniques de synthèse conventionnelles est en effet

4. Pour plus de détails sur les choix effectués et le développement du générateur, le lecteur intéressé trouvera de quoi l'en informer dans les chapitres portant sur les apports personnels et le développement du générateur.

plus indiquée. Partant de ce constat, l'objectif est donc de produire comme sortie un flux de commandes audio (typiquement un flux MIDI) à placer en entrée d'un synthétiseur. Trois options sont donc possibles :

1. Uniquement utiliser un générateur de partitions (c'est à dire ne pas ajouter de caractéristiques de performance sur les partitions).
2. Utiliser deux unités pour la génération : une pour la partie partition et une seconde pour l'ajout des caractéristiques de performance.
3. Utiliser un réseau capable de générer à la fois la partition et les caractéristiques de performance.

Pour les traiter dans l'ordre, la première possibilité (option 1), bien que pouvant être fonctionnelle, engendrera une sortie audio monotone, peu naturelle et sûrement assez désagréable à écouter. C'est l'ajout de caractéristiques de performance qui donne en bonne partie à la musique son côté vivant et captivant. Il est donc possible de se passer de la seconde étape du processus de génération, mais à un lourd prix en termes de qualité finale. Par conséquent, si cette possibilité est évitable, il est préférable de l'écartier. La deuxième possibilité (option 2) représente quant à elle, l'approche plus classique, puisque traitant les parties du processus de génération de manière indépendante. Néanmoins, cette vision engendrera par défaut une complexité totale plus grande. Il faudra dans cette configuration deux réseaux fonctionnels capables de tourner en parallèle pour la production du flux désiré. Cette complexité s'accompagne également d'une fort probable consommation de ressources plus importante. Enfin, la dernière possibilité (option 3) représente la solution brute aux problèmes rencontrés. L'idée est donc dans ce cas de travailler avec un compositeur de performances expressives. Cette idée bien que complexe n'est pas impossible, le système de génération a « juste » besoin de connaître dès le départ un modèle de langage expressif. L'ensemble de ces possibilités évaluées et étudiées plus en détails, l'option 3 semble la plus indiquée dans le cas applicatif de ce rapport.

Générateurs de performances expressives : Un domaine de recherche en constante évolution

Cette préférence pour la génération de performances expressives se justifie majoritairement par sa plus grande facilité d'utilisation dans un contexte embarqué et/ou d'improvisation. Toutefois bien que d'un intérêt certain, cette sous-branche de la génération musicale a pour l'instant bien moins attiré l'attention que la génération de partitions. Comme a pu le relever l'analyse du domaine de la génération musicale de S. Ji, J. Luo et X. Yang [15], les modèles de génération de performances expressives sont globalement moins complexes que ceux consacrés aux partitions, car moins de chercheurs portent leur attention sur la question. Ainsi, cette sous-branche a encore beaucoup de possibilités à expérimenter et de futures recherches pourraient bien la rendre très intéressante à l'avenir. Pour illustrer cette différence de traitement, le lecteur intéressé pourra trouver en annexe B différentes lignes du temps traitant des différentes voies de génération.

Malgré le fait que cette voie de génération ait été moins explorée, elle n'en demeure pas moins intéressante. Néanmoins, il est à concéder que cet intérêt a presqu'exclusivement été porté au piano ; les performances de ce type étant bien plus simples à quantifier. Le seule

autre type de performance ayant été sujet à de multiples études de ce domaine a pour l'instant été celui des percussions [9] [8]. Ce type de génération sort toutefois du cadre de ce rapport.

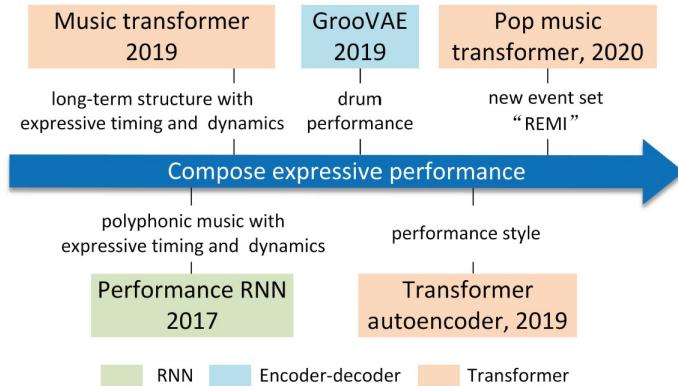


FIGURE 1.12 – Chronologie de la composition de performances expressives. Les couleurs représentent les différentes architectures de modèles.

Dans le cadre des générateurs basés sur des performances de piano, le projet Performance RNN [21], datant de 2017, est le vrai investigateur de l'approche actuelle. Par la même occasion, il est aussi l'initiateur de l'intérêt porté à cette sous-branche de la génération. Cette première approche utilisait un modèle de RNN⁵ (réseau de neurones récurrents) basé sur LSTM⁶ permettant la composition de séquences musicales polyphoniques (plusieurs notes jouables en même temps par un unique instrument) dotées d'une dynamique expressive. Malheureusement, bien que fonctionnel, les productions de Performance RNN manquaient de cohérence globale. Basé sur ce travail vient quelques années plus tard Music Transformer [11]. Ce projet de 2019 reprend alors le modèle de langage expressif de Performance RNN pour le réexploiter dans un tout autre type de modèle génératif. Ce nouveau type de modèle est alors développé autour de l'utilisation de transformateurs. Ce remaniement se montre alors des plus appréciables puisqu'il élève grandement le niveau de qualité des productions, que ce soit dans leur contenu à proprement dit, ou dans leur cohérence. Ce type de réseau utilisant les transformateurs, suite à la parution de l'article [11], devient alors la référence pour ce qui est de la génération musicale de performances expressives. Suite à l'apparition de cette approche vont apparaître en très peu de temps de nouveaux travaux abordant ces transformateurs. Comme il peut être vu sur la ligne du temps ci-dessus (fig. 1.12), deux travaux majeurs ont été publiés : « Transformer autoencoder » [2] et « Pop music transformer » [12]. Le premier, par un nouveau type de traitement des données d'entrée, permet un encodage dans le temps de ces entrées, afin d'obtenir une représentation globale du style d'une performance donnée. Ses auteurs

5. Un réseau neuronal récurrent (RNN) est un type de réseau neuronal qui utilise des données séquentielles ou des données de séries temporelles. Ce type de réseau se distingue des autres de par son utilisation des données des entrées précédentes pour influencer ses sorties. Il est donc tout indiqué dans des configurations où les éléments/événements passés doivent affecter le résultat présent.

6. Un LSTM ou Long Short-Term Memory est un type particulier de réseaux de neurones récurrents. De manière concise, les réseaux LSTM possèdent des cellules internes d'état contextuel qui agissent comme des cellules de mémoire à long terme ou à court terme. La sortie de ce type de réseaux est alors modulée par l'état de ces cellules. Il s'agit d'une propriété très importante lorsqu'il est nécessaire que la prédiction du réseau neuronal dépende du contexte historique des entrées, plutôt que de la toute dernière entrée.

ont montré que leur approche permettait un meilleur contrôle des aspects séparés du style et de la mélodie de la performance. Les résultats expérimentaux de cette recherche ont su démontrer que leur modèle conditionnel permettait de générer une performance similaire au style d'entrée, et de générer un accompagnement qui se conforme au style de performance donné pour la mélodie. Néanmoins, « Transformer autoencoder », à travers ses résultats, ne montre pas de comparaison claire en termes de qualité musicale de sortie, notamment par rapport à Music Transformer[11] auquel il se réfère pour la construction de son modèle. Si cette étude présente beaucoup de tests de similarité concluants, il n'est ainsi cependant pas possible d'avancer qu'elle représente un apport significatif en termes de qualité de production. « Pop music transformer », quant à lui, propose un générateur orienté dans un style de musique plus populaire et prend comme fondement les articles [11] et [3]. La grande différence de cette recherche repose en la création d'une nouvelle caractérisation d'ensemble d'événements nommée « REMI ». Cette caractérisation « REMI » est en quelques sortes une amélioration des caractérisations précédemment utilisées dites « MIDI-like », c'est à dire reprenant l'approche du protocole MIDI (voire chapitre 2). Ce nouvel ensemble d'événements se voit comme un meilleur contexte métrique pour la schématisation rythmique de la musique, permettant notamment une meilleure appréhension des changements locaux de tempo. Cet étude présente également une progression des accords plus contrôlable grâce à la définition de structures harmoniques. Dans un contexte davantage multi-instrumental⁷, Pop music transformer permettrait une meilleure coordination entre pistes (telles que piano, basse, batterie, etc.). En fin de compte, cette étude permettrait une production conditionnée offrant une structure rythmique plus raisonnable que Music transformer [11]. Les auteurs de [12] voient également leurs résultats qualitatifs comme une preuve qu'il est bénéfique d'intégrer des connaissances humaines préalables sur la musique en incluant des composants de techniques de recherche d'informations musicales, comme l'estimation des temps morts et la reconnaissance des accords. Néanmoins, ces auteurs voient leur modèle comme pouvant encore évoluer de manière à incorporer d'autres informations de haut niveau comme le groove ou l'émotivité musicale. Enfin, ils aimeraient également être capable de générer d'autres instruments et étudier d'autres architectures de modèles.

N.B. : De manière à éviter la redondance des informations, le lecteur intéressé trouvera de plus amples détails sur les transformateurs et leurs réseaux associés dans le chapitre 5.

1.3.3 Les problématiques de l'évaluation

Comme il vient d'être montré, l'IA a connu énormément d'évolution dans le domaine de la musique. Néanmoins, certains problèmes ont quant à eux réussi à persister au fil des années ; ces problèmes, ce sont ceux de l'évaluation. En effet, un constat simple peut être fait : la musique est quelque chose d'extrêmement difficile à évaluer, et ce que ce soit pour une machine ou un humain. La simple définition de ce qu'est la musicalité est une tâche des plus ardues (voir annexe C), créer des métriques autour de cette notion est donc d'autant plus complexe.

Une preuve simple de la non résolution de ces problématiques est l'existence d'études récentes spécifiques à la question et l'analyse de leurs résultats. Parmi ces recherches, il est intéressant de se pencher sur cette étude issue de l'université d'Athènes [4]. L'étude en

7. Il est d'usage de parler de multi-track/multi-instruments lors d'une situation polyphonique multipiste. En d'autres termes, cela désigne généralement une composition de plusieurs pistes/instruments ayant leur propre dynamique temporelle. De plus, ces pistes/instruments sont interdépendants en termes de temps.

question, en plus d'avoir l'avantage d'être récente (2021), analyse très bien la situation. La génération de musique est un domaine de recherche très actif où l'évaluation est une tâche restée très difficile. Suite à cela, une grande partie des résultats proposés par les études dans ce secteur de recherche reposent sur l'avis de testeurs humains. Cette façon d'évaluer n'est certes pas à jeter, mais son principe la rend intrinsèquement difficile à étendre sur de vastes comparaisons. De fait, en plus d'être gourmandes en ressources, les différentes enquêtes auprès des utilisateurs ne peuvent pas être directement comparées entre elles, ce qui rend impossible ou incomplète la comparaison de différentes approches pour la génération de musique à partir de la littérature. Inévitablement, ces difficultés entravent le progrès. De ces constats viennent donc des tentatives de création de métriques plus ou moins complexes, qui sont par défaut elles-mêmes confrontées à l'oreille de testeurs humains. Malheureusement comme beaucoup d'autres études, cette dernière conclue sur une non atteinte des objectifs. De manière résumée, leur approche semble avoir un potentiel qui n'a pas été encore pleinement découvert, car bien que plus complexe, leur système n'apparaît pas comme un apport significatif.

Si ce constat n'est que le résultat d'une unique étude, il est forcé de constater que dans l'ensemble, personne n'est vraiment capable de mieux. En effet, si le fait que l'humain ait lui-même de grosses difficultés à définir le concept de musicalité (voir annexe C) n'arrange rien, il est à constater qu'un climat d'aveu de faiblesse est tristement en train de se dresser vis-à-vis de la question. Ainsi tenant compte de nombreuses études, la situation peut globalement se résumer en trois positions possibles : la reconnaissance d'une certaine impuissance pour ce qui est de l'évaluation quantitative, une utilisation de métriques d'analyse relativement basiques, ou un bricolage d'une métrique faite maison. Ainsi, si la première possibilité a le mérite de se montrer agréable pour son honnêteté, elle ne fait pas avancer la question. Pour ce qui est de la dernière, par défaut, elle varie suivant les cas. Malheureusement le problème de telles variables est qu'elles ont souvent tendance à aller dans le sens des résultats de l'article les proposant. Ceci posant évidemment quelques questions concernant l'objectivité de la métrique en question, d'autant plus qu'il est souvent difficile, voire impossible, de comparer de tels résultats avec d'autres articles. De cette manière, bien que la démarche soit intéressante, elle n'aboutit que rarement sur quelques choses d'objectivement pertinents et exploitables. Tout ceci faisant, la solution de repli que représente l'utilisation de métriques musicales de plus bas niveau est souvent la meilleure, bien que n'exprimant nullement le concept de musicalité.

Toutefois dans ce climat peu favorable, un article a su prendre position [28]. Celui-ci n'essaie pas de trouver des métriques haut niveau, mais au contraire se base sur des notions plus simples de manière à fournir un ensemble de métriques objectives simples, fondées sur des notions musicales. L'objectif derrière cette approche est, de cette façon, la mise à disposition d'un outil permettant des évaluations objectives et reproductibles. Ainsi, bien qu'utilisant des fondements plus simples que d'autres approches, cette étude a le mérite de présenter une solution viable. De plus, un certain nombre de recherches comme par exemple [4], [13] ou encore [20] font maintenant référence à cet article dans leurs résultats ; ce qui est sûrement la meilleure preuve de l'utilité de cette étude. Ces métriques semblant intéressantes, les résultats de ce rapport font également référence à [4]⁸. En complément d'information, la figure 1.13

8. Il existe bien-sûr d'autres études visant à fournir des métriques plus bas niveau dont ce document ne fait pas référence. Néanmoins, ces autres possibilités semblent moins bonnes ou moins utilisées. C'est pourquoi ce document ne les mentionne pas.

illustre le principe des métriques en question. Ces dernières se décomposent en une exploitation de données inter-set et intra-set basées elles-mêmes sur l'utilisation de caractéristiques de hauteur des notes et de rythmique. Ces métriques représentent grâce à cette composition une bonne source d'informations, malgré l'utilisation de caractéristiques bas niveau.

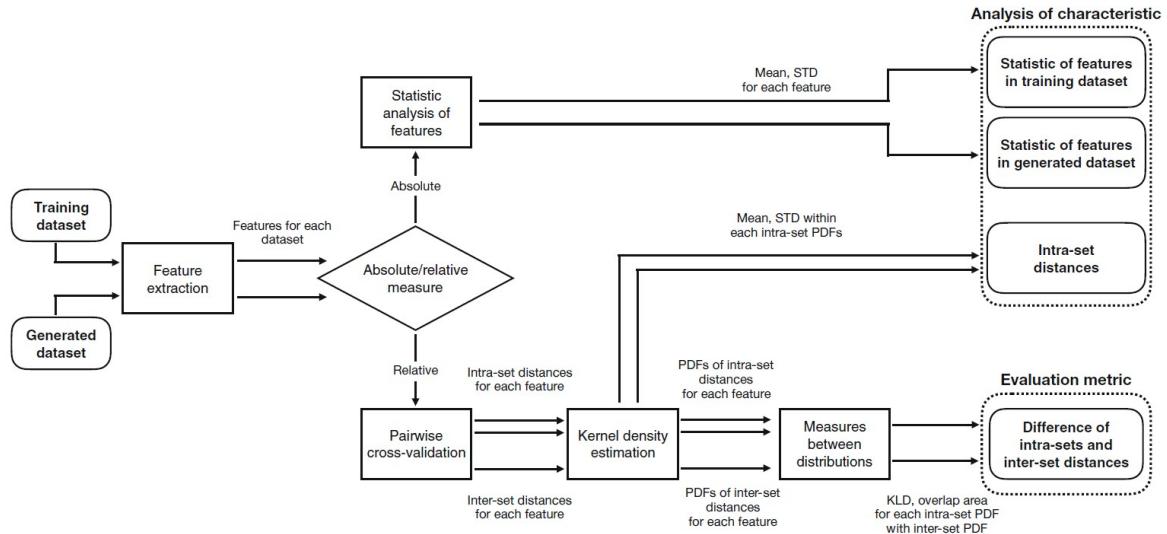


FIGURE 1.13 – Flux de travail général de la méthode proposée dans [28].

Enfin, un autre paramètre rend l'approche de l'évaluation plus complexe encore : tous les travaux ne cherchent pas à démontrer les mêmes résultats. En effet, dans la génération musicale il est d'usage de séparer la génération dite monophonique, de celle dite polyphonique. La première se consacre à la création d'un seul canal, d'une seule source, c'est-à-dire qu'il n'est question que de créer la prestation d'un unique instrument, et de plus un instrument ne jouant que note par note. La seconde, elle, vise la création de plusieurs voix et par conséquent soit d'un instrument capable de jouer plusieurs notes à la fois, soit de plusieurs instruments, dont potentiellement des percussions. Dans de telles configurations de différence, les métriques se montrent forcément elles aussi différentes, ce qui engendre à nouveau la création de nouvelles métriques. Malencontreusement, la polyphonie ne fait que complexifier la situation précédemment décrite, et dans cette suite logique, la plupart des études dans le cas polyphonique présentent sans unanimité différentes métriques musicales de bas niveau comme dans [6], [5] ou encore [27].

En résumé, le monde de l'évaluation des générateurs musicaux repose dans un chaos manquant cruellement de normes ; ce désordre trouvant son origine dans l'immense difficulté que représente la quantification de la musicalité. Dans cette logique, aucune étude n'est obligée de présenter ses résultats sous un certain format. Si de prime abord cela ne semble pas si problématique, l'absence de métrique commune rend la navigation dans les diverses approches technologiques de l'état de l'art anormalement compliquée et chronophage pour tout chercheur s'intéressant à la question. Ainsi, la problématique de l'évaluation est pour l'instant un problème aux allures insolubles engendrant son lot de conséquences néfastes, dont il est nécessaire de s'accommoder.

Chapitre 2

Outils

Tout système, aussi bon soit il, se base sur l'utilisation de divers outils ; ce projet ne fait pas exception à la règle. Dans cette logique, ce chapitre a pour intention d'informer sur quelques-uns de ces outils clés permettant l'existence de ce projet.

2.1 MIDI

Le MIDI est à la fois un format de fichier et un protocole de communication. De son nom complet « Musical Instrument Digital Interface », cet outil est fondamental dans ce projet puisqu'il y est présent sous ses deux aspects.

Pour bien comprendre l'utilité du MIDI, il faut faire la distinction entre le domaine symbolique et le domaine audio. Ceux-ci peuvent être sélectionnés, l'un comme l'autre, pour travailler la musique, mais ils ne représentent pas le même type de traitement. L'approche du domaine symbolique est composé de variables discrètes caractérisant au fil du temps certaines valeurs. En résumé, il y est davantage question de décrire un évènement que de le copier. Par exemple, avec le MIDI (qui fait donc partie du domaine symbolique), il est possible de décrire le début d'une note à un moment précis, avec une hauteur de note précise et avec une certaine vitesse. Le domaine audio quant à lui travaille avec les formes d'onde de la musique. Cette approche est donc plus proche de la réalité. En effet, dans la plupart des cas le timing et la dynamique des notes jouées par les musiciens ne sont pas parfaitement conformes à la partition. Ces petites permissions, volontaires ou non, sont la conséquence de l'interprétation du musicien, et ce qui pourrait être perçues comme de petites erreurs sont (dans la plupart des cas) en réalité un apport d'émotivité et de vie dans la performance musicale. De plus, à quelques pertes près, l'utilisation des formes d'onde conserve plus d'informations liées à la musique et possède de riches détails acoustiques, tels que le timbre, l'articulation, etc. Néanmoins, l'approche audio a aussi des limitations qui la rendent moins modulaire dans certains aspects de la musique numérique où il peut être plus aisément de travailler avec des descriptions d'évènement. Ainsi, on retrouve aujourd'hui énormément de contrôleur MIDI dans le monde musical. Ces appareils de diverses allures (mais souvent sous forme de piano) présentent en effet une très grande facilité d'utilisation. Une fois branché à un outil de synthèse, les données qu'ils produisent peuvent être entrées dans n'importe quel instrument virtuel ou encore utilisées dans un logiciel de production musicale tel qu'Ableton. En dernier argument, dans un contexte de génération par intelligence artificielle, il est appréciable de travailler avec des données plus quantifiées.

Le MIDI est donc un choix stratégique appréciable et commun. Il est de ce fait une norme industrielle qui décrit le protocole d'interopérabilité entre divers instruments, logiciels et appareils électroniques et il est ainsi utilisé pour connecter des produits de différentes entreprises, notamment des instruments numériques, des ordinateurs, des tablettes, etc. Le MIDI est donc très pratique pour sa facilité de mise en connexion. Dans la pratique, le protocole transporte les données de performance et des informations de données de contrôle en temps réel. Cependant dans les faits, ce sont avant tout les évènements *Note-On* et *Note-Off* (début et fin de notes) qui sont cruciaux. De plus, la représentation MIDI étant à la fois simple et efficace, beaucoup de systèmes emploient des descripteurs d'évènements « MIDI-like » comme représentation musicale. De manière résumée, ces systèmes emploient les événements Note-on, Note-off et la hauteur du son pour le codage de la partition et ajoutent davantage d'éléments descriptifs (vitesse, tempo, etc.) pour le codage de performances.

Enfin, petit détail, mais pas des moindres, depuis quelques années la connectique MIDI DIN (la connexion classique MIDI) a globalement été mise de côté. Cette connectique, dont des exemples sont visibles ci-dessous (fig. 2.1¹ et 2.2), bien que très fonctionnelle est riche en contraintes. Cette dernière étant unidirectionnelle, un grand nombre de câble est vite nécessaire, d'autant plus que des appareils d'interconnexion intermédiaires sont nécessaires pour créer des réseaux et qu'il est commun de placer des instruments MIDI en cascade. Mais avec l'évolution technologique, l'USB se généralise et il devient de plus en plus commun d'utiliser ce type de connectique à la place de celles DIN, et aujourd'hui USB est pour ainsi dire la norme dans l'industrie. Derrière cette facilité en termes de nombre de câble (l'USB n'est pas unidirectionnel) vient également celle d'exploitation, car pour la plupart des ordinateurs dotés d'USB, il devient natif de pouvoir communiquer avec des messages MIDI. De plus, d'autres alternatives sont aussi nées avec le temps tel que le Bluetooth. Dans cette optique, en plus de représenter un format de données et de message tout indiqué pour le type d'exploitation ciblée, le MIDI a su devenir un protocole simple et efficace dans ses connectiques également. Cet aspect plug-and-play combiné aux principes de son exploitation rend ainsi le MIDI incontournable.



FIGURE 2.1 – Connectiques MIDI DIN femelles



FIGURE 2.2 – Connectique MIDI DIN male

1. Le port MIDI THRU, visible sur la figure 2.1, permet de récupérer une copie directe du MIDI IN, sans modification ni temps de latence induite par les traitements internes (dans un contexte de mise en cascade d'instrument MIDI).

2.2 Google Magenta

Comme il a pu être identifié précédemment dans ce rapport, l'intelligence artificielle est un secteur en vogue attirant l'attention d'énormément d'organismes et de chercheurs. Parmi ces organismes se trouve Google. Ce géant présente en effet un intérêt pour diverses questions technologiques dont l'IA est une partie significative. Suite à cet intérêt a pu naître le projet Google Brain, ce dernier ayant pour objectif de rassembler une équipe autour des questions de l'apprentissage profond. Quelques années après sa création, l'équipe Google Brain crée le projet de recherche open source Magenta. Ce projet a pour objectif l'exploration du rôle de l'apprentissage automatique comme outil dans le processus créatif. Derrière cela, l'idée plus précise est d'offrir aux musiciens, artistes et programmeurs une plateforme leur permettant la création de musique et d'œuvre d'art sur base d'utilisation d'intelligence artificielle. Aujourd'hui, Magenta est parvenu à s'illustrer comme un grand nom dans le monde de l'IA et de la musique, comme peut en attester le grand nombre d'articles liés à ce projet de Google. On y retrouve ainsi par exemple [11], [21] ou encore [8].

Projet Google oblige, Magenta se base sur l'utilisation de TensorFlow, la plateforme open source de Google dédiée au machine learning. De manière extrêmement simplifiée, TensorFlow est une boîte à outils permettant le développement de projets orientés IA. La principale alternative à cette boîte à outils est PyTorch, ces deux géants se partageant le cadre de développement de ce domaine de recherche.

Au fil des années, Magenta a pu proposer énormément d'outils pour l'expression musicale aux technophiles intéressés. Parmi ces projets, il est possible de trouver Magenta Studio [24] qui représente bien le type d'aboutissement qu'il est possible d'atteindre. Comme il est observable dans la figure 2.3, on y retrouve des options de continuation, de génération (fichier par fichier), d'interpolation ou encore d'ajustement de performance de percussions. Ce projet peut également prendre la forme de plugins Ableton Live. Bien que certains des réseaux utilisés commencent à dater, ce projet démontre que l'utilisation d'IA dans la musique n'est plus si loin d'une utilisation grand public abordable. En bref que ce soit à travers ses fondations ou l'exemple de Magenta Studio, Magenta se montre comme un cadre des plus intéressants pour le développement de projets musicaux tels que celui décrit dans ce rapport.

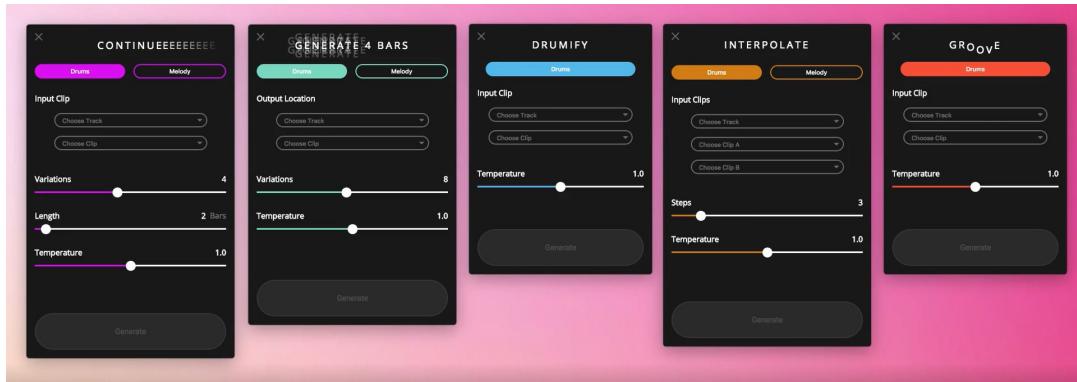


FIGURE 2.3 – Magenta Studio

2.3 Synthèse audio

La synthèse musicale représente un étape charnière dans le processus de création musicale. C'est le passage dans la synthèse musicale qui permet de produire le son que l'utilisateur va percevoir. Cette synthèse peut prendre différents aspects, du plus simple au plus complexe, mais dans la grande majorité des cas, il est possible d'interagir sur des paramètres de la synthèse pour modifier le rendu audio. De nos jours, la synthèse audio est retrouvable partout et l'industrie ne cesse de produire de nouveaux synthétiseurs pour satisfaire l'appétit des créateurs. Combinant à la fois savoir faire de composition, de prise de son et de traitement du signal, la création de solution de synthèse audio est un domaine technique des plus intéressants et productifs.

Cette variété de productions que compte le domaine de la synthèse musicale s'accompagne également d'une variété dans les types d'approche qu'il propose. En effet, si la solution que représente l'utilisation d'un synthétiseur est la plus connue et abordable, elle ne représente pas l'unique solution. Une de ces alternatives est le code ; SuperCollider en est un bel exemple. Cet outils de composition algorithmique représente ainsi un langage de programmation pour la synthèse audio parfaitement praticable. Toujours dans le domaine de programmation, il existe de très belles options de synthèse grâce à la programmation graphique. Max/MSP et Pure Data en sont les deux plus grands représentants. Ce type de programmation permet aussi bien la synthèse sonore, que de l'enregistrement ou encore du contrôle d'instrument. L'idée globale de ce type de démarche est de composer bloc par bloc le traitement de signal sonore (ou non) désiré. Bien que d'apparence simple au premier abord ces solutions permettent une approche complète et complexe (ci-dessous un exemple d'utilisation de Pure Data).

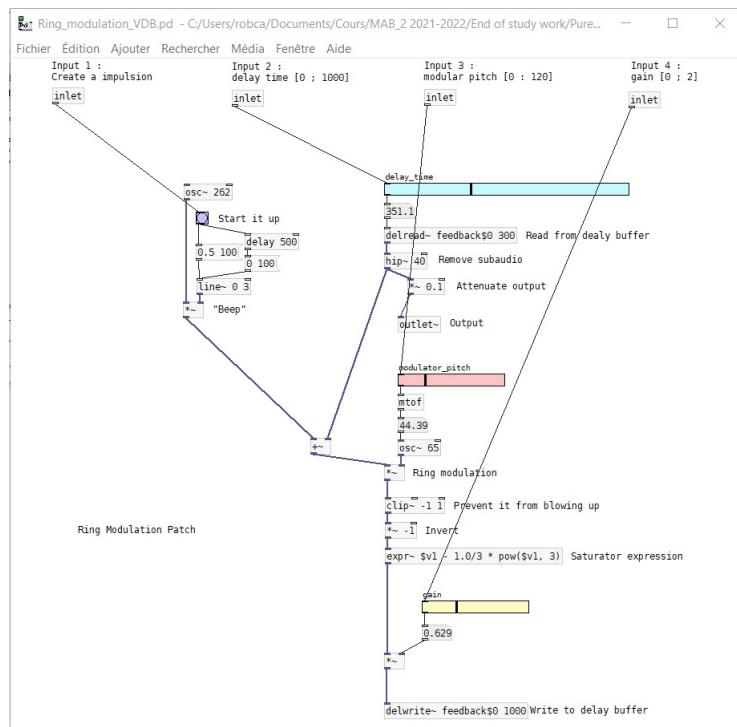


FIGURE 2.4 – Exemple de fichier Pure Data implémenté (ring modulator).

Enfin, la solution la plus typique dans le contexte de la synthèse musicale reste l'emploi d'instruments numériques (synthétiseur ou sampleur). Le principe est alors assez simple : une traduction directe en son des commandes entrées (typiquement du MIDI) paramétrée par des variables spécifiques à l'instrument. On retrouve souvent des paramètres tels que la réverbération, la fréquence du cutoff d'un filtre ou le délai. Ajouté à cela, il est intéressant de noter que le MIDI permet d'interagir avec ces paramètres. En effet, avec une étape de mappage, il est possible d'attribuer tel ou tel bouton, potard, etc. à tel et tel paramètre. Nous verrons dans le prochain chapitre que cet aspect est très intéressant dans le cas du Probatio. De plus, il est possible d'appeler ces instruments depuis des logiciels de MAO. Ils sont donc très pratiques.

2.4 Libmapper

Libmapper [16] est un système permettant de représenter des signaux d'entrée et de sortie sur un réseau et de créer dynamiquement des mappages/connexions arbitraires entre eux. Plus en détails, Libmapper est une bibliothèque logicielle open-source et multiplateforme permettant de déclarer des signaux de données sur un réseau partagé et d'établir des connexions entre eux au désir de l'utilisateur. L'objectif principal du développement de Libmapper est de fournir des outils pour la création et l'utilisation de systèmes de contrôle interactif de synthèse de médias (comme la musique). Récemment, ce système a reçu une amélioration : l'application Webmapper [26]. Cet outil est également un projet de l'IDMIL et a pour principe de fournir une visualisation et un nouvelle manière de manipuler les mappages dans le contexte de la conception d'instruments de musique numériques. Webmapper est ainsi une interface utilisateur permettant d'interagir avec les dispositifs du réseau Libmapper. Cet outil permet un accès à la cartographie des connexions en tant qu'entité distincte et permet une manipulation flexible par n'importe quel outil résidant sur le réseau. Libmapper et Webmapper représentent ainsi des portes d'entrée adéquates pour l'utilisation du Probatio.

Chapitre 3

Le Probatio

Les objectifs que cherche à atteindre ce travail se placent dans la continuité d'un projet de l'IDMIL : le Probatio. Ce projet créé en 2016 par Filipe Calegario se définit comme une boîte à outils open-source pour le prototypage de nouveaux instruments de musique numériques. Dans la pratique, cette boîte à outils se compose d'un ensemble de diverses formes de base et de supports dans lesquels il est possible de positionner arbitrairement tout un set de blocs. Ceux-ci possèdent tout un ensemble d'apparences et d'utilité puisqu'ils ont pour mission de représenter le champ des possibilités d'un prototypeur. Il est de cette façon possible de retrouver des blocs avec des boutons, des pistons, des potards tournants, des surfaces à frapper et bien d'autres (la figure 3.1 illustre certains de ces blocs). Il est également à noter que ce set de près de dix blocs est en évolution constante. En effet, plusieurs nouvelles solutions d'interaction sont actuellement en développement et de multiples pistes d'amélioration sont à l'étude.



FIGURE 3.1 – Exemples de plusieurs blocs du projet Probatio.

Probatio est de fait un projet qui a su évoluer au fil de son développement. Initialement, ce dernier trouve ses origines dans l'analyse de tableaux morphologiques de postures et de contrôle d'instruments de musique. L'idée était alors de permettre aux utilisateurs d'expérimenter différents dispositifs d'entrée pour l'interaction musicale dans différentes positions et postures,

tout ceci à travers l'utilisation de cette boîte à outils de prototypage. Pour répondre à cette volonté, la forme actuelle du projet n'a pas directement été envisagée, mais représente l'objet d'une avancée logique, pas à pas, en termes de praticité. Cette évolution se faisant, d'autres fonctionnalités et utilisations du Probatio ont émergé, si bien qu'aujourd'hui le Probatio atteint ce premier objectif, mais propose aussi des possibilités de prototypage dans un champ large d'applications et n'est plus centré sur les postures. Ainsi à l'heure actuelle, le Probatio emploie des bases et des cubes aimantées qu'il est aisément possible de moduler. De plus, ce système a été et est encore développé en fabrication numérique et utilise deux techniques différentes : la découpe laser et l'impression 3D. Nonobstant, ce système ne se voit pas comme un aboutissement ultime et est toujours sujet à de multiples améliorations possibles. C'est pourquoi des études telles que celle-ci sont, aujourd'hui encore, organisées autour de cette boîte à outils de prototypage.

Pour bien comprendre le Probatio et son potentiel, il est utile de repasser sur les objectifs fondamentaux définis par son concepteur [1] :

1. Comment pouvons-nous fournir des voies structurées et exploratoires pour générer de nouvelles idées de DMI (Digital Musical Instrument) ?
2. Comment pouvons-nous réduire le temps et l'effort nécessaires pour construire des prototypes fonctionnels de DMI ?

Le Probatio dans sa version actuelle (ci-dessous, la figure 3.2 montre un montage du Probatio avec un support de 3 blocs sur une base de 3 sur 3) est donc la solution à ces questions de recherche. Nous garderons, dans une logique de recherche, ces questions en tête dans le but d'améliorer la réponse que propose le Probatio à ces interrogations (voir chapitre 4).

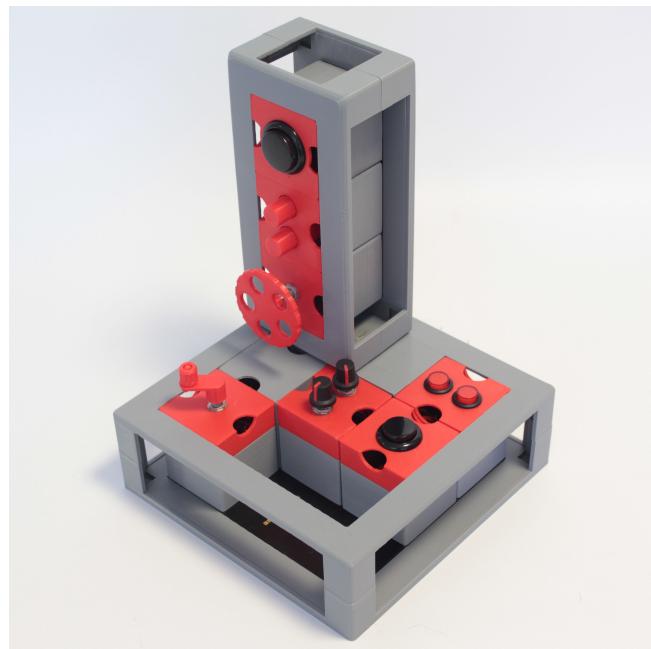


FIGURE 3.2 – Exemple de construction de Probatio.

Chapitre 4

Apports personnels

Comme il peut être compris à travers cette présentation du contexte, le domaine de la musique, et ici tout spécialement celui des outils permettant le développement musical, est particulièrement vaste. De plus, les exigences technologiques liées à ce type d'utilisation sont importantes et spécifiques. Le rôle de ce projet est donc de s'immiscer dans ce domaine technologique, le comprendre, l'adapter au cas spécifique du Probatio, le tout pour en retirer les meilleures pistes d'améliorations pour le projet Probatio. En résumé, il s'agit de poser un œil d'ingénieur sur la situation de ce projet préexistant de l'IDMIL.

De cette façon, la première étape est un processus d'analyse du projet Probatio, afin d'en identifier les voies d'amélioration et/ou les failles. Par la même occasion, cette exploration de voies d'amélioration représente une recherche de potentielles meilleures réponses aux deux objectifs principaux du Probatio (offrir des possibilités exploratoires et réduire le coût, aussi bien en temps qu'en efforts, du prototypage).

De cette analyse en est alors ressorti plusieurs éléments, certains représentant des problèmes mineurs et d'autres des voies d'amélioration significative. Parmi ces points identifiés, certains de ceux-ci sont particulièrement importants et ont un thème commun : l'installation ! Ainsi, la voie d'amélioration dans laquelle se place le travail ici rapporté est celle de l'installation d'autres instruments à employer avec le Probatio. Dans l'optique de cette voie d'amélioration, une solution est proposée dans ce rapport : la création d'un générateur musical, intelligent, modulaire et embarqué.

Cette création représente à la fois une solution à des problèmes importants existants, mais également l'ouverture d'un nouveau champ applicatif. Cette solution permet ainsi de :

1. Diminuer grandement le coût en temps et en effort d'un premier pas dans le prototypage d'un appareil musical lié un autre instrument numérique ou non.
2. Offrir un support musical pour l'improvisation.
3. Mettre à disposition un outil/support de test rapide.
4. Offrir de nouvelles possibilités de création à partir du Probatio.

Pour ce qui est du premier des apports de la solution proposée, en effet, il est parfaitement possible qu'un utilisateur désire utiliser le Probatio en lien avec d'autres instruments. Rappelons

que le Probatio est une boîte à outils de prototypage ; rien n'impose donc à ce projet d'être cantonné à la création d'instruments isolés. Il est tout à fait concevable qu'un utilisateur désire, par exemple, uniquement moduler du son et non être à son origine. Il pourrait alors brancher le Probatio sur le son d'un autre instrument, y compris un purement acoustique (équipé d'un microphone). De cette manière, lors de son utilisation des blocs du Probatio, quelqu'un dans cette optique de création ne pourra pas correctement tester toutes les possibilités d'emploi du Probatio sans l'installation complète de l'instrument auquel il désire adjoindre sa création. Or ce type d'installation peut très vite devenir très compliqué et chronophage. L'idée pour lutter contre ces contraintes est alors de proposer une source musicale de substitution. De cette manière, l'utilisateur n'aurait qu'à lancer cette source pour réaliser ses premiers pas dans le prototypage de manière bien plus légère et aisée. De plus, par le choix du synthétiseur utilisé par le générateur, il est possible de simuler un grand nombre d'instruments. Dans l'exemple d'un utilisateur saxophoniste désirant augmenter son instrument grâce à une solution numérique, ce dernier sera obligé de monter son instrument, de l'équiper de microphones (ceci allant avec tout le matériel nécessaire comme une carte son, des câbles, etc.) et de jouer, pour ne serait ce que tester quel(s) modulation(s) et bloc(s) lui plairaient dans sa situation. Pour un premier pas dans le prototypage, ça n'est vraiment pas des plus agréables en termes d'expérience. Or avec un générateur, cette personne pourrait simplement lancer le logiciel, sélectionner un synthétiseur de saxophone et directement jouer avec le Probatio. Dans cette situation, il n'est évidemment pas question de résumer tout le prototypage à ce type d'utilisation, il est bien question ici des premiers pas dans le prototypage. La création d'un tel générateur (étant le délivrable du présent projet) permettrait ainsi un énorme gain en praticité, en temps et en effort.

Pour ce qui est des possibilités de conception d'un générateur, plusieurs alternatives étaient en réalité possibles :

1. Lire un fichier audio ou midi existant ;
2. Utiliser une génération aléatoire ;
3. Répéter une suite logique simple (suites d'accords, arpèges, etc.) ;
4. Générer un flux « musicalement intéressant ».

Fondamentalement parlant, toutes ces options sont parfaitement viables et fonctionnelles. Néanmoins d'un point de vue plus artistique et technique, toutes ne se valent pas. La première a pour avantage de produire une entrée musicale de qualité en tout temps aux blocs du Probatio. Toutefois, il impose à l'utilisateur de posséder un enregistrement (et de préférence de qualité) préalablement capturé. Ceci peut se montrer incommodant pour un premier pas dans le prototypage. Enfin d'un point de vue technique, si l'utilisateur désire utiliser un enregistrement, que ce soit de lui-même ou non, d'autres outils de lecture seront sûrement plus indiqués que de passer par ce module de source musicale. Pour ce qui est du second choix, ce côté purement aléatoire risque très vite d'être très désagréable pour l'utilisateur. De plus, cette option n'a que très peu d'intérêt musicalement parlant. La troisième option quant à elle représente typiquement l'image de la solution sans prise de risques. Elle est assez indiquée, ne sera sûrement pas désagréable, bien qu'extrêmement répétitive, mais ne présente pas de grand intérêt musical. En bref, fonctionnelle, mais sans plus value. Enfin, la dernière est techniquement plus complexe, mais est musicalement parlant bien plus intéressante. Ainsi,

bien que plus proche d'un challenge technique, l'intérêt pratique et musical de cette quatrième option la place sur le haut du podium. Enfin, il ne faut pas non plus perdre de vue que le Probatio a été créé pour donner des possibilités ; un générateur imprévisible (riche, en termes de son) délivrant un flux « musicalement intéressant » a pour sûr plus de chance de représenter un nouveau bloc d'intérêt pour une potentielle utilisation alternative. Avec cette quatrième approche, l'utile rencontre l'agréable. C'est pourquoi le générateur conçu se veut intelligent. Il offre ainsi une sortie suffisamment qualitative pour être un support d'improvisation, mais il se veut également suffisamment facile d'utilisation que sa sortie pourra être utilisée pour des tests quelconques. Enfin, il se veut modulaire et embarqué de façon à être utilisé facilement partout et dans plusieurs circonstances.

Cette direction de développement sélectionnée, la contribution personnelle représente ainsi l'implémentation d'un générateur et par la même occasion la création d'un ensemble de nouvelles possibilités offertes par la boîte à outils Probatio. Pour atteindre cet objectif, différentes connaissances sont utilisées telles que celles relatives à l'intelligence artificielle. Cette option technologique représentant en effet la meilleure voie de développement pour de tels objectifs. Néanmoins, il n'est donc pas ici question de révolutionner le domaine de l'intelligence artificielle en termes de qualité de production musicale, mais bien de trouver une solution d'adaptation de ce type d'IA permettant d'être utilisé dans un cadre tel que le Probatio et de soutenir le processus créatif.

N.B. : La démarche dans ce projet a été comme expliqué plus tôt de placer un œil d'ingénieur sur le cas du projet Probatio. C'est pourquoi d'autres petites améliorations ont pu être apportées au projet. Par exemple, le Probatio utilise un petit module nommé M5 de façon à obtenir une communication sans fil avec un ordinateur. Ce dernier est placé sur un bloc relié par câble à la base du Probatio. Une des améliorations a alors été de supprimer ce bloc extérieur et son câble et créer un design 3D d'un bloc de Probatio pouvant accueillir le M5. De cette façon, il n'est plus question de casser le côté compact du Probatio avec un module externe (voir fig. 4.1 et 4.2).

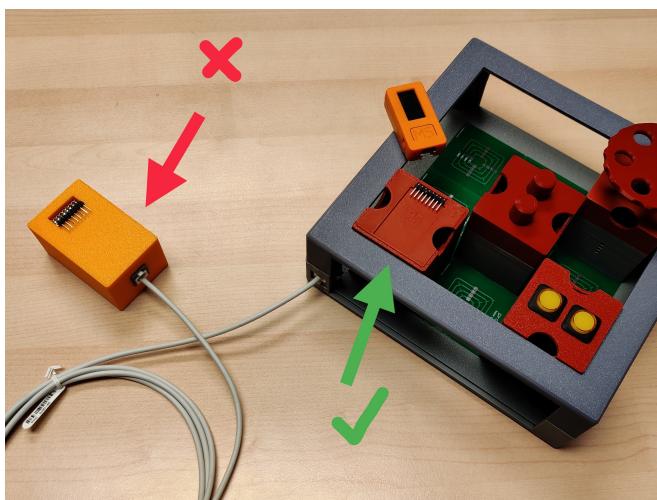


FIGURE 4.1 – Remplacement du boîtier externe du M5.



FIGURE 4.2 – Nouveau bloc de connexion du M5.

Chapitre 5

Développement du générateur musical

Ce chapitre est consacré au processus de développement du générateur musical intelligent, modulaire et embarqué. De manière à correctement appréhender ce processus, ce chapitre se divisera en différentes sections portant sur les différents niveaux de conception du générateur. L'approche globale est la première de ces sections et définit les grandes lignes de la logique du développement. En deuxième, vient la présentation du réseau neuronal exploité, suivie par la présentation du programme complet. Enfin, en quatrième, il est question des possibilités d'exploitation du système dans un contexte embarqué. Il y est notamment abordé l'utilisation d'outils tels que le Raspberry Pi.

5.1 Approche globale

Suite à la présentation des apports personnels, des objectifs ont pu être définis. Cette section aborde donc les voies de développement permettant d'atteindre ces objectifs.

L'idée étant de créer un flux continu de type MIDI, il est nécessaire de construire un encadrement de la partie génératrice. En effet, cette dernière n'est capable que de générer des fichiers MIDI un à un. Ceci n'est nullement un problème dans un contexte d'utilisation similaire à Magenta Studio (générer une séquence puis travailler dessus), mais ce n'est pas ce qui est recherché ici. Globalement, nous ne cherchons pas à nous attacher à un fichier créé en particulier, mais bien à en parcourir un ensemble de manière continue. Le principe va donc être la mise en parallèle de 3 parties majeures : la génération (à proprement parlé), la lecture et un menu permettant de modifier des paramètres spécifiques. Ajouté à cela se greffe également une quatrième partie puisque, par défaut, il est nécessaire d'effectuer la synthèse du son à partir du flux MIDI généré.

Bien que l'ensemble du générateur soit composé de parties distinctes, il est de bon usage de les considérer de manière non dissociées pour obtenir un résultat plus agréable à l'utilisation. Un exemple de cela est l'évolution de la lecture. Dans un cas où cette partie lecture devancerait la partie génération, le flux s'en trouverait interrompu ; ce qui n'est bien évidemment pas désiré. Il faut donc que l'avancée de tout à chacun à l'intérieur du générateur conditionne l'évolution des différentes autres parties. Pour ce qui est de cette communication entre parties,

l'approche globale employée dans ce travail a été de préserver au plus le flux de sortie et la qualité d'utilisation. Pour ce qui est du menu, l'idée va donc être d'appliquer les changements de manière continue et la moins brusque possible. On préférera donc éviter de couper dans la lecture d'une séquence musicale pour le changement d'un paramètre, pour plutôt mémoriser le choix et l'appliquer à un moment plus opportun. Le système gagne donc en inertie, mais pour un bien. Pour la lecture, l'idée va être l'observation des paramètres passés par le menu et l'analyse de la production de la partie génératrice. Cette lecture étant à l'écoute des autres parties, elle sera capable de s'adapter, bouclant si nécessaire par exemple. La partie génératrice étant le cœur du système, c'est avant tout aux autres parties de se plier à son fonctionnement. Pour sa part, son changement majeur est sa possibilité de passer en mode conditionné. Le reste du temps, son processus est continu et imperturbable.

Bien qu'il soit intéressant de travailler sur un système possédant son autonomie, il reste bon de posséder des moyens de contrôle simple sur ce dernier. Ainsi pour ce qui est des interactions avec l'utilisateur, le système propose en continu différentes options :

1. **Boucler** : le générateur créant de manière continue, il n'est initialement pas prévu de stagner sur une des séquences générées. Toutefois, grâce à cette commande l'utilisateur peut demander au système de boucler sur la séquence qu'il est en train de lire. Une fois lassé de cette boucle, l'utilisateur pourra reprendre le parcours classique des séquences grâce à cette même commande.
2. **Pause** : élémentaire, mais utile, la lecture peut être mise en pause sans pour autant couper la partie génératrice.
3. **Encucher le mode conditionné** : ce mode spécifique consiste en l'exploitation d'un fichier MIDI de petite taille pour le prolonger. L'utilisateur a donc la possibilité de travailler sur des fichiers sélectionnés ou enregistrés par ses soins dans le processus de génération. Cette continuation représente ainsi une génération conditionnée par une petite séquence de l'utilisateur. (La section consacrée à l'implémentation reviendra plus en détail sur cette option.)
4. **Volume** : dans le cas où l'utilisateur souhaite conserver la synthèse audio interne au générateur, il peut modifier le volume de cette production audio.
5. **Éteindre le système** : rien n'est éternel, la volonté d'usage du générateur d'un utilisateur ne fait pas exception.

Ajouté à cela, comme expliqué dans le chapitre 4, le système propose à son allumage le choix du type de sortie (MIDI ou audio). Le mode conditionné possède également son propre menu permettant de relancer une nouvelle génération sur le même fichier base sélectionné, de changer de fichier base et bien-sûr de sortir de ce mode conditionné.

Résumé en quelques phrases, le générateur est un système composé de trois parties : la partie génératrice, la lecture et le menu. Sans interactions nécessaires, le lecteur va créer un flux MIDI continu sur base des séquences générées par la partie génératrice. Enfin, le menu permet un contrôle facultatif de la lecture et offre la possibilité de déclencher un mode conditionné de la partie génératrice. Ajouté à tout cela s'ajoute un ensemble de mesures visant à augmenter la robustesse du système et la synthèse audio interne si sélectionnée par l'utilisateur.

5.2 Présentation du réseau de neurones exploité

Ce fameux processus de génération ne peut toutefois pas fonctionner sans son réseau de neurones entraîné. Il est donc appréciable d'en comprendre le fonctionnement. Voici donc la présentation de Music Transformer [11], le cœur de la génération.

5.2.1 Les transformers

Avant d'entrer plus en détails sur le réseau utilisé, il est important de revenir sur un des deux termes qui composent son nom : transformer. Ce terme n'est pas anodin et fait référence à un nouveau type d'architecture de réseaux de neurones développé à partir de 2017 et initialement imaginé pour des tâches de traduction. Celui-ci a permis une belle avancée dans le domaine de l'intelligence artificielle et représente maintenant une approche utilisée dans énormément de types de tâche. Ce type de réseaux se confronte principalement aux RNN (réseaux de neurones récurrents) dont il se montre une amélioration sur bien des points, dont un très important qui est la possibilité de parallélisation. Cette amélioration offre ainsi la capacité de traiter d'énormes bases de données, ce qui représente une ouverture du champ des possibilités et une force pour la qualité et la pertinence des réseaux.

En ce qui concerne le fonctionnement des transformers (ou transformateurs en français), leur développement se base sur trois concepts principaux : l'encodage positionnel, l'attention et l'auto-attention (plus souvent appelée sous son appellation anglaise « self-attention »).

Le premier de ces concepts, l'**encodage positionnel** représente à lui seul un énorme pas en avant face aux RNN. En effet, l'avantage des RNN est leur manière séquentielle de traiter leurs entrées et donc de permettre au passé d'influencer le présent. Mais dans les faits cette vision est limitante, à la fois car le simple traitement des éléments les uns après les autres ne considère pas toutes les influences des éléments de la séquence entre eux, et également car empêchant la parallélisation. L'encodage positionnel, lui, permet d'entrer toutes les données, mais en leur assignant un ordre. Bref, tout rentre, mais on n'en oublie pas pour autant la structure d'origine. Dans l'exemple d'un phrase (puisque c'est de la traduction qu'est originaire les transformers), il s'agirait de ne plus traiter les mots un par un en les parcourant, mais bien d'utiliser ensemble tous les mots « numérotés ». Dans les faits, la théorie est plus complexe et utilise des fonctions sinus pour le positionnement, mais cette vulgarisation reste une bonne approche.

L'**attention**, quant à elle, représente un concept relativement simple : certains éléments sont plus importants/impactants que d'autres. C'est à dire que certains des éléments ont tendance à conditionner les autres éléments autour d'eux. A nouveau dans le cas de la traduction, ce mécanisme permet notamment de répondre à des problématiques telles que l'accord de l'adjectif. Cette capacité d'identification est typiquement issue de l'apprentissage et de la possibilité de passer sur de nombreuses données.

Enfin, la « **self-attention** » est la partie la plus extraordinaire des transformers puisqu'elle consiste en la compréhension, par le système, de sens sous-jacents aux informations. Ceci est rendu possible grâce des représentations internes des éléments que le système va acquérir lors de son entraînement sur les données. Ce concept permet l'obtention d'un système plus

intelligent puisqu'il donne l'impression de comprendre le sens des éléments qui lui sont passés. Dans le cadre de la linguistique, cela s'apparente à la compréhension de synonymes ou encore à la différenciation de mots d'apparences identiques mais au sens différents.

Basés sur ces concepts, les transformers prennent la configuration d'encodeur-décodeur. En effet, comme leur nom l'indique les transformers ont été imaginés pour prendre une entrée et la transformer en quelque chose d'autre (à mettre en sortie). Pour cela, l'architecture se scinde en deux parties. La première prend l'entrée et en déduit une certaine interprétation. Il s'agit de l'encodeur. Cette interprétation est ensuite placée dans un décodeur qui va générer une sortie sur base de cette interprétation. Les données ont alors été transformées, le transformateur a réalisé son œuvre.

5.2.2 Music Transformer

Music Transformer est donc un réseau de neurones, d'architecture similaire à celles décrites dans la sous-section précédente, adaptant ce type d'approche au cas de la génération musicale. Avant tout chose, il est également à souligner la cohérence des transformers dans le domaine musical avant même une quelconque adaptation. Ces principes d'intérêt pour les composants d'une séquence et leur ordre, pour la perception de l'importance des composants et pour la compréhension de sens sous-jacents, sont ainsi autant de concepts également applicables au domaine des performances musicales. Il y a donc, dès l'initiale, un croisement logique entre musique et transformer.

En effet, un morceau de musique possède une structure composée la majorité du temps d'éléments récurrents, de motifs, de phases spécifiques, comme la découpe bien connue refrain-couplet. Pour être cohérent un générateur de morceaux de musique doit avoir conscience des éléments qu'il génère, de manière à pouvoir faire référence à certains de ses propres éléments générés. La cohérence est donc issue en bonne partie d'un bon référencement à la fois d'éléments d'un passé proche comme lointain. Ainsi, sur base de composantes passées, le système doit être par répétition, variation ou encore développement, capable de générer de la cohérence, du contraste et de la surprise. Les transformateurs se posent alors comme la solution idéale, car permettant l'accès à n'importe quelle partie de la sortie précédemment générée à chaque étape de la génération. La figure 5.1 issu de [11] en est une illustration assez parlante puisqu'elle montre la recherche dans le passé de notes de référence¹. Il est aussi à remarquer que les deux représentations de la figure (dans leur prospection de références) vont majoritairement étudier des notes voisines et similaires à celles dont émane la recherche. Il s'agit là d'une image de la compréhension du contexte. De ces observations, Music Transformer prend le parti d'une répétition de l'auto-attention dans plusieurs couches successives d'un décodeur Transformer, le tout de façon à capturer efficacement les multiples niveaux issus des phénomènes d'auto-référence existant dans la musique.

Une grande différence par rapport aux transformers « de base » est toutefois à noter. Dans [11], les auteurs se séparent du référencement absolu des positions, pour opter pour un

1. Il s'agit d'un type de représentation 2D nommé pianoroll. L'axe horizontal peut être vu comme l'évolution temporel de gauche à droite. L'axe verticale, lui, représente la hauteur des notes. Ce type de représentation est très courant en musique.

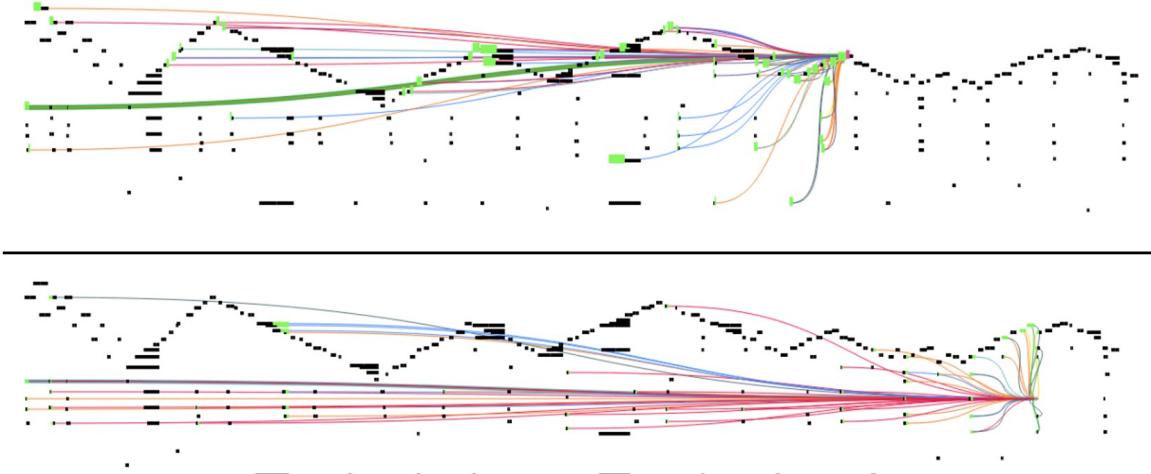


FIGURE 5.1 – Illustration du processus d'influence du passé sur la génération à travers des lignes d'attention.

référencement relatif. Cette approche relative satisfait plus efficacement les besoins liés au travail de la musique et permet une plus grande liberté en termes de longueur de séquences générées. De plus, il adopte la représentation « MIDI-like » de Performance RNN [21]. Enfin, ajouté à cela, les auteurs ont développé une amélioration algorithmique de grande importance, car nécessaire au travail des structures plus longues de la musique. Cette amélioration consiste en un passage des besoins en mémoire de $O(L^2D)$ à $O(LD)$ avec L, la longueur de séquence et D, la dimension de l'état caché du modèle (pour plus de détails voir l'annexe A).

L'aboutissement de Music Transformer est la démonstration que les transformers équipés de l'attention relative sont très bien adaptés pour la modélisation générative de la musique symbolique. En effet, au moment de sa sortie, l'article [11] a surpassé l'état de l'art, et aujourd'hui encore les recherches de ce domaine utilisent les transformers pour ce type de tâches en se référant à cet article.

Dans un contexte plus pratique, pour ce travail, un réseau entraîné a pu être récupéré. Ce dernier est issu d'un entraînement sur un ensemble de données particulièrement massif. Cet ensemble est en effet composé de plus de 10 000 heures de musique symbolique au piano, issues de vidéos YouTube (dont la licence permettait l'utilisation). A noter également, que comme le modèle nécessite une représentation symbolique, un autre projet Google a été utilisé pour former la base de données. Il s'agit de « Onsets and Frames » [10], un projet permettant la traduction automatique de performance de piano en données de format MIDI.

5.3 Implémentation du système

L'implémentation du générateur ne cache pas de grand mystère et représente à quelques détails près la traduction codée de l'approche globale. Le langage sélectionné pour la réalisation du code est python. Ce langage a en effet su s'imposer comme le grand choix de prédilection

pour l'implémentation d'intelligence artificielle ces dernières années. De plus, sa structure, sa clarté et sa polyvalence en font un choix des plus adaptés pour le code de manière générale. Ajouté à cela, le réseau neuronal utilisé est lui aussi implémenté en python, ce qui éloigne les possibles hésitations sur le langage informatique à utiliser.

L'écriture du générateur se développe en de multiples fonctions permettant la composition des trois grandes parties du générateur (génératrice, lecture et menu). Ces dernières tournent en parallèle grâce à du *threading* et échangent leurs variables grâce à des queues FIFO (First In First Out). Ajouté à cela, pour le bon fonctionnement du générateur, vient un ensemble d'importations comprenant également celle du réseau entraîné. Ce rapport portant sur l'utilisation du réseau pour la production musicale, il ne rentre pas dans les détails de l'entraînement de ce réseau.

Pour un plus grand confort d'utilisation, un ensemble de mesures de robustesse sont apportées à la logique du générateur. Parmi celles-ci, on peut compter une protection de la mémoire à travers un encadrement des séquences générées, des sécurités de base pour tous les menus, le bouclage si nécessaire de la dernière séquence lue si la lecture venait à rattraper la génération, et quelques autres petits points d'amélioration pour le bon fonctionnement du générateur. Également de manière à donner un bon départ au générateur et ne pas risquer de boucler trop vite sur la dernière séquence produite, le système attend la génération des deux premières séquences. Cette protection n'est pas fondamentalement utile sur des ordinateurs haute performance, mais sera appréciable sur des appareils ou cartes de moindre puissance.

Enfin, deux options supplémentaires ont été ajoutées lors de l'implémentation. La première de ces options est un moyen de récupération aisée des séquences générées. De cette manière, un utilisateur désirant récupérer les séquences d'une session d'utilisation aura, sans difficulté, accès à celles-ci dans un dossier dédié. La seconde plus importante est la mise à disposition d'un système de synthèse audio embarqué dans le générateur. Cette option mise à disposition permet un premier prototypage encore plus rapide puisque ne nécessitant pas le démarrage de systèmes de synthèse annexe. Elle offre également un équivalent de système de débogage ou de test plus rapide et efficace. Cette synthèse embarquée n'est paramétrable qu'à travers son volume et compose un son de piano. Il ne s'agit évidemment pas de la synthèse la plus élégante, mais elle a le mérite de permettre une lecture MIDI sans dépendance externe au générateur. La qualité de cette synthèse n'est donc pas un problème pour son utilisation, bien qu'elle soit tout de même convenable.

5.4 Systèmes embarqués

L'utilisation de ce système est volontairement très simple. Celui-ci peut fonctionner de son côté sans interaction, l'utilisateur n'a qu'à lancer le code et sélectionner la sortie désirée (audio ou MIDI, et si MIDI quel port utiliser). Cette seule interaction obligatoire pouvant être automatisée sans peine, le système peut donc être placé n'importe où, à l'unique condition d'avoir installé les librairies nécessaires et le réseau entraîné.

Dans cette optique, bien que tout ce système soit parfaitement fonctionnel sur un ordinateur « classique », la question de placer ce générateur sur de petites cartes électroniques telles

qu'un Raspberry Pi² s'est posée. Dans la logique du Probatio cela correspondrait à placer le générateur dans un bloc du kit. Cependant, bien que l'idée semble simple, elle engendre un ensemble de sous problématiques. Dans le cadre de ce travail, un proof of concept (POC) sur Raspberry Pi a été réalisé pour tester les possibilités embarquées. De plus, la logique a été poussée plus loin avec un système de traitement local pour Probatio, permettant une suppression de la dépendance à l'ordinateur et offrant par la même occasion une option de synthèse audio alternative.

5.4.1 Première approche : le tout embarqué

La première voie d'exploration pour ce type de question est tout simplement de placer l'intégralité du processus dans une carte électronique. Ceci est tout à fait possible, mais ne représente pas l'unique solution. Bien évidemment les performances de petites cartes telles que les Raspberry Pi 4 ou zero sont moindres face à un ordinateur. Par conséquent, l'inertie du générateur s'en retrouve plus grande dans ce type de configuration. Bien évidemment, cela ne rend pas l'utilisation impossible, mais moins prolifique. Le lecteur risque ainsi de plus souvent rattraper la génération. Par conséquent, des boucles de lecture de temporisation ont plus de chance d'être utilisées lors de l'emploi du générateur. Cependant, la robustesse du système est prévue pour pouvoir supporter de telles configurations.

Il est néanmoins à noter qu'il est à l'heure actuelle relativement difficile de placer toutes les librairies nécessaires à la partie génératrice sur Raspberry Pi. De plus, l'actuelle transition de TensorFlow vers sa version 2 rend le processus réellement ardu. Cette transition toujours en cours a par ailleurs empêché la réalisation de l'intégralité des tests de cette voie 1 de développement, la Raspberry Pi ne pouvant accéder et utiliser correctement toutes les librairies nécessaires. Il est cependant fort probable que la situation s'améliore avec le temps et avec la stabilisation de cette transition entre versions de TensorFlow.

5.4.2 Deuxième approche : génération en ligne

La voie d'exploration 2 quant à elle prend le contre-pied de ces problèmes d'installations et de performance. L'idée est celle-ci : si ces cartes possèdent une connexion wifi rien ne nous empêche d'exploiter une génération en ligne. Et de fait, cette solution est parfaitement fonctionnelle ! La seule vraie contrainte de ce type de fonctionnement est la nécessité de posséder une connexion internet sur un wifi accessible à la carte. Un POC de cette approche a été entièrement réalisé et a su montrer un fonctionnement parfaitement intégré. Il a bien entendu été nécessaire d'adapter le code de cette version, mais dans l'application cela n'a rien de très contraignant pour l'utilisateur. La démarche à adopter dans cette configuration est de placer en ligne le processus de génération uniquement. Ce dernier n'a alors qu'à lancer le téléchargement de ses séquences à la fin de chaque génération. De plus, de façon à lutter contre l'impossibilité qu'ont le générateur et le lecteur à communiquer ensemble, le lecteur change complètement son processus d'analyse de l'évolution du générateur. Pour remplacer l'échange de variables entre eux, le lecteur va inspecter en permanence un dossier prévu pour le dépôt

2. Les cartes Raspberry Pi sont des nano-ordinateurs à processeur ARM qu'il est possible de trouver en différentes tailles, performances et dotés de différentes fonctions. Le Raspberry est né d'un projet de l'université de Cambridge de démocratisation de l'accès à l'informatique. Il possède également son propre OS et est aujourd'hui au cœur d'énormément de projets de par le monde.

des séquences générées (en ligne) téléchargées. De par cette analyse constante du dossier, le lecteur peut ainsi savoir exactement où en est le générateur en tout instant. A noter que pour le POC réalisé, il n'a pas été recherché d'utiliser la génération conditionnée. De fait, dans le cadre d'une utilisation d'un bloc générateur à l'intérieur d'un instrument construit grâce au Probatio, la logique est avant tout d'avoir quelque chose de « plug and play ». Il est tout à fait concevable de rajouter des manières d'interagir avec le bloc telles que des encodeurs rotatifs, des boutons, un écran, etc. Mais puisque la partie génération est en ligne, s'il est désiré de la modifier autant le faire en ligne directement grâce à un autre appareil. Il est donc malgré tout possible de conditionner la génération, bien que l'utilisation de ces fonctions soit plus indiquée dans la version classique (sur ordinateur). Pour ce qui est des autres options du menu d'interactions avec la lecture, elles restent tout à fait accessibles et fonctionnelles sur le Raspberry Pi.

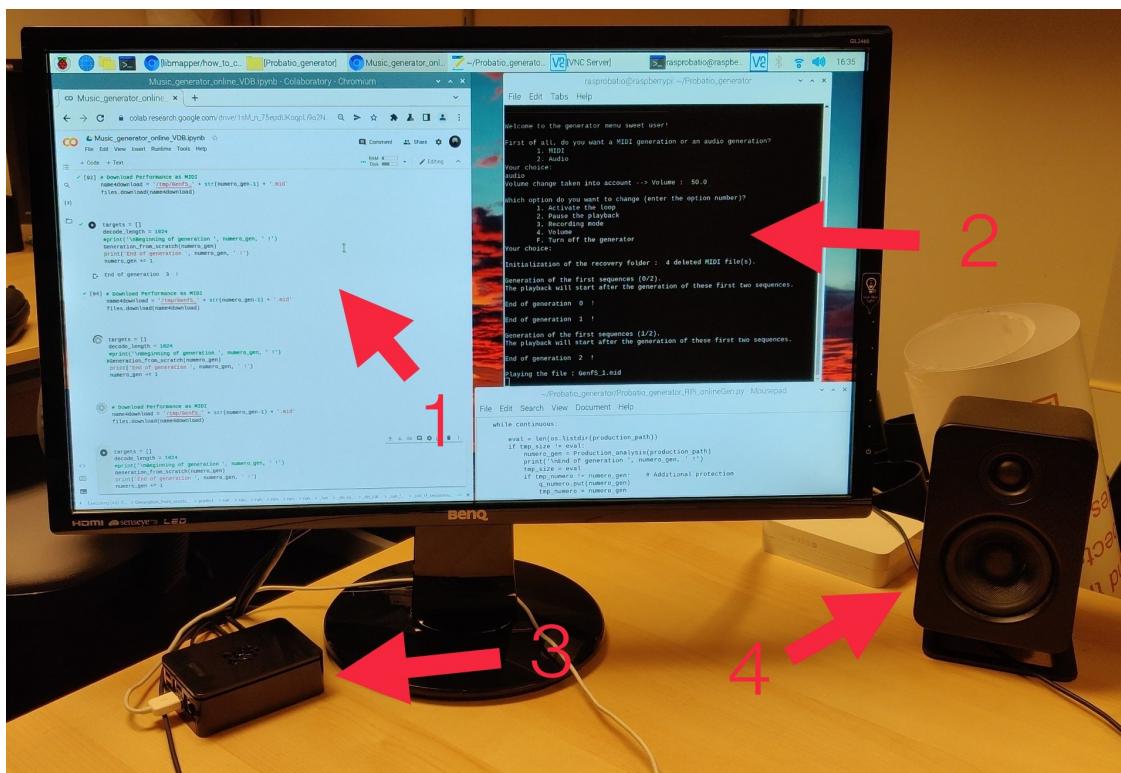


FIGURE 5.2 – Photo du POC : système embarqué et génération en ligne.

La figure 5.2 est une photo du POC réalisé³. Sur la photo, le Raspberry Pi (modèle 4) est relié à un écran pour offrir une visualisation et utilise la synthèse audio interne. Pour ce POC,

3. Éléments présents sur la photo du POC :

 1. Générateur en ligne : dans le cas du POC, Google Colab ;
 2. Console du générateur avec menu de contrôle du lecteur ;
 3. Carte contenant le générateur dans sa version adaptée : Raspberry Pi 4 modèle B ;
 4. Sortie audio.

la solution en ligne expérimentée est Google Colab. Ce choix représente une solution simple à mettre en œuvre et suffisante pour un proof of concept. Il a donc pu être prouvé par voie expérimentale que le générateur pouvait être employé sur des cartes électroniques de plus faible performance grâce à une génération en ligne et une adaptation du code. De plus, la génération en ligne éliminant la quasi intégralité des besoins en puissance de calcul, il est tout à fait concevable de passer sur de plus petites cartes encore, telle que le Raspberry Pi Zero W. Il est à ajouter également que si les paramètres ont pu être modifiés par la console, il est également tout à fait possible de les voir modifier via les GPIO de la carte grâce à de simples boutons, encodeurs, etc.

5.4.3 Troisième approche : Ajout d'une unité locale de traitement de signal

Cette dernière ne se place pas en concurrence des deux précédemment expliquées, mais vient davantage se placer en tant que complément logique. En effet, bien que ces options de génération soit fonctionnelles, il n'en demeure pas moins qu'elle ne possède pas d'interaction avec les autres blocs du Probatio sans l'utilisation d'un ordinateur. Or dans un processus d'intégration du générateur, cette limitation est un manque à gagner. Cette troisième approche propose donc la création d'un patch Pure Data pour l'utilisation du flux généré, permettant ainsi une voie d'interaction avec les blocs du Probatio sans l'aide d'un ordinateur complémentaire.

Le choix de Pure Data n'est pas anodin. L'avantage de ce logiciel est qu'il est assez léger et est capable de tourner sur des cartes aux performances limitées telles que des Raspberry Pi. De plus, le projet Libmapper [16] a par le passé créé un objet spécifique permettant le mappage d'instruments de musique numériques dans Pure Data. Le logiciel possède donc déjà une porte d'entrée pour les interactions avec les blocs du Probatio. Cet objet en question porte le nom de [mapper]. Son fonctionnement, comme imaginé spécifiquement pour ce type d'usage, n'est pas très contraignant. Ci-dessous la figure 5.3 illustre assez simplement un envoi et une réception de signal. Dans le cas de cet exemple, à gauche un signal de sortie est défini (add output /sendsig) et indiqué comme de type float (@type f). La valeur de ce signal peut alors être modulée grâce à la case grisée reliée au bloc [mapper]. De manière similaire, un signal de réception est défini à droite (de type float également). A ce moment, si l'interface graphique de mappage de Libmapper est utilisée pour créer une connexion entre les signaux sendig et recvSIG, toute modification apportée à la valeur de la case grisée sur la gauche entraînera une sortie correspondante sur la droite. Dans cette logique, il est parfaitement concevable de recevoir les signaux mappés du Probatio à travers l'objet [mapper].



FIGURE 5.3 – Exemple d'utilisation de l'objet [mapper].

Cette réception étant dessinée et cartographiée, les signaux entrant peuvent être placés

en entrée de différents blocs Pure Data. De façon à garder une conception claire, l'idéal est de concevoir des blocs spécifiques à des traitements du son définis, comme un bloc dédié à la réverbération par exemple. Il est également à ajouter que les blocs Pure Data ne sont pas uniquement cantonnés à la modulation du son, mais peuvent être conçus comme des synthétiseurs où même comme des instruments à percussion, comme peut le montrer la figure 5.4. Dans ce cas, le bloc Pure Data créé a quatre entrées : une pour déclencher un coup de batterie et indiquant quel son produire (kick, snare, hat⁴ ou une combinaison de ceux-ci) et trois autres pour régler le volume des différentes parties de la batterie. Ainsi, un utilisateur pourrait très bien assigner un bouton d'un bloc de Probatio comme un coup de snare ou de hat. Il n'y a en réalité pas tant de limites à ce qui peut être assigné aux signaux du Probatio.

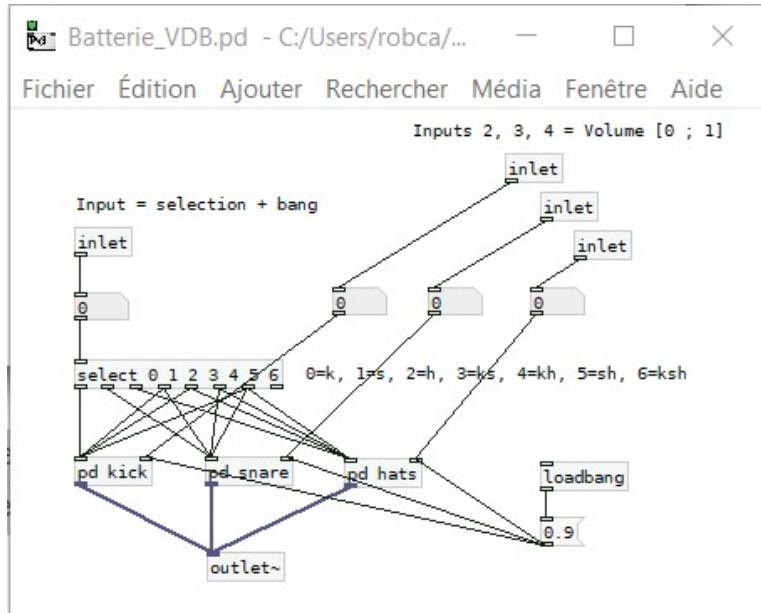


FIGURE 5.4 – Exemple blocs Pure Data : la batterie.

Dans tout cela, le générateur peut très facilement trouver sa place. En effet, celui-ci peut soit générer un flux MIDI ou un flux audio (si la synthèse audio interne est sélectionnée), or Pure Data possède des objets spécifiques aux entrées MIDI et audio. Il n'y a donc en réalité qu'à relier les blocs ensemble, suivant la volonté de l'utilisateur. Le logiciel Pure Data peut ainsi aussi bien offrir une simple synthèse audio au générateur, qu'un moyen d'obtenir des interactions complexes avec le Probatio. Le tout pouvant être embarqué, cette solution représente une alternative intéressante pour une utilisation du Probatio sans dépendance à l'ordinateur.

Néanmoins, bien qu'il soit très intéressant que l'ordinateur ne soit pas utile lors de l'utilisation du Probatio, cette utilisation engendre la limitation d'un code Pure Data statique. Une alternative a donc été imaginée pour offrir à l'utilisateur une version modifiable au désir et

4. Le kick est le nom que l'on donne à un son de grosse caisse, le snare est lui un son de caisse claire et le hat est un son de charleston.

sans ordinateur. Cette alternative repose sur l'utilisation d'un écran tactile intelligent⁵. Cette interface Homme-Machine permet alors de sélectionner pour chaque bloc l'effet désiré dans un contexte « user friendly ». Malheureusement, bien que toute la logique ait été écrite, un POC n'a pas pu être réalisé pour cause de disponibilité de matériel, mais dans la théorie, cette alternative complémentaire est fonctionnelle.

5. L'écran sélectionné est un écran de type Nextion. Ce dernier possède le grand avantage de posséder son propre processeur et une mémoire, le tout permettant d'embarquer toute la logique du menu offert à l'utilisateur. Dans cette optique, le logiciel Pure Data récupère une entrée serial simple prétraitée par l'écran (prenant la forme d'un tableau d'entiers). Cet avantage est significatif, car il évite d'alourdir le processus embarqué par la carte. De plus, l'écran offre un confort d'utilisation que la carte aurait du mal à générer.

Chapitre 6

Résultats

Au terme de la création du générateur décrit à travers le chapitre précédent, de multiples générations ont pu avoir lieu. De ces productions ont pu être tirées des analyses permettant de caractériser différents aspects de l'utilisation du générateur. Ce chapitre est ainsi dédié à ces analyses et à la compréhension des données que nous livre le fonctionnement du générateur.

Toutefois, comme expliqué dans la section 1.3.3, l'évaluation est un problème conséquent qui impose de trouver des alternatives de caractérisation. Pour rappel, la musicalité est un concept très complexe qu'il est difficile de définir aussi bien par des mots que par des équations. Une étude parallèle a par ailleurs été réalisée, demandant à un panel de personnes varié (musiciens et non musiciens d'âges variés) comment il définissait le concept de musicalité. Les résultats de cette dernière (voir annexe C) sont très clairs : il n'y a pas d'unanimité, mais uniquement une difficulté commune à définir le concept. Néanmoins, il en ressort tout de même certains points clés. De manière synthétique (et basée sur les résultats du sondage), la musicalité serait quelque chose d'intrinsèquement lié au qualitatif, donnant à la fois un sentiment de qualité à l'écoute et des émotions. En bref, ce sondage n'a pas fait exception à la règle et montre bien qu'évaluer la qualité d'une production musicale sur sa musicalité est quelque chose de compliqué et de fondamentalement lié à l'appréciation qualitative.

Dans ce contexte, les données présentées dans ce chapitre prendront diverses formes : purement quantitatives, purement qualitatives, mais aussi entre ces deux mondes en tentant d'obtenir des liens logiques entre ceux-ci. Pour cela, ce travail utilise des métriques issues de l'article de Yang et Lerch [28] permettant l'analyse de certaines composantes bas niveau de la musique. Dans un second temps, il se base également sur un test de perception humaine réalisé sur plus d'une trentaine de participants (exemple de formulaire de test en annexe D) aux profils variés.

Pour répondre à ces analyses variées, ce chapitre se scinde en une analyse de productivité du générateur, une analyse de l'appréciation humaine à travers un test qualitatif, une analyse quantitative du générateur et une analyse critique globale.

6.1 Analyse de la productivité du générateur

Ce premier résultat (fig. 6.1) illustre très simplement l'évolution de la lecture face à la génération grâce au temps cumulé de leur consommation et production. La courbe bleue représente donc l'apport de temps musical disponible à la lecture issu de chaque création de la partie génératrice. La courbe orange, elle, représente le temps musical consommé au total par la lecture.

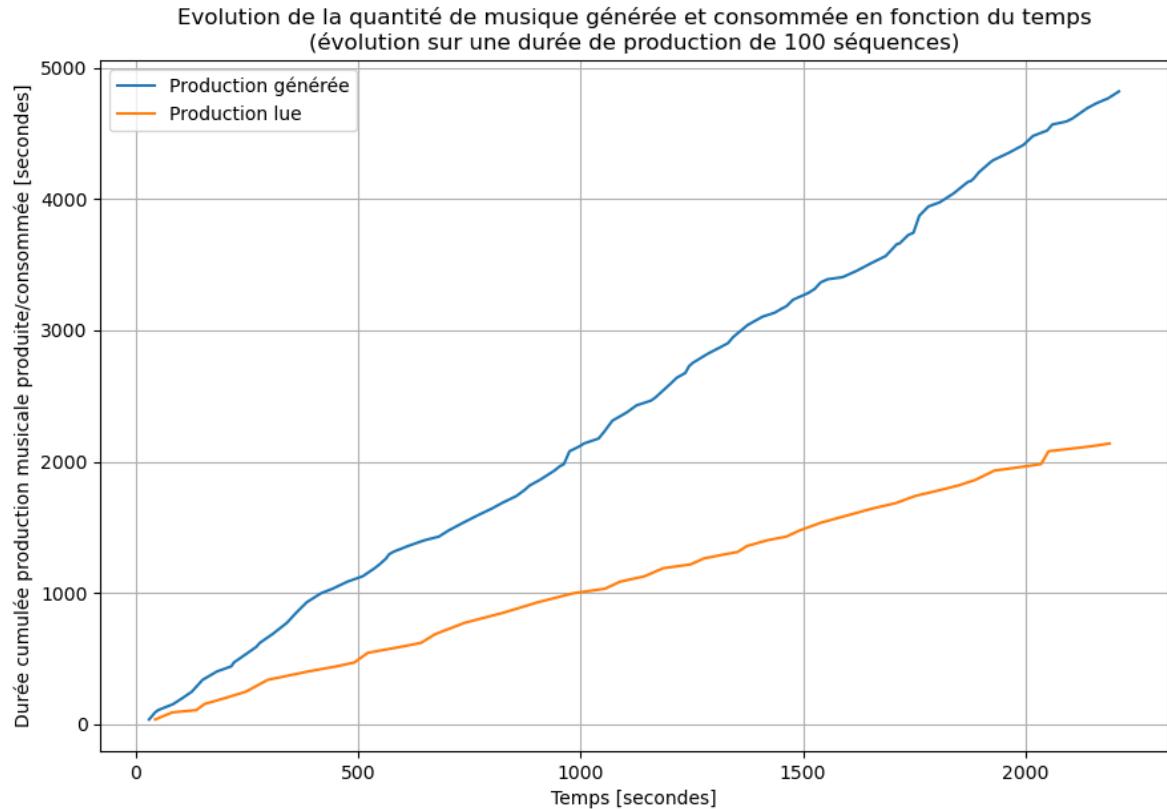


FIGURE 6.1 – Évolution de la quantité de musique générée et consommée en fonction du temps.

On peut remarquer dans ce graphe différentes choses. Tout d'abord, dans l'ensemble, la génération se réalise plus vite que la lecture, ce qui permet d'accéder au cours de la lecture à une nouvelle séquence à chaque fin de séquences. La mesure de robustesse consistant à boucler sur la dernière séquence ne s'est donc pas montré utile, lors de l'utilisation du générateur, cela étant illustré par la figure 6.1. En effet, l'utilisation de cette mesure de robustesse se serait marquée d'un plateau horizontal dans la courbe orange, car le lecteur réexploiterait des données plutôt que d'en consommer des nouvelles. Une autre mesure de robustesse est quant à elle visible sur le graphe, il s'agit du départ retardé de la lecture. Il peut ainsi être observé que le lecteur ne commence sa consommation qu'après deux apports de temps musical par la partie génératrice. Il est par ailleurs possible de comprendre l'intérêt de cette robustesse, car c'est bien au début de l'utilisation que la situation est la plus critique. Enfin, une

dernière observation repose sur l'allure des courbes. Comme il est possible de le voir, il y a des variations de pente tout au long du tracé. Ces modifications sont liées à la longueur fluctuante des séquences générées. Cette fluctuation peut d'ailleurs également se vérifier sur certaines moyennes du générateur. Ainsi pour l'exemple de l'utilisation liée à ce graphe, le temps de génération moyen après 10 séquences générées était de 20.8 secondes pour des séquences produites de longueur moyenne de 40.2 secondes. Après 50 générations, ces deux valeurs montent respectivement à 22.2 secondes et 54.4 secondes. Et enfin, après 100 générations, le système est à un temps moyen de génération de 21.9 secondes pour des séquences de durée moyenne de 48.62 secondes (données provenant de la console d'exécution du générateur intelligent). Ce point précis représente à la fois un avantage et un désavantage. En effet, ces longueurs fluctuantes apportent du naturel dans la lecture d'une série de séquences, car dans les performances composées par l'Homme, il en est également ainsi. Cependant ces fluctuations peuvent engendrer des remontées plus ou moins rapides du lecteur (partie lecture) sur le générateur (partie génération), et ce tout particulièrement en début de session d'utilisation. Mais grâce aux mesures de robustesse, ces aspects ne représentent plus un problème. Un dernier point d'intérêt sur la figure 6.1 est également sa constance. De fait, le système ne s'emballe pas et reste stable ; cette caractéristique étant bien évidemment appréciée.

Enfin, avant de passer à d'autres métriques, il est important de relever le contexte de ce graphe. La session d'utilisation du générateur a en effet été réalisée sur CPU et non sur GPU. De plus, le CPU en question est un « Intel Core i7-7500U @ 2.70GHz (4 CPUs), 2.9GHz ». Ce processeur n'est donc pas à la pointe de la technologie actuelle et est, de plus, âgé de 5 ans au moment du test. L'utilisation a donc été faite sur un support un peu daté et de performance moyenne, mais présente tout de même de très bons résultats. Cette indication est donc très encourageante pour les possibilités embarquées et modulaires du générateur.

6.2 Appréciation humaine et test qualitatif

Comme dans une majorité d'études dans ce domaine, un passage clé est celui de l'oreille humaine. Ce travail ne fait pas exception et un sondage basé sur l'écoute a ainsi été mis en place. Celui-ci s'inspire principalement d'une approche effectuée par E. Dervakos, G. Filandrianos et G. Stamou dans leur article portant sur les méthodes d'évaluation pour la génération par IA [4]. Il a ainsi été demandé aux participants de donner deux notes sur 10 à chaque séquence. La première de ces notes porte simplement sur l'appréciation de la séquence par le participant. La seconde note, elle, est une note attribuée pour l'intérêt du morceau et la musicalité perçue de la séquence. Enfin, de manière à rendre le tout un peu plus ludique et challengeant, les participants devaient deviner s'il s'agissait ou non d'une production humaine. Par la même occasion, le test interrogeait sur la définition de la musicalité et les critères employés par le participant pour différencier « le vrai du faux » (le formulaire distribué est visible en annexe D).

Pour ce qui est de la constitution du test proposé, ce dernier était constitué de 5 types de séquences : des séquences générées par Music Transformer (et donc similaires aux productions du générateur), des séquences générées par Performance RNN, des séquences issues de la base

de données MAESTRO¹ (et qui sont donc des séquences jouées par des humains), des extraits de flux du générateur (qui sont donc des enchaînements de différentes séquences) et une vidéo montrant l'utilisation du générateur combiné au Probatio. A noter que pour ne pas orienter les opinions, toutes les séquences à l'exception des extraits du flux et la vidéo ont été prises de taille similaire (plus ou moins une minute) et ont été réenregistrées de la même manière. La qualité du son ainsi baissée permet de ne pas compromettre les résultats. Cette démarche a été définie comme utile, car la qualité sonore de la synthèse influence de manière significative les participants². Ajouté à cela, d'autres approches ont été explorées, comme l'utilisation de divers synthétiseurs (toutes les données étant sous format MIDI).

Les moyennes des scores obtenus pour chacun des morceaux du test sont disponibles dans les figures 6.2 et 6.3, respectivement dédiées à l'appréciation et à la note d'intérêt/musicalité. Différentes données d'analyse statistiques y sont également ajoutées pour rendre le tout plus parlant (la donnée « Somme » est une simple addition de toutes les notes qui ont été données par les participants). Le tableau 6.1, quant à lui illustre le score obtenu par séquence en termes de perception. A noter que 1 point est attribué par participant trouvant correctement s'il s'agit d'une génération d'IA ou d'un humain et qu'il y avait 33 participants.

De ces données, plusieurs choses sont à extraire. Tout d'abord, il est assez amusant de jeter un oeil à la note minimal et maximal de chaque séquence. Il peut alors être remarqué que presque chaque séquence a eu au moins une personne la trouvant vraiment bien ou à l'inverse vraiment mauvaise. Et bien que ces deux données soient très basiques, cela plante le décor de l'approche qualitative : l'unanimité ne fait pas loi. Une fois les séquences classées par moyenne, une certaine cohérence apparaît. Globalement, on peut remarquer que les séquences dans le classement sont clairement discriminées dans le cas de IA1 qui correspond à des séquences générées par Performance RNN. Dans le dessus du classement, on voit par contre que les places se partagent entre l'humain et IA2 (qui est Music Transformer). De plus, si on observe le podium, si la première place est gardée par l'humain, la deuxième place est détenue par Music Transformer. La séquence 10 a ainsi énormément plu et on peut aussi remarquer qu'elle possède un score de perception très faible. Par conséquent, pour la grande majorité des participants, ce top est humain, alors que ce n'est pas du tout le cas. Les positions des deux séries et de la vidéo sont également intéressantes. On peut ainsi remarquer que dans l'ensemble, elles ont plu. Pour ce qui est des scores de perception, on remarque que dans l'ensemble, l'IA2 se débrouille assez bien, que les humains restent assez identifiables et que l'IA1 est clairement à la traîne. A titre de comparaison, le score moyen de perception est de 10.8 séquences sur 21.

Bien que ces données soit parlantes, elles le deviennent d'autant plus en traitant les résultats par source, comme dans le tableau 6.2. Dans ce tableau, il est clairement identifiable que

1. MAESTRO (MIDI and Audio Edited for Synchronous TRacks and Organization) est un ensemble de données composé d'environ 200 heures de performances de virtuoses du piano capturées avec un alignement fin (3 ms) entre les étiquettes des notes et les formes d'ondes audio. Ces données ont été récupérées dans le cadre de concours de piano et reflètent donc un très bon niveau de jeu.

2. Ceci peut être affirmé, car dans les divers sondages réalisés dans le cadre de ce travail, une question portait sur les critères que les personnes appliquaient pour départager ce qu'ils pensaient être issu d'une IA ou d'un humain. Dans ces réponses, il était alors possible de retrouver une partie non négligeable de personnes se laissant influencer par la qualité du son. Ce critère n'étant pas un critère juste, il a été jugé judicieux de volontairement l'écartier.

Nom	Identité réelle	Score de perception par séquence
1	Humain	21
2	IA 1	21
3	IA 2	17
4	IA 2	20
5	Humain	17
6	Humain	24
7	IA 1	26
8	IA 1	24
9	IA 2	12
10	IA 2	5
11	IA 2	26
12	Humain	5
13	IA 1	17
14	IA 2	7
15	Humain	27
16	Humain	17
17	IA 1	16
18	IA 2	17
19	IA 1	25
Serie_1	IA 2	14
Serie_2	IA 2	20

TABLE 6.1 – Score de perception par séquence.

	♦ Identité	♦ Synthétiseur	▼ Moyenne	♦ Mediane	♦ Ecart-type	♦ Min	♦ Max	♦ Somme
15	Humain	Synth. interne	7.79412	8.00000	1.34343	3	10	265
10	IA 2	Piano	7.70588	8.00000	1.03072	5	10	262
Video	/	Autre	7.58824	8.00000	1.51992	4	10	258
18	IA 2	Synth. interne	7.41176	8.00000	1.28199	5	10	252
6	Humain	Piano	7.23529	7.00000	1.15624	5	10	246
9	IA 2	Ceremony	6.97059	7.00000	1.35926	3	10	237
S_1	IA 2	Piano	6.82353	7.00000	1.08629	5	9	232
S_2	IA 2	Piano	6.64706	6.00000	1.32304	4	10	226
1	Humain	Piano	6.64706	7.00000	1.85672	2	10	226
14	IA 2	Piano	6.52941	7.00000	1.48192	4	10	222
3	IA 2	Synth. interne	6.47059	6.50000	1.61874	2	10	220
5	Humain	Ceremony	6.20588	7.00000	1.87131	1	10	211
16	Humain	Analog	5.91176	6.00000	2.12300	1	9	201
4	IA 2	Kalimba	5.67647	6.00000	1.73591	3	9	193
7	IA 1	Analog	5.50000	5.50000	1.84637	0	10	187
17	IA 1	Piano	5.44118	6.00000	1.89403	0	8	185
11	IA 2	Analog	5.20588	5.00000	1.75429	1	10	177
8	IA 1	Kalimba	4.79412	5.00000	1.87131	0	9	163
12	Humain	Kalimba	4.73529	5.00000	1.95880	0	9	161
2	IA 1	Piano	4.64706	5.00000	1.92090	0	8	158
13	IA 1	Synth. interne	4.61765	5.00000	2.36149	0	10	157
19	IA 1	Ceremony	3.70588	4.00000	2.00801	0	8	126

FIGURE 6.2 – Informations déduites des résultats du test d’appréciation.

	Moyenne appré.	Moyenne int./mus.	Ecart-type moyen appré.	E-type moyen int./mus.
Humain	6.42	6.36	163.8	164.7
Perf. RNN	4.78	4.59	198.4	198.8
Music Trans.	6.6	6.45	146.6	148

TABLE 6.2 – Moyenne globale (note sur 10) et écart-type par type de source

Music Transformer est un très bon modèle. De plus, l’analyse de l’écart-type nous annonce que en plus d’avoir une bonne moyenne, ce score fait relativement l’unanimité. En effet, cet écart-type étant plus faible, cela signifie qu’il y a moins d’oscillations autour de la moyenne. Une autre donnée très importante pour nous est le cas des séries puisqu’elles traduisent le comportement du générateur en temps que tel. Ces deux séries ont ainsi comme moyenne regroupée et en écart-type, la note de 6.7/10 et 120.5 pour l’appréciation, la note de 6.7/10 et 133.5 pour l’intérêt/musicalité (au plus l’écart-type est grand, au plus les participants ont tendance à mettre des notes éloignées de ladite moyenne). Ce qui les place, comme attendu, sur le haut du podium.

Enfin, il est intéressant de manière complémentaire de s’intéresser au cas des synthétiseurs sélectionnés, à travers le tableau 6.3. Ici, il peut clairement être identifié que les synthétiseurs un peu plus originaux ont une moyenne nettement inférieure relativement à la synthèse interne (similaire à un piano basique) et au piano. De plus, de par leur écart-type plus élevé, il est possible de comprendre que ces synthétiseurs plus originaux font moins l’unanimité.

	♦ Identité	♦ Synthétiseur	▼ Moyenne	♦ Mediane	♦ Ecart-type	♦ Min	♦ Max	♦ Somme
15	Humain	Synth. interne	7.91176	8.00000	1.08342	6.00000	10.00000	269.00000
10	IA 2	Piano	7.55882	8.00000	1.07847	5.00000	9.00000	257.00000
18	IA 2	Synth. interne	7.26471	8.00000	1.39933	2.00000	9.00000	247.00000
Video	/	Autre	7.14706	7.50000	1.94051	3.00000	10.00000	243.00000
6	Humain	Piano	7.05882	7.00000	1.04276	5.00000	9.00000	240.00000
S_1	IA 2	Piano	6.76471	7.00000	1.18216	5.00000	9.00000	230.00000
S_2	IA 2	Piano	6.70588	7.00000	1.48792	2.00000	10.00000	228.00000
9	IA 2	Ceremony	6.67647	7.00000	1.45061	4.00000	10.00000	227.00000
14	IA 2	Piano	6.41176	7.00000	1.65360	3.00000	10.00000	218.00000
1	Humain	Piano	6.38235	7.00000	1.98502	1.00000	10.00000	217.00000
5	Humain	Ceremony	6.16176	6.00000	1.72645	1.00000	9.00000	209.50000
16	Humain	Analog	6.11765	6.00000	2.10000	1.00000	9.00000	208.00000
3	IA 2	Synth. interne	6.11765	6.00000	1.51287	3.00000	9.00000	208.00000
4	IA 2	Kalimba	5.50000	5.00000	1.58114	3.00000	9.00000	187.00000
7	IA 1	Analog	5.35294	5.50000	2.04321	0.00000	9.00000	182.00000
17	IA 1	Piano	5.26471	6.00000	1.97421	0.00000	8.00000	179.00000
11	IA 2	Analog	5.11765	5.00000	1.68352	2.00000	10.00000	174.00000
12	Humain	Kalimba	4.52941	5.00000	1.94212	0.00000	8.00000	154.00000
8	IA 1	Kalimba	4.50000	4.00000	1.82989	1.00000	8.00000	153.00000
2	IA 1	Piano	4.44118	4.00000	1.95698	1.00000	9.00000	151.00000
13	IA 1	Synth. interne	4.38235	4.00000	2.22948	0.00000	10.00000	149.00000
19	IA 1	Ceremony	3.61765	4.00000	1.89120	0.00000	7.00000	123.00000

FIGURE 6.3 – Informations déduites des résultats du test d’intérêt/musicalité.

	Kalimba	Analog	Ceremony	Synth. Interne	Piano
Moyenne d’appréciation	5,06862745	5,53921569	5,62745098	6,57352941	6,36764706
Moyenne d’intérêt	4,84313725	5,52941176	5,4818776	6,41911765	6,18627451
Ecart-type moyen d’appréc.	185.534	190.789	174.619	165.141	155.676
Ecart-type moyen d’intérêt	178.438	194.224	168.942	155.628	161.517

TABLE 6.3 – Influence du synthétiseur sur la moyenne (note sur 10) et l’écart-type.

Il y aura donc des appréciations plus variées chez les testeurs avec de telles méthodes de synthèse. Si fondamentalement cette information ne semble pas des plus utiles, elle nous indique clairement que le choix d’un synthétiseur ou d’un autre peut complètement biaiser des résultats. Quelqu’un de peu soucieux de l’éthique professionnel pourrait ainsi parfaitement biaiser des résultats en changeant simplement de synthèse audio.

6.3 Analyse quantitative du générateur

Comme il a pu être expliqué dans la section traitant des problématiques d’évaluation, trouver des métriques pertinentes est une tâche assez complexe. Néanmoins, issu de [28], voici quelques-unes des caractéristiques de base qu’il est possible de comparer pour se faire une avis objectif sur la génération. Il est plus précisément question ici de comparer la production de Music Transformer avec la base de données MAESTRO (fig. 6.4).

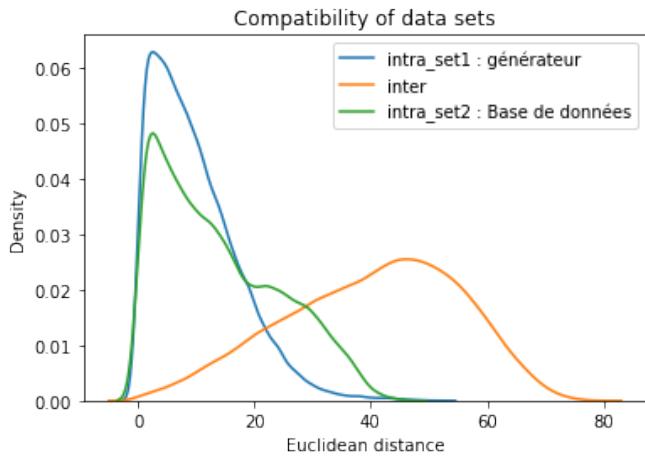


FIGURE 6.4 – Évaluation de la compatibilité des sets générés et de la base de données MAESTRO.

L'intérêt de cette analyse est de voir à quel point la génération est proche d'une base de données différente de celle d'entraînement. Cette base de données étant de plus basée sur des concours de piano, s'en rapprocher signifie que la qualité apparente de la production du générateur se rapproche de celle d'un jeu humain lors d'un tel concours. Ce qui serait évidemment appréciable. Or, bonne nouvelle, c'est ce qui se passe. Comme il peut être vu sur la figure 6.4, le match n'est pas parfait, ce qui est normal, mais les données restent concordantes. Ces résultats reposants sur l'utilisation de métriques de [28], le lecteur intéressé pourra y trouver plus de détails. Mais de manière concise, l'analyse se base sur le nombre de notes différentes dans un morceau. Ici, les variations internes des bases de données sont voisines et donc assez similaires. Par contre, la comparaison inter-set (variation du nombre de notes différentes par morceaux à l'intérieur d'un set) montre une différence moyennement prononcée.

6.4 Analyse critique

De toutes ces données ressortent différentes conclusions plus importantes que d'autres. Tout premièrement, il ressort que le générateur est pertinent dans son fonctionnement comme dans sa production. De par l'évolution de la production de la partie génératrice par rapport à l'évolution de la lecture sur la musique générée, la parallélisation s'illustre fonctionnelle. Sur un ordinateur daté (vieillissant), le système ne montre même pas la nécessité d'utiliser des méthodes supplémentaires de robustesse, c'est à dire le bouclage automatique. Ceci est très encourageant aussi bien pour l'aspect embarqué que modulaire du générateur. En effet, il apparaît que le générateur pourrait relativement aisément être placé dans différents supports d'utilisation et ne fermerait pas les possibilités d'être utilisé avec de potentiels autres logiciels sur un même appareil. Pour ce qui est de sa production, le générateur s'illustre également comme pertinent puisque cette nouvelle utilisation du réseau Music Transformer [11] n'engendre pas de perte en termes de qualité perçue. De fait, lors du test de perception humaine, les enchaînements de séquences, compris dans le flux des sorties du générateur, n'ont pas choqué à l'audition. Au contraire, ces enchaînements ont même augmenté la note des résultats qualitatifs

relatifs à Music Transformer. Bien évidemment, ce fait est extrêmement positif et valide l'intérêt et la pertinence de la génération de flux continus. De plus, l'utilisation du générateur couplé au Probatio a également su susciter l'intérêt des testeurs, ceux-ci ayant attribués de très bonnes notes à l'exemple fourni dans le test. Là encore, les résultats sont très encourageants sur la pertinence de cette approche.

Chapitre 7

Perspectives d'amélioration

De ces résultats, il est à tirer que le générateur tel qu'il est actuellement est opérationnel, mais (comme toutes choses) n'est pas parfait. Des possibilités d'amélioration sont encore possibles. De plus, le générateur n'a encore été que peu utilisé et des perspectives d'avenir encore non atteintes pourraient bien se révéler avec la pratique de ce dernier.

Dans un premier temps, de simples améliorations en termes de confort d'utilisation pourraient s'ajouter au projet. Il serait par exemple parfaitement concevable d'adoindre au générateur une interface graphique et de le rendre plus proche d'un exécutable. De cette façon, il serait possible d'entièrement supprimer la vue du code à l'utilisateur. Dans un second temps, le POC du système embarqué sur Raspberry Pi pourrait être renforcé par l'ajout de moyens d'interaction avec lui, que ce soit boutons, encodeurs rotatifs, écrans ou autres.

Les autres grandes voies d'amélioration seraient davantage liées aux réseaux de neurones. En effet, les alternatives telles que Pop Music Transformer sont toujours à explorer et pourraient se montrer très intéressantes.

Enfin, pour ce qui est des perspectives, il serait bon d'exploiter le générateur dans différents cadres applicatifs : cours d'improvisation, outils de test, etc. De cette manière, il serait possible d'adapter plus précisément le générateur aux besoins de ces utilisateurs grâce à leurs retours.

Toutefois, n'oublions pas que le générateur s'inscrit dans le cadre du projet Probatio. Son rôle est donc d'ouvrir des portes aux explorateurs et non pas d'explorer. C'est à dire que le rôle de la boîte à outils est de fournir de la « matière brute » pour que des utilisateurs puissent expérimenter au grès de leur désir. Le Probatio est un fournisseur de possibilités. Il est ainsi fort à parier que de nouvelles utilisations pourraient se révéler dans un avenir proche.

Conclusion

En conclusion de ce travail, il peut être avancé qu'un générateur intelligent, modulaire et embarqué a bien été créé. Celui-ci se basant sur l'utilisation du réseau de neurones Music Transformer, il a su apporter une plus value et un nouveau champ applicatif à ce type de réseaux. En effet, jusqu'alors ce type de réseaux travaillait de manière discrète. Il s'agissait de créer une séquence et puis de travailler dessus. Ici, la logique est de faire tourner la génération en permanence de façon à obtenir un flux continu. Plus précisément, grâce à sa structure, parallélisant génération, lecture et menu, il est possible d'obtenir un flux MIDI sur lequel l'utilisateur garde la main.

Dans son principe de fonctionnement, ce travail propose donc une nouvelle approche répondant par la même occasion aux deux questions maîtresses du Probatio. En effet, le générateur est à la fois la proposition d'une toute nouvelle approche de la musique pour la boîte à outils et une manière de baisser le temps de prototypage. C'est à dire qu'un utilisateur désireux d'expérimenter trouvera en ce nouveau « bloc » du Probatio un terrain riche en possibilités. En termes d'outils, le générateur implémenté peut ainsi aussi bien servir de supports pour réaliser de petits tests rapides comme représenter un support d'improvisation riche et complet. Pour ce qui est du gain de temps de prototypage, le générateur a été initialement développé pour lutter contre la nécessité d'installation complète d'instruments « classiques » dans tout contexte où le Probatio est vu comme un outil de création pour un module annexe à adjoindre sur un instrument existant (ce cadre d'utilisation est nommé augmentation d'instruments). Il est important de comprendre qu'une installation d'instruments est très chronophage, or dans un premier temps, tout ce que désire un utilisateur est de pouvoir toucher/essayer. En prototypage, on ne cherche pas à toucher du doigt l'aboutissement final du premier coup. Ce n'est d'ailleurs pas dans cette optique qu'a été créé le Probatio. Par contre, dans un contexte de premier pas, le générateur se montre très pratique. En effet, comme sa sortie est en MIDI, n'importe quelle synthèse peut être placée en bout de queue, voir même plusieurs pour un seul flux.

Ce générateur a également été pensé pour être modulaire et embarqué, de manière à pouvoir être utilisé dans le plus de contextes possibles. A cette fin, un Proof-of-Concept a pu être réalisé avec un Raspberry Pi. Ce POC a notamment permis de mettre en évidence la possibilité d'une adaptation du code pour une génération en ligne. Cette configuration a également mis en exergue l'intérêt qu'elle avait de faire tomber toutes les contraintes de performance embarquée de la carte, à condition de disposer d'une connexion Internet. Un patch Pure-Data a également pu être conçu de façon à gérer de manière complètement locale (sur une carte de type Raspberry) les interactions avec les blocs du Probatio. Pour plus de modularité et d'aisance dans l'utilisation, le générateur est également doté d'un système de synthèse audio

interne. Ce dernier peut être vu comme une option de débogage embarquée et est très indiquée dans des contextes de tests rapides.

De manière à vérifier la véracité de l'intérêt du générateur, de multiples tests ont pu être réalisés. Des sessions d'enregistrement ont pu être réalisées avec et sans le Probatio de manière à tester en conditions réelles le générateur. Certaines de ces performances ont également été utilisées pour créer un sondage permettant d'évaluer la pertinence du travail. De ce sondage ayant eu le retour d'une trentaine de personnes sont ressorties des données très encourageantes en ce qui concerne la qualité perçue d'un flux musical. Combiné à des analyses de performance, il en ressort que le générateur est tout à fait fonctionnel et pertinent, notamment de par sa robustesse. De cette façon, cet outil répond de manière qualitative aux questions de recherche du Probatio, et fournit de nouvelles fonctionnalités pour les réseaux générateurs de fichiers.

De par son bon fonctionnement, il est à l'heure actuelle tout à fait concevable d'utiliser ce générateur même en dehors du cas du Probatio. Ce projet possède d'intéressantes perspectives et pourrait se profiler comme un outil d'aide à la création dans de multiples contextes. Il est également à signaler que d'autres types de réseaux sont encore en cours de développement aujourd'hui et les intégrer à l'avenir pourrait engendrer des solutions des plus captivantes.

Bibliographie

- [1] Filipe CALEGARIO. *Designing Digital Musical Instruments Using Probatio. A Physical Prototyping Toolkit*. Springer Cham, 2019.
- [2] Kristy CHOI et al. *Encoding Musical Style with Transformer Autoencoders*. Department of Computer Science, Stanford University, 3 juin 2020.
- [3] Zihang DAI et al. *Transformer-XL : Attentive Language Models Beyond a Fixed-Length Context*. Carnegie Mellon University, 2 juin 2019.
- [4] Edmund DERVAKOS, Giorgos FILANDRIANOS et Giorgos STAMOU. *Heuristics for Evaluation of AI Generated Music*. School of Electrical et Computer Engineering National Technical University of Athens, 15 jan. 2021.
- [5] Hao-Wen DONG et Yi-Hsuan YANG. *Convolutional Generative Adversarial Networks with Binary Neurons for Polyphonic Music Generation*. Research Center for IT innovation, Academia Sinica, Taipei, Taiwan, 6 oct. 2018.
- [6] Hao-Wen DONG et al. *MuseGAN : Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment*. Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan, 24 nov. 2017.
- [7] Stephano FASCIANI et Jackson GOODE. *20 NIMEs : Twenty Years of New Interfaces for Musical Expression*. Département de musicologie, Université d’Oslo, 29 avr. 2021. URL : <https://nime.pubpub.org/pub/20nimes/release/1?readingCollection=71dd0131> (visité le 20/05/2022).
- [8] Jon GILLICK, Adam ROBERTS et Jesse ENGEL. *Groovae : Generating and controlling expressive drum performances*. 2 mai 2019. URL : <https://magenta.tensorflow.org/groovae> (visité le 28/05/2022).
- [9] Jon GILLICK et al. *Learning to Groove with Inverse Sequence Transformations*. School of Information, University of California, Berkeley, U.S.A, 26 juill. 2019.
- [10] Curtis HAWTHORNE et Erich ELSSEN. *Onsets and Frames : Dual-Objective Piano Transcription*. 12 fév. 2018. URL : <https://magenta.tensorflow.org/onsets-frames> (visité le 03/06/2022).
- [11] Cheng-Zhi Anna HUANG et al. *Music Transformer : Generating Music with Long-Term Structure*. Google Brain, 12 déc. 2018.
- [12] Yu-Siang HUANG et Yi-Hsuan YANG. *Pop Music Transformer : Beat-based Modeling and Generation of Expressive Pop Piano Compositions*. Taiwan AI Labs et Academia Sinica Taipei, Taiwan, 10 août 2020.

- [13] Hsiao-Tzu HUNG et al. *Improving Automatic Jazz Melody Generation by Transfer Learning Techniques*. Institute of Information Science, Academia Sinica, Taipei, Taiwan, 26 août 2019.
- [14] IFPI. *Global Music Report 2022*. 22 mars 2022. URL : https://cms.globalmusicreport.ifpi.org/uploads/Global_Music_Report_State_of_The_Industry_5650fff4fa.pdf (visité le 20/05/2022).
- [15] Shulei JI, Jing LUO et Xinyu YANG. *A Comprehensive Survey on Deep Music Generation : Multi-level Representations, Algorithms, Evaluations, and Future Directions*. School of Computer Science et Technology, Xi'an Jiaotong University, China, 13 nov. 2020.
- [16] *Libmapper*. URL : <http://www-new.idmil.org/project/libmapper/> (visité le 30/05/2022).
- [17] Joseph MALLOCH et Marcelo WANDERLEY. *Embodied Cognition and Digital Musical Instruments : Design and Performance*. Université McGill, 13 jan. 2017.
- [18] Joseph MALLOCH et al. *Towards a new conceptual framework for digital musical instruments*. IDMIL, Université McGill, 18 sept. 2006.
- [19] Jens RASMUSSEN. *Information Processing and Human-Machine Interaction : An Approach to Cognitive Engineering*. Elsevier Science Inc, New York, USA, 1^{er} sept. 1986.
- [20] Yi REN et al. *PopMAG : Pop Music Accompaniment Generation*. Zhejiang University, 18 août 2020.
- [21] Ian SIMON et Sageev OORE. *Performance RNN : Generating Music with Expressive Timing and Dynamics*. 29 juin 2017. URL : <https://magenta.tensorflow.org/performance-rnn> (visité le 28/05/2022).
- [22] *Slapbox*. URL : <http://www-new.idmil.org/project/slapbox/> (visité le 23/05/2022).
- [23] Le SNEP. *Le CNM publie les certifications export CNM / SNEP – ANNEE 2021*. 27 avr. 2022. URL : <https://snepmusique.com/chiffres-ressources/le-cnm-publie-les-certifications-export-cnm-snep-2021-annee-2021/> (visité le 20/05/2022).
- [24] Google Brain TEAM. *Magenta Studio (v1.0)*. URL : <https://magenta.tensorflow.org/studio> (visité le 30/05/2022).
- [25] *The T-Stick*. URL : <http://www-new.idmil.org/project/the-t-stick/> (visité le 23/05/2022).
- [26] Johnty WANG et al. *Webmapper : A Tool for Visualizing and Manipulating Mappings in Digital Musical Instruments*. Input Devices et Music Interaction Laboratory, CIRMMT, McGill University, 14 oct. 2019.
- [27] Ziyu WANG et al. *PIANOTREE VAE : Structured Representation Learning for Polyphonic Music*. Music X Lab, Computer Science Department, NYU Shanghai, 17 août 2020.
- [28] Li-Chia YANG et Alexander LERCH. *On the evaluation of generative models in music*. Center for Music Technology, Georgia Institute of Technology, Atlanta, USA, 3 nov. 2018.

Annexes

Annexe A

Détails sur Music Transformer

Cette annexe a pour but de délivrer les détails sur Music Transformer et les traite par point (ces informations sont issues de l'article du même nom [11]).

A.1 Représentation des données

La représentation des données majoritairement utilisées est un vocabulaire « MIDI-like ». On y retrouve 128 événements NOTE_ON, 128 NOTE_OFF, 100 TIME_SHIFTS permettant un timing expressif à 10ms et 32 niveaux de VELOCITÉ pour une dynamique expressive. Une alternative a aussi été utilisée pour la base de données « JSB Chorale » qui se présentait sous la forme d'une matrice où les lignes correspondent aux voix et les colonnes au temps discrétisé en doubles croches. Quoiqu'il en soit, la musique est représentée dans tous les cas comme une séquence d'éléments discrets (des tokens) dont le vocabulaire est déterminé par la base de données.

A.2 Self-attention dans les transformers

La self-attention est le mécanisme au centre du modèle de décodeur Transformer. Celui-ci est autoregressif et utilise principalement des informations de positions apprises ou sinusoïdales en plus de la self-attention. De manière plus détaillée, chaque couche est constituée d'une sous-couche d'auto-attention suivie d'une sous-couche de feedforward. La couche d'attention transforme d'abord une séquence de L vecteurs $X = (x_1, x_2, \dots, x_L)$ à D dimensions en requêtes $Q = XW^Q$, en clés $K = XW^K$ et en valeurs $V = XW^V$, où W^Q , W^K et W^V sont chacunes des matrices carrées de dimensions $D * D$. Chaque matrice de requêtes, de clés et de valeurs $L * D$ est ensuite divisée en $HL * D_h$ parties ou têtes d'attention, indexées par h , et de dimension $D_h = D/H$, qui permet au modèle de se concentrer sur différentes parties de l'historique. Sur ces bases, la séquence de sorties vectorielles pour chaque tête est calculée sur base de :

$$Z^h = \text{Attention}(Q^h, K^h, V^h) = \text{Softmax}\left(\frac{Q^h K^{hT}}{\sqrt{D_h}}\right) V^h. \quad (\text{A.1})$$

Les sorties d'attention pour chaque tête sont concaténées et transformées de façon linéaire pour obtenir Z , une matrice de dimension L par D . Un masque triangulaire supérieur garantit que les requêtes ne peuvent pas s'occuper des clés plus tard dans la séquence. La sous-couche d'anticipation (FF) prend ensuite la sortie Z de la sous-couche d'attention précédente, et la

passe dans deux couches selon l'équation (où W_1 , W_2 et b_1 , b_2 sont respectivement les poids et les biais des deux couches) :

$$FF(Z) = \text{ReLU}(ZW_1 + b_1)W_2 + b_2. \quad (\text{A.2})$$

A.3 Self-attention et position relative

Comme présenté précédemment, cet article introduit la notion de position relative de manière à s'adapter au cas applicatif de la musique. Cela implique l'apprentissage d'un encodage de position relative séparé E^r de forme (H, L, D_h) , qui comporte un encodage pour chaque distance par paire possible entre une requête et une clé. Les incorporations sont ordonnées de la distance $-L + 1$ à 0 , et sont apprises séparément pour chaque tête. Cette considération relative des positions apporte cependant avec elle une complexité de $O(L^2D)$. Cette dernière est trop grande pour le traitement efficace des données et c'est pourquoi les auteurs proposent une amélioration algorithmique permettant de redescendre la complexité à $O(LD)$.

Ces améliorations algorithmiques étant très spécifiques à ce réseau, cette annexe n'en traite pas. Le lecteur intéressé pourra cependant retrouver ces informations dans [11]. A noter également que cette amélioration vient aussi avec une retouche des notions d'attention via l'utilisation d'attention locale relative.

Annexe B

Lignes chronologiques des différentes générations musicales

Voici quelques lignes du temps montrant la variété des développements dans le domaine de la Deep Generation. Toutes ces lignes sont issues de [15].

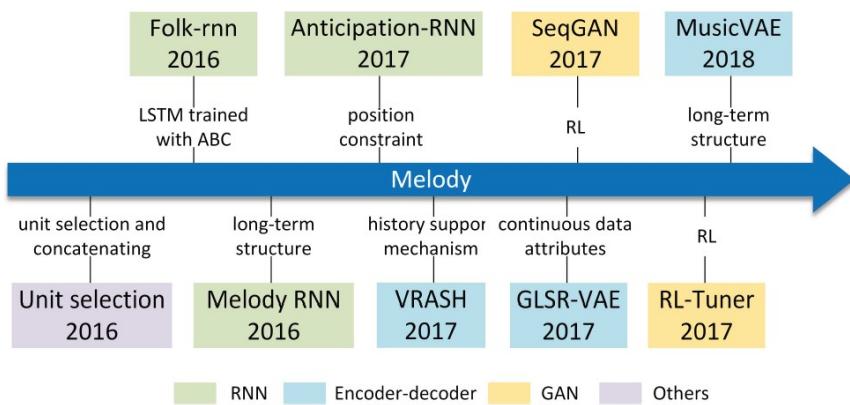


FIGURE B.1 – Ligne du temps : génération de partition monophonique.

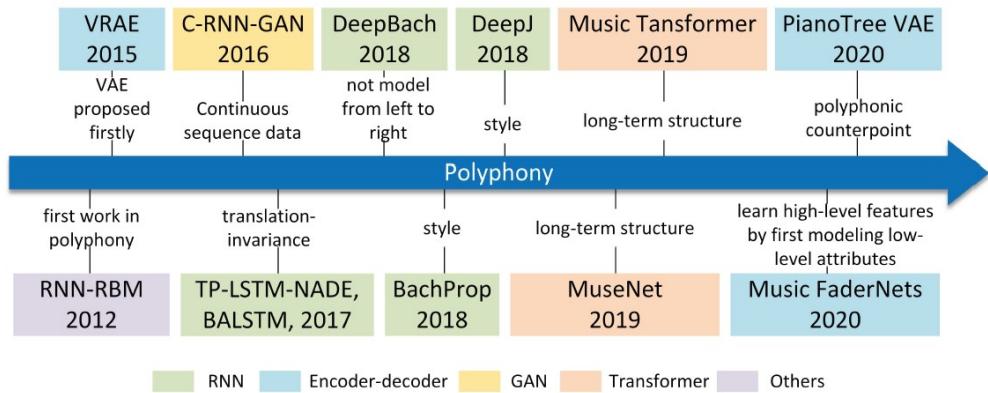


FIGURE B.2 – Ligne du temps : génération de partition polyphonique.

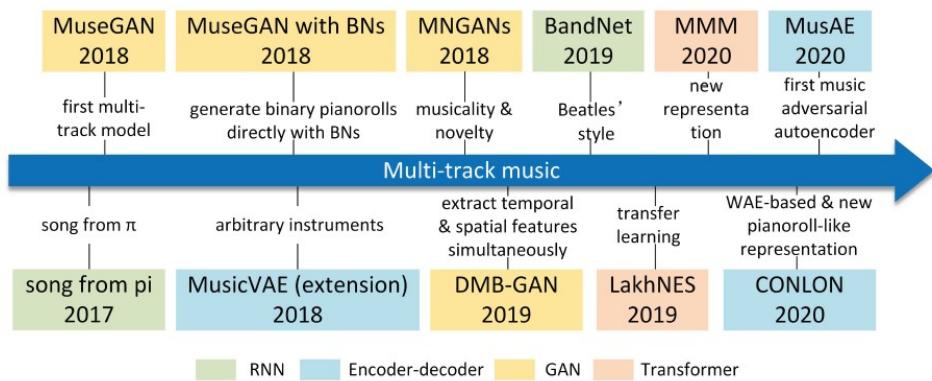


FIGURE B.3 – Ligne du temps : génération de partition multi-track.

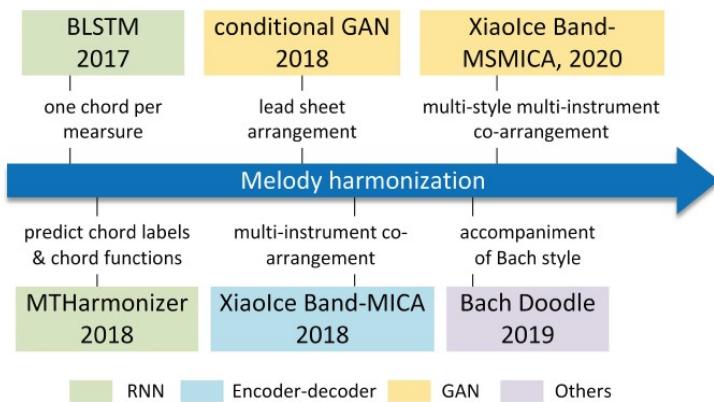


FIGURE B.4 – Ligne du temps : harmonisation de mélodie.

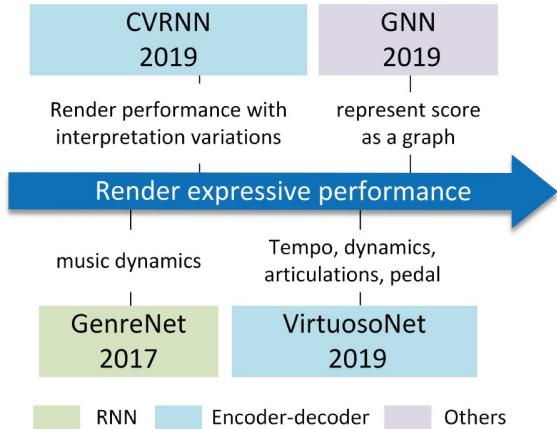


FIGURE B.5 – Ligne du temps : génération de caractéristiques de performance.

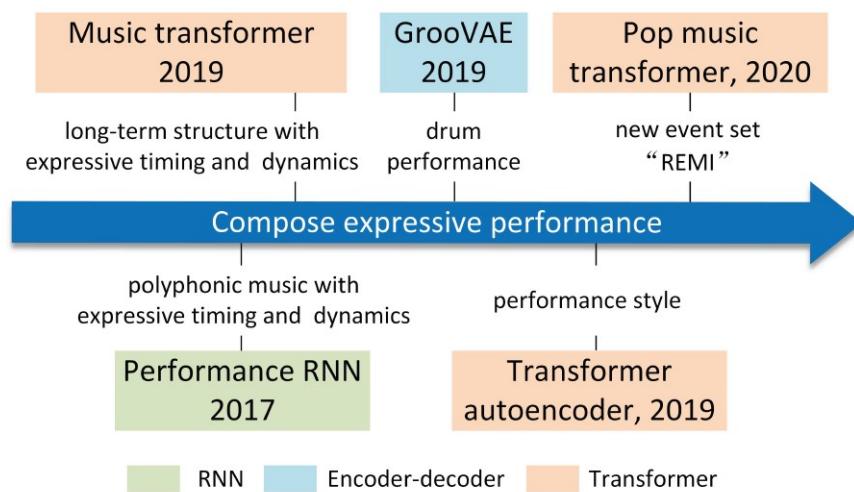


FIGURE B.6 – Ligne du temps : génération de performance expressive.

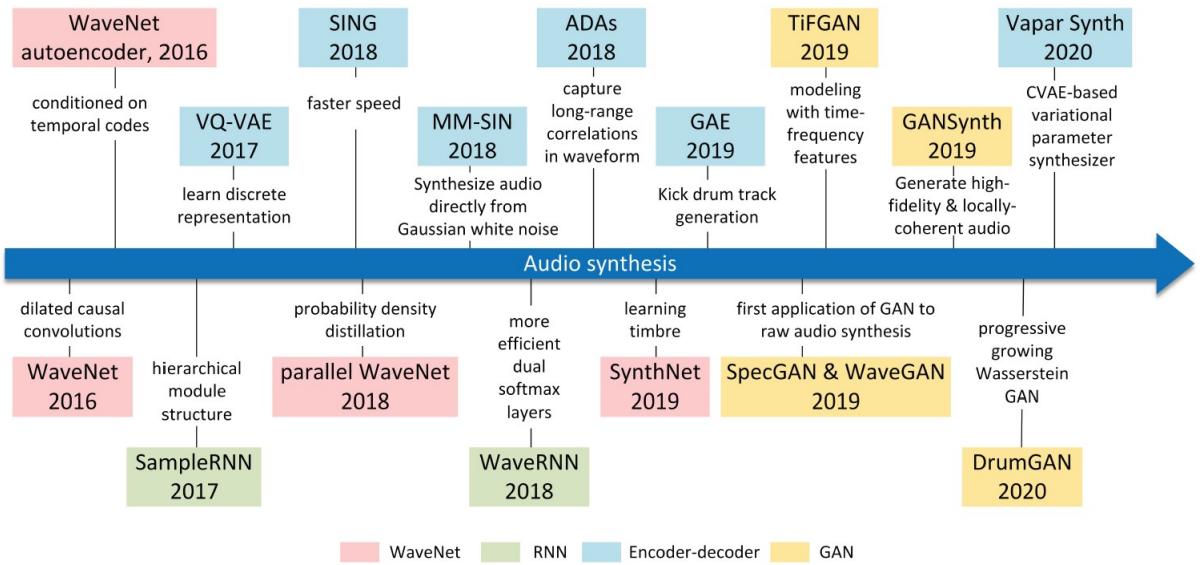


FIGURE B.7 – Ligne du temps : génération de synthèse audio.

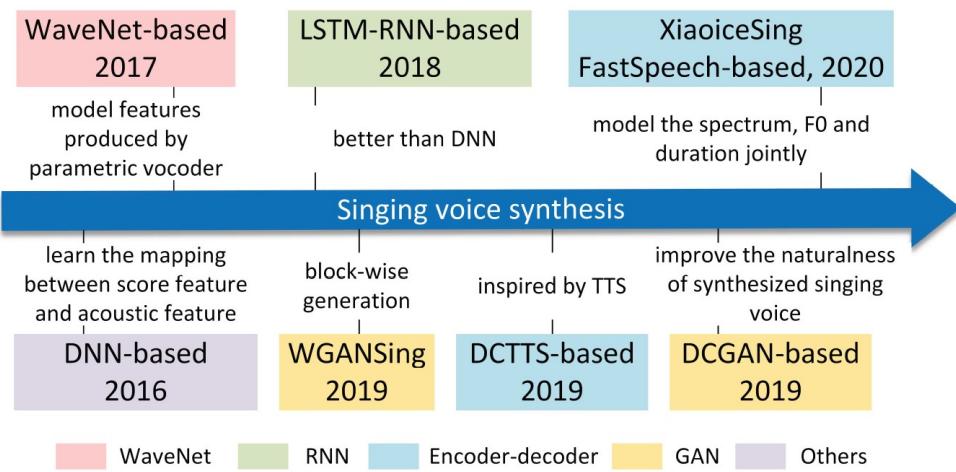


FIGURE B.8 – Ligne du temps : génération de synthèse de voix humaine.

Annexe C

Qu'est-ce que la musicalité ?

Voici quelques-unes des réponses fournies par les participants sur le sujet de la musicalité. Il ne s'agit donc que d'un échantillon, mais ces quelques réponses retenues sont assez représentatives de la variété des réponses fournies (réponses laissées telles qu'écrites par les participants) :

- « Ensemble de notes que l'on trouve agréable, entraînante, selon les personnes. »
- « Un son qui sonne bien. »
- « C'est la capacité d'exprimer une émotion à travers la musique en utilisant des harmoniques et des structures temporelles. »
- « Une séquence de sons qui suscite des sentiments agréables et émotions. »
- « Avoir des notes et/ou une suite de note qui donne une sonorité harmonieuse et qui est agréable à écouter. »
- « Un son, une musique, un rythme, etc. qui arrive a attiré mon attention et qui m'emporte dans son univers. »
- « La musicalité est une succession de note de musique dont l'ensemble forme un son, une musique. »
- « Application de la théorie de la musique dans un morceau . »
- « La bonne est celle qui suscite l'émotion. »
- « Ensemble de notes qui rend un morceau harmonieux, plaisant à l'oreille, mélodieux. La richesse en termes de variété de notes. »
- « Quelque chose de beau à entendre. »
- « Je me dis que c'est la complexité des notes, la douceur de la mélodie, l'immersion intellectuelle que les notes nous procurent... »
- « Ensemble de sons pour produire une mélodie. »
- « Harmonie. »
- « Qqch qui sonne bien, agréable à l'oreille et cohérent. »
- « Le fait que complexité, nuance et créativité d'un morceau sonne bien et provoque des émotions. »
- « Une bonne musicalité donne envie d' écouter et de réécouter une musique. »
- « La qualité de la musique d'un point de vue objectif. »
- « Une succession de notes d'un ou de plusieurs instruments qui sonne juste (contient de la subjectivité). »
- « Des accords qui, une fois assemblés ensemble, créent une harmonie à l'oreille humaine, mais ceci est hautement subjectif. »

- « La qualité de la séquence musicale. »
- « Du son qui ressemble à une musique, une suite harmonieuse de bruit qui sonne bien à l'oreille. »
- « L'harmonie des sons, un taux de syncope adapté à l'émotion que l'on veut transmettre à l'auditeur avec une mélodie en toile de fond, dans un film typiquement c'est ce qui raconte l'histoire et non pas ce qui l'illustre : une BO, pas un bruit de fond. Il faut aussi qu'elle renferme un peu de surprise dans sa mélodie. »
- « Il s'agit de la qualité rythmique et des accords de la mélodie. »
- « La musicalité est selon moi une propriété subjective donnée à un son, qualifiant la manière dont les émotions pouvant être ressenties lors de son écoute procurent du plaisir à l'auditeur (satisfaction, bien être, joie, énergie, mais aussi tristesse voire mélancolie). La musicalité est selon moi dépendante de la mélodie présente dans ce son et de la manière dont cette mélodie a été produite (instrumentalité, style, enregistrement, diffusion). »

Annexe D

Formulaire du test de perception humaine

Dans les deux pages suivantes est illustré le formulaire distribué aux participants. Les données des résultats qualitatifs sont donc issues de ce sondage.

Perception Humaine & Générateurs Musicaux

Bonjour et merci à vous de vous êtes déclaré participant !

Ce test a pour objectif de récupérer une évaluation de l'appréciation humaine de différentes séquences audio. Vous retrouverez ainsi dans ce test différents types de production comme des enregistrements de concours de piano rejoués sur synthétiseur, mais également certaines petites créations d'Intelligences Artificielles.

Votre mission, si vous l'acceptez, est de donner deux notes sur 10 à chacune de ces séquences :

1. A quel point avez-vous apprécié la séquence ?
2. A quel point avez-vous trouvé la séquence intéressante (en termes de musicalité) ?

Enfin, de manière à pimenter tout ça... pensez-vous être capable de reconnaître le vrai du faux ???

3. Le compositeur de la séquence est-il un Humain ou une IA ???

A vous de jouer !!!

Mais avant, apprenons un peu à nous connaître :

Votre **nom & âge** :

Etes-vous **musicien** ? :

Si oui, quelle est votre **expérience** :

Pour vous, qu'est-ce que la **musicalité** ? :

FIGURE D.1 – Formulaire du test de perception, page 1.

Vous trouverez tous les fichiers dans ce drive. Ne soyez pas surpris, de manière à ne pas biaiser les résultats, les données ont été réenregistrées et jouées sur différents synthétiseurs. Ainsi, pas de favoritisme ! Ne vous fiez donc pas à la qualité audio, mais bien au contenu.

Vous n'avez qu'à compléter ce tableau :

NOM	APPRECIATION	INTERET / MUSICALITE	HUMAIN OU IA
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
Serie_1			
Serie_2			

Vous pensez avoir été un bon enquêteur ? Quelles ont été vos **critères de détection** d'IA ? :

Pour terminer, passons à une petite vidéo ! En effet, un des générateurs intelligents que vous avez noté est voué à pouvoir être utilisé avec des appareils musicaux tel que celui présent dans la vidéo. Trouvez-vous cette performance combinant générateur intelligent et utilisation humaine intéressante ?

Appréciation	Intérêt / musicalité

ENCORE MERCI A VOUS POUR VOTRE TEMPS !!!

FIGURE D.2 – Formulaire du test de perception, page 2.