

Percussion aérienne: amélioration de la détection des coups dans le jeu d'un percussionniste

DEA d'Acoustique, Traitement du Signal,
et Informatique Appliqués à la Musique
Université de la Méditerranée, Aix-Marseille II

Vincent Goudard
vincent.goudard [at] ircam.fr

Sous la direction de:
Myriam Desainte-Catherine
Christophe Havel
Sylvain Marchand

9 septembre 2004



Résumé

Ce mémoire présente le travail réalisé durant mon stage de DEA ATIAM¹ au SCRIME² sur le projet de percussion aérienne (précédemment nommé ‘percussion virtuelle’).

Ce stage de recherche a comme objectif une analyse des gestes de percussionniste, en vue d’une meilleure détection des coups, et d’une reconnaissance des gestes et enchaînements de la percussion classique. Le jeu d’un percussionniste a été étudié, dans les conditions d’utilisation de cet instrument virtuel. Enfin, un modèle de prédiction linéaire a été développé servant à la fois les besoins de correction des valeurs issues des capteurs, et d’anticipation de la détection des impacts.

MOTS CLEFS : Interaction Homme-Machine, Instrument de musique digital, Percussion, Analyse gestuelle, Prédiction Linéaire

This thesis describes the work achieved during my master ATIAM internship at SCRIME, on the air percussion (formerly known as ‘virtual percussion’).

This research internship aimed at analysing percussionist’ gestures, for a better impacts’ detection, and classical percussion strikes recognition. The gestures of a percussionist were investigated, while playing this virtual instrument. Also, a linear prediction model was used to both correct the incoming sensors data in real-time, and anticipate the impacts detection.

KEY WORDS : Human Computer Interaction, Digital Musical Instrument, Percussion, Gesture Analysis, Linear Prediction

¹Acoustique, Traitement du Signal, et Informatique Appliqués à la Musique

²Studio de Création et de Recherche en Informatique et Musique Electroacoustique

Table des matières

1	Structure du rapport	5
2	Remerciements	6
3	Introduction	7
3.1	Le SCRIME	7
3.2	Le projet ‘Métamorphose’	7
4	Recherches déjà menées	9
4.1	Sur le contrôle gestuel	9
4.2	Sur les caractéristiques des gestes	9
4.2.1	Les types de gestes par David Efron	9
4.2.2	Les gestes instrumentaux	10
4.2.3	Le cas des gestes de percussion	10
4.3	La correspondance entre geste et contrôle	11
4.4	Antécédents sur la percussion aérienne au SCRIME	13
4.4.1	Technologie	13
4.4.2	Instruments	14
4.4.3	Détection des impacts	14
4.4.4	Reconnaissance des coups	15
4.4.5	Reconnaissance des enchaînements	16
5	Modélisation	18
5.1	Modèle général	18
5.2	Mapping entre capteurs et paramètres gestuels	20
5.3	Mapping entre paramètres gestuels et sonores	21
6	Acquisition de données	22
6.1	Matériel utilisé	22
6.2	Méthode	22
6.3	Première acquisition : réaction de l’instrumentiste face à un DMI	23
6.3.1	L’importance du jeu en contexte	23
6.3.2	Influence du retour auditif	24
6.4	Deuxième acquisition : plus proche du jeu naturel ?	24
6.4.1	Influence du retour auditif	25
6.4.2	Influence des gestes d’accompagnement	25
6.5	Troisième acquisition : les mêmes séquences avec le son	26

7	Analyse des signaux	27
7.1	Visualisation	27
7.1.1	Visualisation des données brutes et des paramètres	27
7.1.2	Visualisation des méta-données	28
7.1.3	Visualisation dynamique	28
7.2	Valeurs erronées	29
7.2.1	Distorsion de l'espace	29
7.2.2	Valeurs aberrantes	30
7.3	Algorithme de correction	34
7.3.1	LPC et méthode de Burg	35
7.3.2	Description de l'algorithme de correction	36
7.4	Détection des impacts	38
7.5	Analyse des types de coups	40
7.5.1	Extraction de statistiques de vitesse et accélération	40
7.5.2	Extraction des types de coups par corrélation	40
8	Résultats	41
8.1	Correction des données	41
8.2	Détection des impacts	42
8.3	Analyse de types de coups	44
8.3.1	Analyse des paramètres	44
8.3.2	Extraction des types de coup	44
9	Bilan et perspectives	47
9.1	Bilan	47
9.2	Perspectives	47
A	Méthode de Burg	49
B	Extraits du code Matlab	50
B.1	Extraction des paramètres de la LPC	50
B.2	Correction des données	51
B.3	Détection des impacts	56
B.4	Etude statistique	57
B.5	Reconnaissance des coups	59
	Bibliographie	61

Table des figures

3.1	La partition de Métamorphose 7 (©C.Havel)	8
4.1	La chaîne de contrôle d'après Joel Ryan	12
4.2	Les flocks of birds montés sur les baguettes de batterie	13
4.3	Un éditeur d'instrument permet de placer des volumes dans l'espace	15
4.4	Les différents types de coup	16
4.5	Automate de reconnaissance des enchaînements	17
5.1	Modèle général pour la partie analyse gestuelle percussion aérienne	19
6.1	L'influence des conditions de jeu : Enregistrement de coups types et Performance réelle . . .	23
6.2	La partition jouée à 3 nuances et tempi différents	24
6.3	Les gestes d'accompagnements : inspiration sur la pulsation avant un frisé	25
7.1	Le repère défini par rapport à la base émettrice	27
7.2	Animation en 3 dimensions : baguette gauche et droite	29
7.3	Étude de la distorsion statique de l'espace	30
7.4	Retournement des axes par franchissement du plan (oYZ)	31
7.5	Discontinuité des valeurs d'angle	31
7.6	Les données bloquées dues à une saturation des capteurs	32
7.7	Influence de perturbations électromagnétiques sur les données	33
7.8	Problèmes de contact	34
7.9	Prédiction d'un <i>frisé</i> par modèle LPC coefficients calculés par la méthode de Burg	35
7.10	Réponse en fréquence du filtre pour la prédiction du frisé	35
7.11	Erreur quadratique de prédiction des 100ms en fonction de l'ordre et la taille du modèle LPC	37
7.12	Valeurs maximales de l'accélération et de sa dérivée	39
8.1	Position non corrigée, et corrigée.	41
8.2	Correction des sauts d'angle.	42
8.3	Détection de l'impact sans anticipation	43
8.4	Détection de l'impact avec anticipation	43
8.5	Statistiques pour l'accélération tangentielle autour de l'impact	44
8.6	Statistiques pour la norme de la vitesse autour de l'impact	45
8.7	Similarité pour un enchaînement de coups	46

Chapitre 1

Structure du rapport

Ce rapport est composé de 7 parties.

La 1ère partie présente le SCRIME ainsi que le projet Métamorphose dans lequel s'inscrit le développement de la percussion aérienne présentée dans ce rapport.

La 2ème propose une revue des différentes études déjà menées sur le geste en général, et le geste instrumental en particulier, afin de situer cette étude dans un contexte plus général. Sont présentés également dans cette partie les développements réalisés par les groupes d'étudiants ayant travaillé sur la percussion aérienne du SCRIME.

Un modèle général est proposé pour la percussion virtuelle dans la 3ème partie.

La 4ème partie explique le déroulement et la méthodologie des séances d'acquisition de données du geste de percussionniste.

La 5ème partie présente les programmes réalisés pour l'étude et l'analyse des données des capteurs.

La 6ème partie exhibe les résultats de ces programmes : correction de données erronées, détection d'impacts, et reconnaissance de coups.

La 7ème partie dresse un bilan du travail réalisé, et indique les développements possibles pour la suite de ce projet.

Enfin, des détails mathématiques, ainsi que des extraits du code Matlab sont mis en annexes.

Chapitre 2

Remerciements

Je tiens à remercier tout d'abord les personnes qui encadraient ce stage : Myriam Desainte-Catherine qui m'a accueilli au SCRIME pour ce stage sur un sujet qui me tenait à cœur ; Christophe Havel qui est à l'origine de ce projet pour les discussions sur la composition (et pour ses cours sur Max/MSP !) et Sylvain Marchand pour ses conseils. Un grand merci au percussionniste Clément Fauconnet, pour sa bonne humeur et sa patience face aux caprices de la technologie. Je remercie également l'équipe pédagogique du DEA, et particulièrement Cyrille Defaye, et Gérard Assayag pour leur disponibilité tout au long de cette année enrichissante.

Je tiens aussi à remercier les personnes que j'ai côtoyées durant ce stage : Julien Villain pour son aide au milieu de l'hécatombe des machines. . . , Nicolas Louis qui a généreusement digitalisé la vidéo des mesures, François Dumeaux –aka Drücpa Dracous–, pour la musique et le thé tout l'été. . .

Je tiens aussi à remercier particulièrement Györgi Kurtág, pour ses conseils et les discussions sur le geste, la musique, le rapport avec les machines, qui ont été très enrichissantes.

Merci à Christian Eloy de m'avoir permis de suivre les cours d'électroacoustique du Conservatoire National de la Région de Bordeaux, les compositeurs Jean-Michel Rivet, Laurent Soulier, David Trescos, ainsi qu'aux étudiants de la classe d'électroacoustique, pour des discussions, des écoutes, et des échanges qui m'ont fait connaître davantage cette musique.

Je remercie également à Jean-Patrick Allant pour les cours de percussions qu'il m'a gracieusement donné, et dont les réflexions ont sûrement guidé mon approche de l'analyse du geste de percussion.

Chapitre 3

Introduction

3.1 Le SCRIME

Le SCRIME¹ est une cellule d'activité rassemblant artistes et scientifiques. Son objectif est de permettre aux premiers de bénéficier d'un transfert de connaissances scientifiques et aux seconds d'une expertise musicale. Les membres du SCRIME sont des chercheurs en informatique musicale du LaBRI, et des compositeurs issus pour la plupart de la section électroacoustique du Conservatoire National de Région de Bordeaux.

Rattaché administrativement au Laboratoire Bordelais de Recherche en Informatique (LaBRI), le SCRIME résulte d'une convention de coopération entre le Conservatoire National de Région de Bordeaux, l'ENSEIRB (Ecole Nationale Supérieure d'Electronique, Informatique et Radio-communications de Bordeaux), et l'Université Bordeaux I.

La recherche fondamentale menée au SCRIME concerne principalement la modélisation informatique du son (analyse, transformation et synthèse), et l'aide à la composition (calcul symbolique); tandis que la recherche appliquée concerne davantage les interfaces musicales Homme-Machine (logiciels pédagogiques, et instruments virtuels).

3.2 Le projet 'Métamorphose'

Christophe Havel, compositeur membre du SCRIME, est à l'origine du projet 'Métamorphose', initié en 2000 : il souhaite créer et réaliser un spectacle musical associant des instrumentistes traditionnels (interprètes et improvisateurs) à un méta-instrument électroacoustique.

L'originalité de ce projet réside dans l'utilisation du formidable potentiel gestuel et musical d'instrumentistes professionnels, en l'appliquant à un instrument autre que leur instrument traditionnel. Chaque musicien joue ainsi virtuellement, avec ses propres réflexes mécaniques et auditifs, et des capteurs adaptés à son instrument. L'ensemble des informations recueillies est alors envoyé vers un ordinateur qui génère des sons en fonction d'une partition écrite à l'avance. Mais à la différence d'une écriture traditionnelle, la partition consiste en divers environnements de réactions aux sollicitations instrumentales. L'oeuvre est alors définie par la forme générale proposée aux instrumentistes, les environnements et les modes de jeu associés, et la facture des sons électroniques. La formation alors envisagée était : guitare, saxophone,

¹Studio de Création et de Recherche en Informatique et Musique Electroacoustique

percussion et thérémin.

Pour les deux premiers instruments, le problème de la captation du geste fut confiée aux instruments MIDI déjà existants², quant au thérémin, il fournit un signal audio directement utilisable. Ces informations de jeu instrumental (MIDI ou audio) sont alors collectées sur l'ordinateur pilotant les sons générés en fonction de la partition, dans le logiciel Max/MSP³. 'Métamorphose 1' pour guitare MIDI, saxophone MIDI et thérémin, a ainsi été présentée au Molière – Scène d'Aquitaine en Mai 2000.

En revanche, le cas du percussionniste est beaucoup plus délicat, car il s'agit de capter et de reconnaître une gestique se déroulant en trois dimensions, sans qu'il n'y ait de frappe sur des points précis. Les *pads* MIDI couramment employés par les batteurs voulant disposer de sons de synthèse, réduisent considérablement la finesse du jeu, la ramenant à un simple déclenchement, avec contrôle du volume. De plus, il génèrent des bruits percussifs parasites dont on aimerait se passer.

Il était donc nécessaire de trouver une autre alternative, permettant de s'affranchir de ces désagréments, et de rendre mieux compte de la richesse gestuelle du percussionniste. Le projet de la percussion aérienne fût lancé la même année par la section scientifique du SCRIME. Plusieurs groupes d'étudiants ont commencé l'étude d'un dispositif de percussion aérienne depuis 2001, et en novembre 2003 eu lieu la première performance avec le dispositif de percussion virtuelle pour la création de 'Métamorphose 7', avec saxophone MIDI et percussion aérienne.

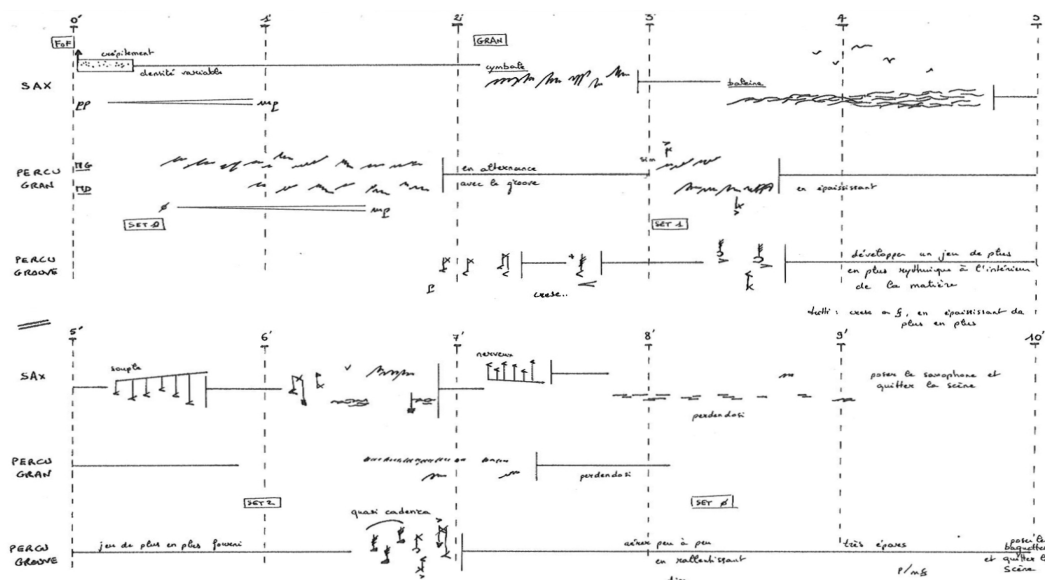


FIG. 3.1 – La partition de Métamorphose 7 (©C.Havel)

²Saxophone MIDI de Yamaha, et guitare MIDI

³Logiciel modulaire de traitement du signal en temps réel. Voir[PZ03]

Chapitre 4

Recherches déjà menées

Les bras ont des mouvements soutenus, liés, s'apparentant en ce sens à la ligne mélodique. De temps à autre, en une forme spatiale définie ou mouvement, ils marquent la pulsation. Ils se déploient en "glissant" le long des plans de l'espace, mais parfois aussi, ils se déroulent en un mouvement centrifuge, dans une des directions spatiales, ou reviennent dans un mouvement centripète.[TrpOA93]

4.1 Sur le contrôle gestuel

L'analyse du geste instrumental est un domaine assez récent qui a commencé à émerger à la fin des années 1970, avec l'étude des interfaces pour l'IHM¹ appliquée à la synthèse sonore [Las77] [Flo78]. L'étude du geste instrumental arrive peu de temps après[Abb82].

L'application à la percussion n'apparaît qu'au début des années 90, notamment grâce au développement du 'Radio-Baton' par Robert Boie[RBS89] aux Laboratoire de la 'Bell Telephone' en 1989.

Durant les deux dernières années, le domaine des interfaces musicales a pris suffisamment d'importance pour que lui consacré une conférence annuelle spécifique. Débutant par un workshop au CHI² 2001, le NIME³ est devenue une conférence internationale indépendante. Un poster sur la percussion aérienne étudiée ici a été présenté [DCH04] à la dernière édition du NIME.

4.2 Sur les caractéristiques des gestes

4.2.1 Les types de gestes par David Efron

Dans son étude[Efr72] des différents gestes de conversation entre groupes ethniques voisins de New York, David Efron distingue trois fonctions principales du geste : *spatio-temporelle*, *interlocutionnelle*, et *linguistique*. Les gestes spatio-temporels représentent le mouvement pur, libre de tout contexte de conversation, dont les gestes de percussionniste semblent faire partie. Ces gestes peuvent selon lui être catégorisés selon 5 aspects :

- rayon (ampleur du mouvement)
- forme

¹'Interaction Homme-Machine'

²Computer Human Interaction

³'New Interfaces for Musical Expression'

- plan (direction et orientation)
- partie du corps effectuant le mouvement
- tempo (ou fréquence)

La classification de David Efron, bien que très générale, fournit un éclairage certain sur les caractéristiques du geste utiles pour leur analyse.

4.2.2 Les gestes instrumentaux

Depuis l'avènement dans les années 1960 de la synthèse sonore et des synthétiseurs grand publics, le développement des interfaces de contrôle n'a cessé de se développer, avec en parallèle, une étude des gestes instrumentaux permettant [d'essayer] de comprendre ce qui fait la richesse du jeu des instrumentistes, sur les instruments traditionnels.

Une bibliographie assez exhaustive sur ce sujet a été établie par Marcello Wanderley et Marc Battier[WB00], qu'il serait superflu de reproduire ici.

Cependant, et comme le sujet proposé ici concerne la reconnaissance de gestes de percussionnistes, et afin de définir un vocabulaire qui sera ré-utilisé plus loin dans ce rapport, je soulignerai deux classifications intéressantes des gestes instrumentaux :

Classification de François Delalande

François Delalande propose [Del88] la classification suivante :

- **Gestes effectifs** : ceux qui produisent effectivement le son.
- **Gestes d'accompagnement** : des autres parties du corps notamment de la tête, des jambes...
- **Gestes figuratifs** : ceux sont les gestes perçus par le public à travers le son produit, mais qui n'ont pas de correspondance directe avec un mouvement.

Classification fonctionnelle de Claude Cadoz

Claude Cadoz propose [Cad88] une classification plus fonctionnelle :

- **Les gestes d'excitation** fournissent l'énergie nécessaire à l'apparition du phénomène sonore. Il peuvent être 'instantanés'⁴, ou bien 'continus'.
- **Les gestes de modification-modulation** changent les propriétés de l'instrument, et affectent la relation entre le geste d'excitation et le son produit. Ils peuvent être paramétriques (e.g. un vibrato) ou bien structurel (e.g. la sourdine d'une trompette).
- **Les gestes de sélection** consistent en un choix entre divers éléments similaires dans un instrument. Ni ils ne fournissent d'énergie à l'instrument, ni ne modifient les propriétés de l'instrument.

4.2.3 Le cas des gestes de percussion

Antécédents

Relativement peu d'étude du geste de percussionniste ont été réalisées, les systèmes utilisées ne permettant souvent pas une étude adéquate : l'acquisition par caméra⁵ est souvent limitée par

⁴Soit *percussifs* comme dans notre cas, mais aussi *pincés* comme pour la harpe par exemple. Wanderley souligne la différence de geste entre ces deux types d'excitation [Wan01] p34

⁵E.g. avec VNS ou BigEye du STEIM.

une faible fréquence d'échantillonnage⁶, ou bien entraîne un calcul très lourd pour l'extraction de la position des baguettes.

Les accéléromètres, comme ceux utilisés dans la '*percussion augmentée*' développée à La Kitchen[CH03], qui souffrent moins des perturbations que les capteurs électro-magnétiques et dont les signaux sont échantillonnables à une fréquence élevée, peuvent se révéler très intéressants pour une étude plus fine de la dynamique du geste. Malheureusement, ils ne renvoient pas la position absolue dans l'espace nécessaire pour disposer d'un set de percussion virtuelle avec des positions définies dans l'espace réel, comme c'est l'intention dans ce projet.

Le geste de percussionniste dans les classification proposées

Au regard des classifications de gestes précédemment citées, il convient de remarquer plusieurs choses concernant la percussion, dont les gestes complexes se situent à la frontière entre ces catégories :

- Les gestes que l'on cherche à analyser et reconnaître sont essentiellement des *gestes effectifs* et d'*excitation*.
- Les *gestes de modulation* sont intimement liés aux *gestes d'excitation* : la résonance de la percussion va dépendre du temps de contact avec la baguette ; contact qui modifie, dans un temps pouvant se réduire à la milliseconde, la structure de l'instrument.
- Les *gestes de sélection* consistant en des déplacements de la baguette en vue du choix d'un instrument à frapper, sont également très liés aux gestes d'excitation. En effet, alors que l'on peut définir assez précisément le début d'un geste d'excitation, pour la guitare par exemple par le moment où l'on commence à tirer la corde, le geste de percussion commence par un élan dont le début est le prolongement continu d'un geste '*de sélection*'.

Gestes d'accompagnement dans la percussion

Les gestes d'accompagnement qu'ils soient de la tête, du pied, ou bien des mains, sont souvent une façon pour le musicien de *marquer une pulsation*, qu'elle soit de l'ordre du tempo, de la mesure, ou de la phrase musicale. Cette fonction musicale est évidemment essentielle dans le cas de la percussion : certes la percussion ne se réduit sûrement pas au marquage de la pulsation, mais plus que tout autre instrument, elle y contribue.

4.3 La correspondance entre geste et contrôle

L'ensemble du projet de la percussion aérienne implique des choix de design qui permettent difficilement de dissocier la partie analyse de la partie synthèse : les choix pour l'une dépendant fortement de l'autre. Il existe un vaste domaine, encore peu formalisé⁷, sur les fonctions de correspondance, ou '*mapping*' en anglais, entre les données instrumentales issues des capteurs et le contrôle de la synthèse sonore.

⁶'Dans les conditions optimales, on peut atteindre 8 frappes par seconde.'[God97]. NB : Avec une fréquence d'échantillonnage de 50Hz.

⁷M. Wanderley a procédé à une revue des stratégies de mapping, soulignant que '*Peu de travaux ont été proposés sur le mapping pour la musique interactive*' [Wan01]. Remarquons toutefois que ce domaine connaît aujourd'hui une forte investigation notamment dans les conférences du NIME

En effet, le contrôle de la synthèse sonore dépend encore beaucoup de la particularité de ses divers algorithmes, et il n'existe pas encore d'interface générale pour piloter l'ensemble des algorithmes de synthèse sonore. De la même façon, les informations issues des capteurs sur les gestes instrumentaux dépendent des types de capteurs, de là où on les place...etc. et il n'y a pas –encore?– de standard permettant une relation clairement définie entre ces deux ensembles de signaux, malgré de nombreuse contribution durant la dernière décennie.

Les premiers travaux de Joel Ryan

Un des premiers à proposer une théorie cohérente sur les correspondances entre interfaces gestuelles et production musicale fût Joel Ryan du STEIM, qui définit la chaîne de contrôle suivante :

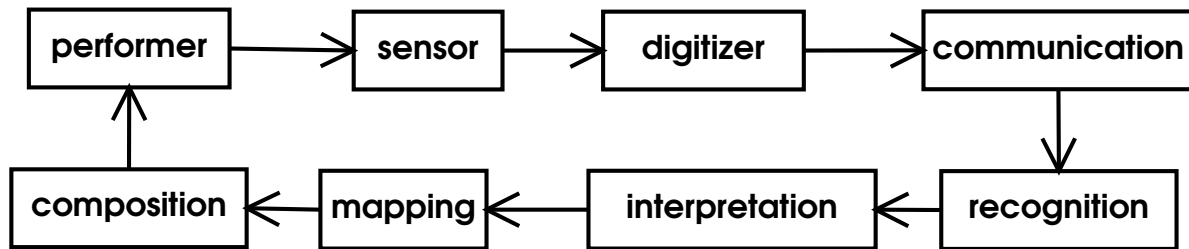


FIG. 4.1 – La chaîne de contrôle d'après Joel Ryan

Les stratégies de mapping proposées se retrouvent autour d'une même classification générale en trois catégories :

- o *Un vers un*
- o *Un-vers-plusieurs* ou *divergent*
- o *Plusieurs-vers-un* ou *convergent*

La catégorie *Plusieurs vers plusieurs* étant une combinaison évidente des catégories pré-citées. Cette formalisation, si elle propose une représentation générale n'évoque cependant pas la nature des variables de mapping, ni celle de leurs relations.

Les apports plus récents

Une division des fonctions de correspondance en deux couches a été proposée par [MWR98], avec une couche de variables intermédiaires. Selon Marcello Wanderley, ces variables intermédiaires sont '*approximativement basées sur des variables perceptives comme la couleur du timbre, le volume sonore et la hauteur, mais peuvent être basées sur des autres caractéristiques perceptives du son, ou n'avoir aucune relation avec la perception, étant alors arbitrairement choisies par le compositeur ou l'instrumentiste*' [Wan01](p.73).

Plus récemment, Loïc Kessous et Daniel Arfib [KA03] ont proposé une division supplémentaire qui se rapproche de celle proposée ici (cf 5.2), en insérant une couche de *paramètres perceptifs liés aux geste*.

4.4 Antécédents sur la percussion aérienne au SCRIME

Plusieurs groupes d'étudiants ont participé à l'élaboration de ce projet, commencé en 2000. Ils ont à la fois contribué au développement de programmes pour l'acquisition de données et leur analyse, et à la modélisation du problème de la détection des coups à partir du cahier des charges fournis par le compositeur.

Malheureusement un problème chronique est l'absence de synthèse des travaux précédents par les groupes d'étudiants ayant successivement travaillé sur la percussion aérienne. Cela pose un problème à la fois au niveau de la modélisation qui est faite, mais aussi un problème de compatibilité des applications développées, chaque groupe étant pratiquement reparti à zéro à chaque fois.

Chacun de ces groupes a cependant apporté une approche différente, et l'ensemble du travail réalisé porte les fruits d'une réflexion sur le sujet avec ses intérêts comme ses oublis.

J'ai donc commencé ce stage par une revue des travaux déjà réalisés. Plutôt que les présenter chronologiquement, j'en rapporte ici les choix par rapports aux principaux thèmes.

4.4.1 Technologie

Capteurs

Au niveau des capteurs, le projet a connu une première phase avec le radio-baton⁸. Mais rapidement, à cause de la faible fréquence d'échantillonnage⁹, et surtout de la faible surface de jeu disponible¹⁰, le radio-baton a été remplacé par les 'flocks of bird' (voir fig. 4.2) permettant une acquisition de la position et de l'angle des capteurs dans une sphère d'environ 1m de rayon, et à une fréquence d'une centaine de Hertz.

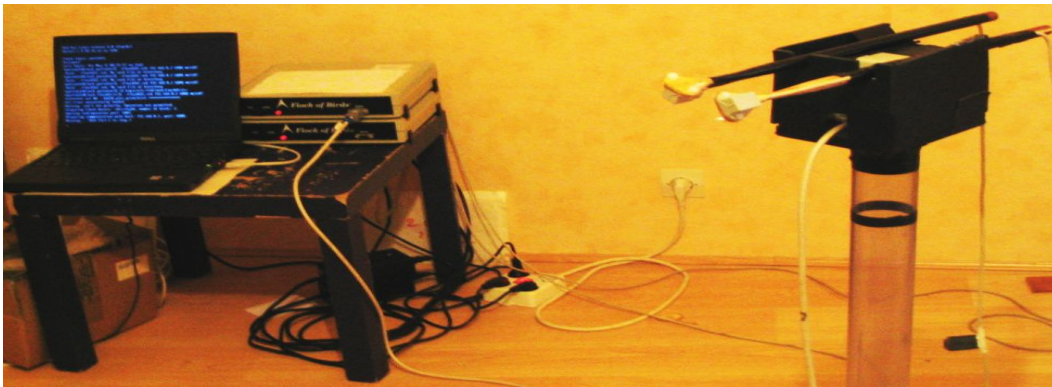


FIG. 4.2 – Les flocks of birds (Ascension Technology) montés sur les baguettes de batterie

⁸[RBS89]

⁹De 50Hz, à l'époque, mais de récents travaux[SD02] ont remis le radio-baton à jour en utilisant une carte son pour l'échantillonnage.

¹⁰Se situant au dessus d'une table de 60cm de côté environ.

Analyse de données et synthèse sonore

La première version présentant une synthèse sonore est implémentées entièrement avec le logiciel jMax¹¹, mais afin de dissocier la synthèse sonore de l'analyse, cette solution est modifiée à partir des travaux suivant : une machine sous Linux effectue l'analyse de données captées, et les données analysées sont envoyées à un MacIntosh équipé du logiciel Max/MSP¹² qui gère la synthèse sonore sans être gêné par la lourdeur de l'analyse gestuelle.

4.4.2 Instruments

Si la première version prototype avec le radio-baton[GM00] ne proposait pas de définition spatiale d'instruments, le passage à un plus grand espace de jeu avec les flocks of birds nécessitait la possibilité de définir des zones de l'espace représentant les percussions.

Ainsi le groupe d'étudiant suivant[MBD01] définit 3 types d'instruments :

- o Instruments 'frappés' (percussifs) : les sons sont générés par des impacts : une **intersection entre une surface active et une baguette**. La vitesse de la baguette au moment de l'impact détermine l'**amplitude** du son produit. La distance entre point d'impact avec un point ou ligne de référence détermine la distorsion du son produit (cette dernière caractéristique a été prévue dans la modélisation mais non implémentée)
- o Instruments 'secoués' (type maracas) Les changements de vitesse et/ou de direction produisent des sons. L'amplitude du changement de vitesse détermine l'amplitude du son produit.
- o Instruments 'toggles' (continus) : les sons sont générés en fonction de la position dans l'espace de l'instrument : l'entrée d'une baguette dans le volume de l'instrument déclenche un unique son continu dont le volume est fonction de la vitesse d'entrée ; la sortie de la baguette stoppe le son.

Les instances de ces instruments sont caractérisées par :

- o Le type de l'instrument.
- o Position, volume et formes (disque, cylindre, sphère, cône, quadrilatère, et parallélépipède.)
- o Surfaces et volumes actifs.

Cette première version ne fonctionnait pas à cause d'un problème avec la bibliothèque d'acquisition des flocks of birds. Le groupe suivant[BBS01] reprendra l'idée générale en ne conservant une reconnaissance des impacts que dans les *volumes* plutôt qu'à travers les *surfaces* des instruments.

Enfin, en 2003 Sébastien Lebreton[Leb03] développe un éditeur d'instruments virtuels, avec une interface graphique 3D (fig.4.3).

4.4.3 Détection des impacts

L'idée proposée à la base de la percussion aérienne est que l'impact n'est pas le simple résultat d'une collision entre la baguette et la surface de l'instrument, mais plutôt un geste particulier, pouvant être réalisé dans le vide (comme c'est le cas de toute façon dans la percussion aérienne), et qui assure un transfert d'énergie entre la baguette et la percussion. Les différentes études menées aboutissent généralement à une segmentation du geste en 3 phases :

¹¹[DSOB03]

¹²[PZ03]

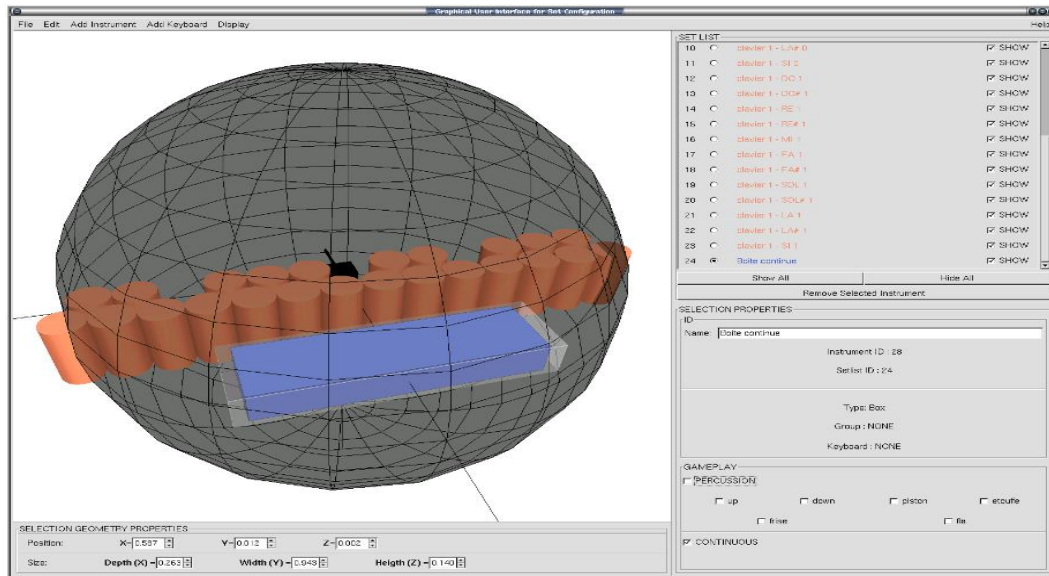


FIG. 4.3 – Un éditeur d'instrument permet de placer des volumes dans l'espace

- **Le début du coup**, c'est à dire le début du mouvement d'élan menant à l'impact, caractérisé par un dépassement de seuil de la vitesse. Nous verrons plus loin qu'un franchissement de seuil de vitesse ne semble pas adapté à la détection d'un début de coup en tant que *détection d'intention*. Elle semble en effet davantage caractérisée par une accélération significative qu'une vitesse.
- **L'impact**, caractérisé par le changement de direction de la baguette. Pour palier le retard qu'entraîne l'attente d'un changement de direction, ce critère avait été remplacé par un franchissement négatif de seuil de vitesse, fixé empiriquement à la moitié de la vitesse maximale atteinte durant la phase d'élan, et rendant l'instant précis de l'impact assez aléatoire temporellement. Nous verrons plus loin comment ce problème a été résolu grâce à la prédiction.
- **La fin du coup**, caractérisé par un retour à une position 'de repos' basse (down), haute (up, piston), pouvant éventuellement correspondre au début du coup suivant, ou bien par un blocage à la position de l'impact (étouffé).

L'analyse des coups ne prenait alors que la *direction verticale* en compte, avec des coups supposés être frappés *vers le bas*. Elle a récemment été étendue aux 3 dimensions en 2003, en conservant le même algorithme et en remplaçant la vitesse selon l'axe z, par la norme du vecteur vitesse dans l'espace tri-dimensionnel.

4.4.4 Reconnaissance des coups

Dans les objectifs initiaux de la percussion aérienne, il avait été envisagé, d'une part de reconnaître les *impacts* et les *types de coups*. Les types de coups sont ceux qui ont été définis par le percussionniste François Dupin pour l'enseignement de la percussion classique.

Il existe 4 coups de base :

- o *down* : le mouvement commence assez loin de l'instrument, le frappe et revient légèrement

- o *up* : le mouvement commence beaucoup plus près de l'instrument, le frappe et remonte assez haut. Ce coup apparaît essentiellement dans un '*fla*'
- o *piston* : le mouvement est celui d'un va-et-vient.
- o *étouffé* : le mouvement est arrêté brusquement après l'impact, en cherchant à bloquer la résonance par contact entre la baguette et la surface de frappe.

Il faut remarquer que dans l'esprit du percussionniste, il ne s'agit pas de taper la percussion, mais vraiment (sauf pour le cas du coup étouffé) de transmettre une énergie, à la manière du karatéka. Cette façon de penser le geste en anticipant le retour de la baguette¹³ a deux objectifs :

- o Être prêt au plus tôt et avoir le maximum d'élan pour la frappe suivante.
- o Avoir un temps de contact avec l'instrument le plus bref possible pour permettre une résonance maximale de son corps.

Les différents types de coups proposés par François Dupin ont été modélisés lors des 3 premiers stages, d'après leur 'allure théorique', par des seuils sur la position [GM00], et la vitesse [BBS01], à divers instants entre le 'début du coup' et la 'fin du coup'.

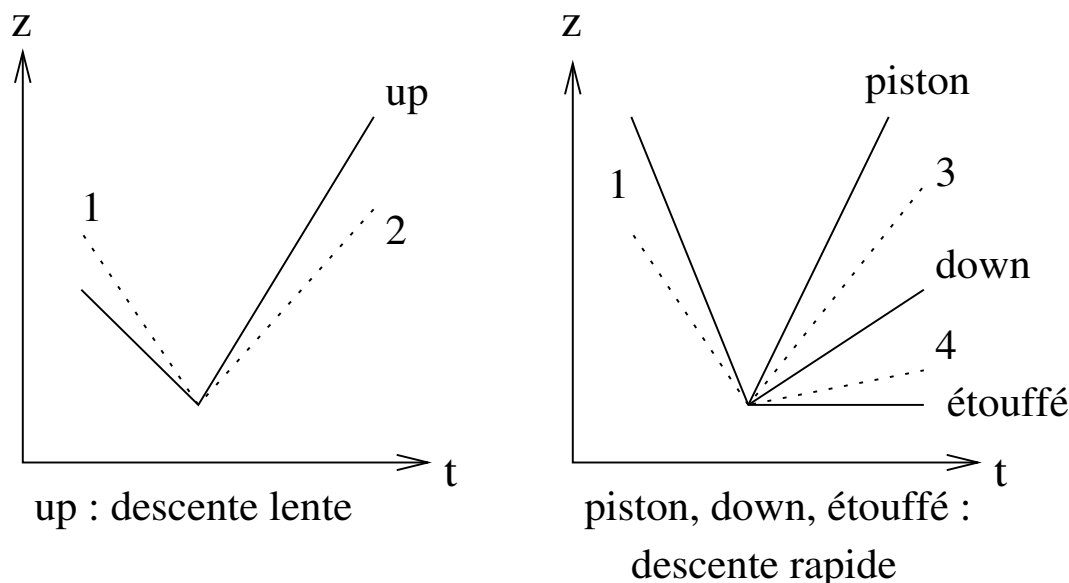


FIG. 4.4 – Les différents types de coup

Une étude statistique [SP02] est envisagée ensuite afin de déterminer au plus tôt le type de coup et diminuer la latence. Malheureusement, les résultats de cette étude, sous la forme de grands tableaux de chiffres, ne sont pas interprétés pour trouver quels seraient les critères d'analyse intéressants ; et l'algorithme de reconnaissance est resté le même.

4.4.5 Reconnaissance des enchaînements

Les enchaînements que l'on cherche à reconnaître sont :

- o Le *fla* qui consiste en une succession rapide d'un *Up* et d'un *Down*.

¹³Cette technique n'est pas propre à la percussion classique : voir les explications données dans [Van88]

- o Le *frisé* ou *roulement* qui consiste en une succession rapide de pistons.

Dès le premier stage sur la percussion aérienne, une reconnaissance des enchaînements a été envisagée comme la reconnaissance d'une succession de coups, et modélisée par des automates (fig.4.5). Cette solution avait alors été abandonnée car trop lourde en calcul.

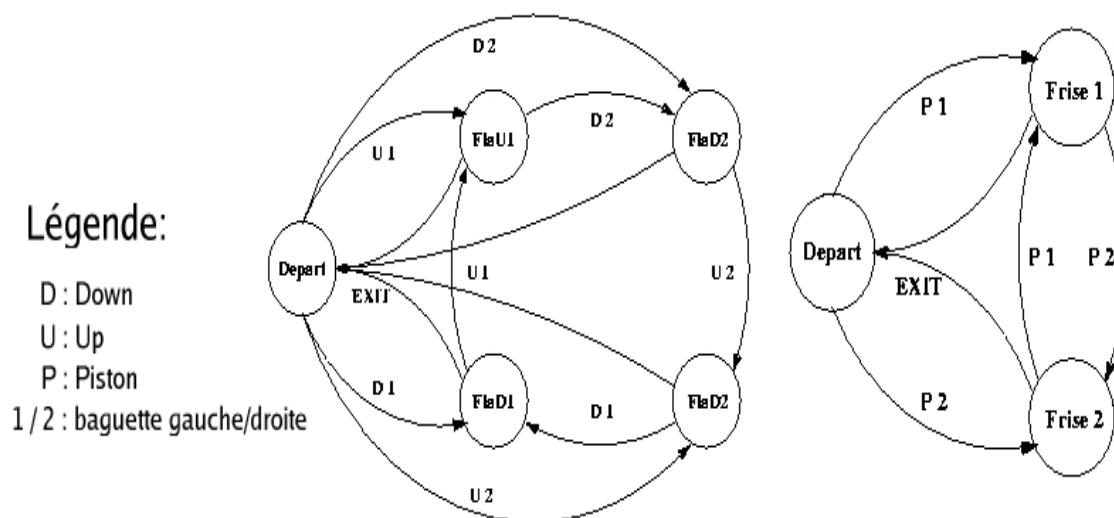


FIG. 4.5 – Automate de reconnaissance des enchaînements d'après [GM00]

Chapitre 5

Modélisation

Both sound and human movement can be represented at various abstraction levels. A mapping will be faster to learn when movement features are mapped to sound features of the same level [Nak00]

5.1 Modèle général

Au regard des travaux cités dans le chapitre précédent, il convient de définir un modèle pour la percussion aérienne sur lequel baser les développements. Ce modèle (voir fig 5.1) se base sur des modèles déjà proposés pour les DMI¹, ainsi que sur les réflexions issues d’une table ronde organisée avec des chercheurs du LaBRI, des compositeurs, et des percussionnistes.

Remarquons que le modèle présenté n’explicite qu’une partie du design du système, celle concernant l’analyse du geste. La partie synthèse fait en effet partie du travail de composition, et n’est pas fixée dans ce projet. D’autre part, un module de visualisation des baguettes et de définition des instruments existe et n’est pas représenté ici.

La partie analyse reçoit comme données d’entrée :

- Les données issues des capteurs : position Px, Py, Pz, et angle Ax, Ay, Az.
- Les données contenue dans le fichier définissant le set de percussion : pour chaque instrument : son nom, son type (percussif ou continu), sa forme (sphère, cylindre, boîte, clavier), sa position et ses dimensions.

Une correction et prédiction par LPC est effectuée en amont de l’extraction des paramètres, et qui est expliquée dans la section 7.3.

Les différents paramètres envoyés en sortie sous forme de message OSC² sont :

- Les impacts (date, No de l’instrument, position relative dans l’instrument, angle et vitesse)
- Le type de coup reconnu.
- Des méta paramètres : tempo, symétrie ...

¹Digital Musical Instrument : (Instrument de Musique Numérique). Est défini dans [Wan01] comme ‘*un instrument comprenant une interface gestuelle séparée d’une unité de synthèse sonore.*’

²Open Sound Control : protocole d’échange de donnée asynchrone développé par le CNMAT [WF97]. NB : tous les messages OSC sont datés

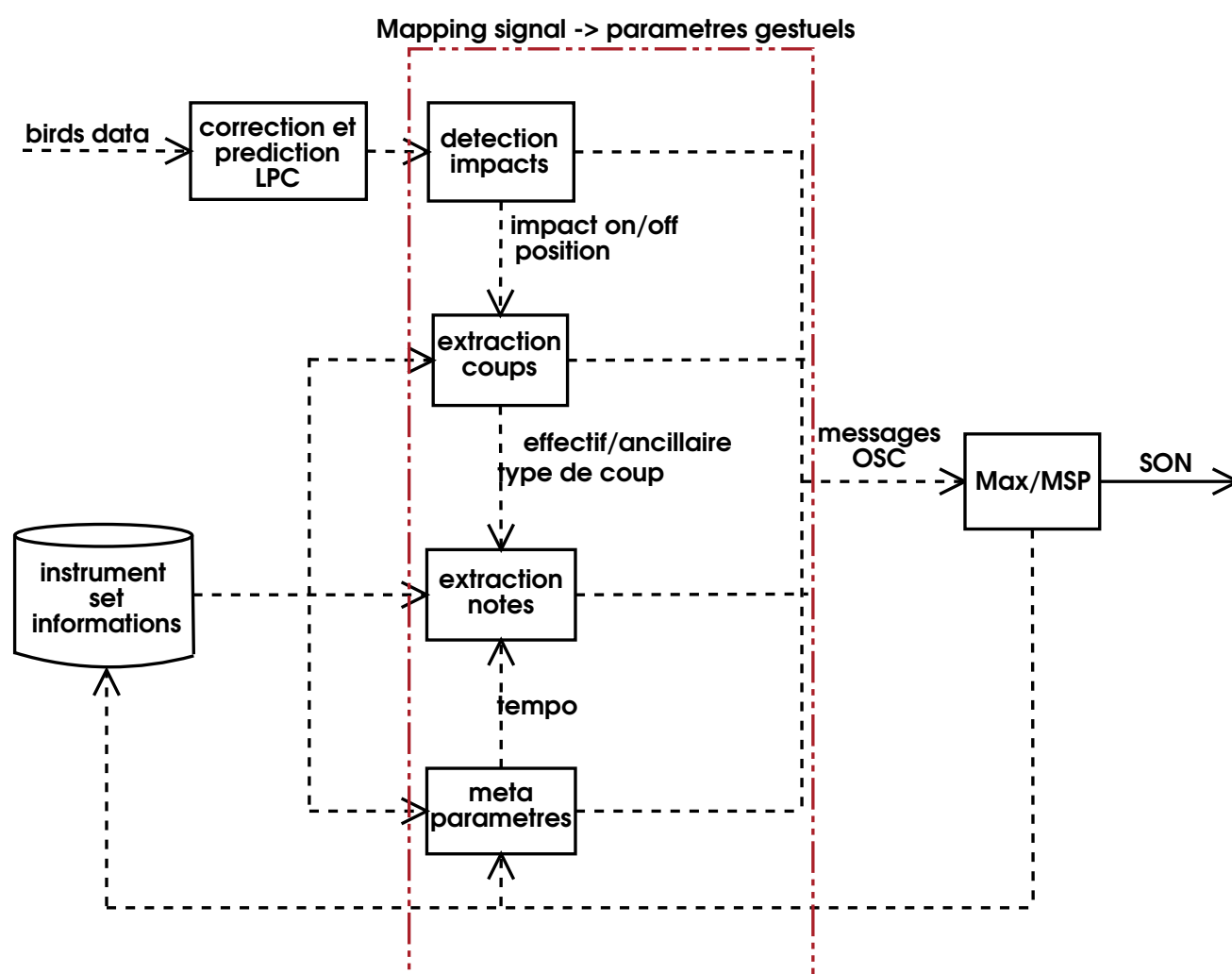


FIG. 5.1 – Modèle général pour la partie analyse gestuelle percussion aérienne

5.2 Mapping entre capteurs et paramètres gestuels

Des paramètres catégoriels

L'analyse des gestes de percussion proposée ici, comprend une couche de variables intermédiaires symboliques abstraites, mais celles-ci ne sont ni des caractéristiques perceptives du geste ou du son, ni des variables 'arbitrairement choisies' par le compositeur. Les *coups* que l'on cherche à reconnaître, sont des éléments d'une *base gestuelle du percussionniste*, issues d'un formalisme proposé par le percussionniste François Dupin. Éléments d'une base, au sens où l'on pourrait représenter la gestique du percussionniste en utilisant une combinaison de ces éléments de base. Cette représentation ne permet certes pas de représenter l'ensemble du mouvement dans toute sa richesse, mais présente deux avantages principaux :

- o Disposer pour le compositeur d'une première représentation quantifiée du continuum gestuel, sous forme symbolique, à laquelle il sera possible d'attacher des attributs perceptifs.
- o Reconnaître une certaine expertise du geste propre au percussionniste professionnel.

Selon Christophe Havel, cette discrétisation du continuum gestuel est une étape nécessaire (quoique qu'insuffisante) pour la composition musicale, qui consiste en la manipulation de *rapports*, d'*intervalles*, entre des valeurs définies sur une *échelle*³.

Un autre intérêt de cette segmentation catégorielle, est d'envisager à terme une analyse de la gestuelle utilisant toute la science issue de la linguistique, déjà appliquée avec succès à la notation musicale classique⁴, et de certains type de jeu de percussion⁵.

Dans cette segmentation, il est envisagé de rendre également compte des gestes ancillaires, dans la mesure où ceux-ci répondrait également de certaines figures typiques, afin qu'ils puissent éventuellement participer de la synthèse sonore. Nous verrons plus loin (Cf 6.4.2), qu'un type de geste ce type a été identifié. Cependant, comme il a été déjà dit, la gestique complexe de la percussion ne se laisse pas forcément réduire à une succession de figures types. Aussi, considérant que l'ambiguïté du geste fait partie intégrante du jeu d'un instrumentiste, on a pensé qu'il serait intéressant que l'interface renvoie à chaque fois les différents coups possibles, avec leur probabilité respective, afin que cette ambiguïté puisse être prise en compte dans le dispositif de synthèse.

Des paramètres continus

Les paramètres gestuels utilisés pour piloter la synthèse sonore ne pourraient se résumer à des catégories finies sans réduire de façon drastique la capacité de modulation propre à tout instrument digne de ce nom. Une étude des paramètres influant la production sonore dans la percussion serait à envisager, mais n'est malheureusement pas possible avec le matériel disponible dans cet étude, trop fragile et à la fréquence d'échantillonnage trop faible pour analyser ces phénomènes transitoires.

On en restera donc à supposer l'influence probable de l'énergie cinétique, du temps et de la surface de contact au moment de l'impact, ainsi que du lieu d'impact. Aussi, ce sont les données de position et de vitesse qui sont renvoyées de manières continues. Les impacts seront codés

³Entretiens personnels avec C.Havel

⁴Voir notamment les travaux de l'équipe Représentation Musicale, de l'IRCAM.

⁵Voir notamment les travaux de B.Bel et J.Kippen[KB] sur les tablas et le système rythmique indien.

avec un début et une fin (à la manière des messages MIDI note on/off), plutôt que sous forme purement événementielle ; ce qui permet de contrôler le temps de ‘contact’ entre la baguette et l’instrument.

On peut également supposer l’influence probable de paramètres plus généraux sur le mouvement des baguettes, comme leur synchronicité, leur écartement, ou encore l’imprévisibilité (par rapport à un modèle LPC par exemple)...

5.3 Mapping entre paramètres gestuels et sonores

Cette tâche est laissée au compositeur et est directement implémentée dans Max/MSP, comme partie intégrante de la composition. Le choix des matières sonores, et des relations entre paramètres constitue un environnement électronique, avec lequel plusieurs musiciens peuvent jouer.

Le mapping utilisé jusqu’alors par C. Havel consiste principalement en un déclenchement de sons à la détection d’un impact dans un instrument. Le volume sonore de ce son est contrôlé par la vitesse maximale avant l’impact, et un filtrage est appliqué en fonction de l’angle de la baguette, afin de modifier le timbre.

La synthèse granulaire, est utilisée comme un instrument *continu*⁶, dont le volume sonore est contrôlé par l’axe Z, les autres degrés (X, Y, et les angles) contrôlant divers paramètres de la synthèse : position dans l’extrait sonore, taille du grain, vitesse de lecture, aléa...proposant ainsi une manière originale de contrôler ce type de synthèse, dans une véritable exploration d’un extrait sonore.

⁶Cf. 4.4.2

Chapitre 6

Acquisition de données

The source of a sound in an interactive computer based instrument is not some abstract or concrete concept of *instrument*, or even the algorithm(s) that have been written, but it is the gesture of the performer, the excitation moment(...)[Pai04]

Il faut remarquer que dans l'esprit du percussionniste, il ne s'agit pas de taper la percussion, mais vraiment (sauf pour le cas du coup étouffé) de transmettre une énergie, à la manière du karatéka. Cette façon de penser le geste en anticipant le retour de la baguette¹ a deux objectifs :

- o Être prêt au plus tôt et avoir le maximum d'élan pour la frappe suivante.
- o Avoir un temps de contact avec l'instrument le plus bref possible pour permettre une résonance maximale de son corps.

6.1 Matériel utilisé

Le dispositif d'acquisition consistait en :

- o Les flocks of birds,renvoyant les données de position et d'angle.
- o Une bibliothèque gérant la communication entre les flocks of birds et le PC.
- o Un programme analysant les données des flocks et renvoyant les informations de position, angle, et impacts au MacIntosh par messages OSC², que je modifiai pour que ces données soient enregistrées dans un fichier séparé.
- o Un Macintosh avec le logiciel Max/MSP pour la synthèse sonore.
- o Une caméra numérique pour la 1ère séance d'enregistrement afin de pouvoir observer les gestes dans leur contexte global.

6.2 Méthode

Des données avaient déjà été acquise par un groupe précédent [SP02], et une analyse statistique réalisé sur les types de coups. Cependant, outre la faible quantité de données, cette analyse présentait deux inconvénients majeurs : elles ne prenaient en compte que l'axe vertical, et surtout, les coups étaient étudiés complètement hors-contexte. La première acquisition de données m'a fait comprendre l'importance de ce point : j'ai rapidement dû revoir la méthode,

¹Cette technique n'est pas propre à la percussion classique : voir les explications données dans [Van88]

²Open Sound Control[WF97]

me rendant compte au fûr et à mesure –des mesures– de différents problèmes liés à l'utilisation de la percussion aérienne, et que je décris ci-après.

6.3 Première acquisition : réaction de l'instrumentiste face à un DMI

Lors de la première séance d'acquisition de données, le jeu du percussionniste consistait à exécuter les frappes 'types' qui avaient déjà été envisagées. J'avais alors pu emprunter une caméra de l'équipe 'Image et Son' du LaBRI, pour pouvoir visionner le mouvement dans son ensemble en même temps que j'observais les données issues des flocks of birds.

6.3.1 L'importance du jeu en contexte

Ce qui apparaissait le plus clairement sur l'enregistrement vidéo, était le manque d'implication et de concentration du percussionniste pour ces mesures (fig. 6.1). En effet, les flocks of birds étant un matériel particulièrement capricieux, il me fallait très fréquemment les réinitialiser, ce qui était très ennuyeux pour le percussionniste. De plus, l'enregistrement de coups isolés n'étant pas en soi un exercice passionnant, on peut supposer que l'expressivité musicale était tout à fait absente dans cette expérience.

Ainsi, tous les autres *mouvements d'accompagnement*³ du reste du corps, qui apparaissent dans une *situation de performance musicale*, n'ont pas ici l'influence qu'ils pourraient exercer. J'avais également demandé à Clément de jouer une séquence contenant différents coups, afin d'avoir un jeu plus en contexte. L'observation de la vidéo me confirma cette nécessité ; malheureusement un bug dans le programme d'acquisition fit que seule les données issues d'une baguette furent enregistrées.

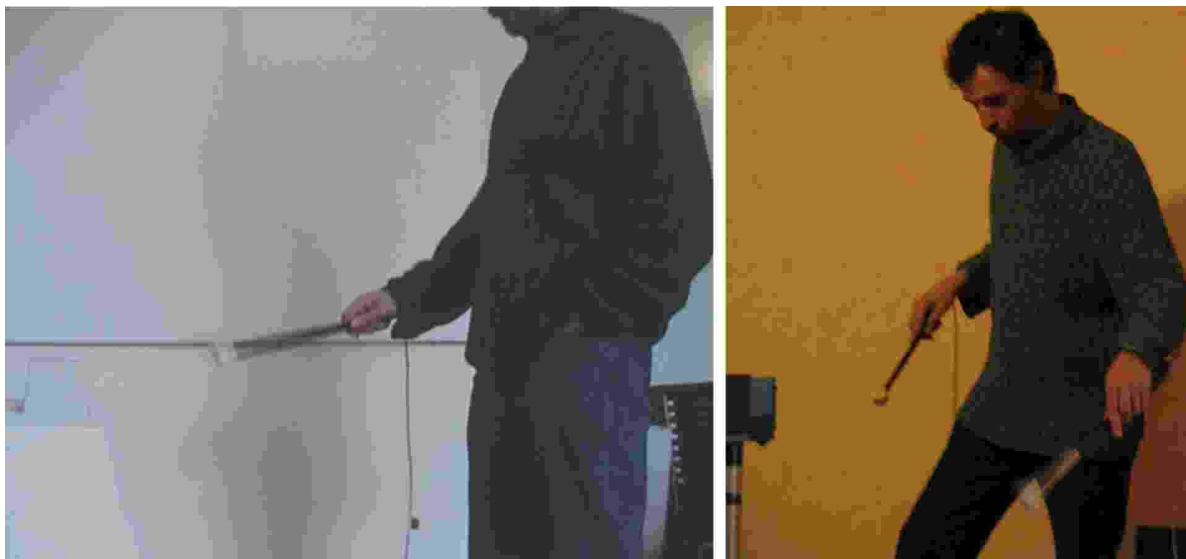


FIG. 6.1 – L'influence des conditions de jeu : Enregistrement de coups types et Performance réelle

³Cf. classification de F. Delalande, section 4.2.2.

6.3.2 Influence du retour auditif

Une autre question intéressante concerne le retour auditif qu'a le percussionniste de ce qu'il joue. Le système actuel pose en effet le problème de ne pas détecter tous les coups, et d'en déclencher d'autres à certains moments où il n'y aurait pas du en avoir, cela à cause de données erronées issues des capteurs et de défauts de l'algorithme de détection des impacts. Or la séance de mesures qui consistait à '*enregistrer 20 downs piano, puis 20 mezzoforte, puis 20 forte, puis 20 Ups piano... etc.*' prenait l'allure d'un test dont le résultat aurait été validé par le déclenchement d'un son, correspondant à une détection correcte de l'impacte.

Ainsi, alors que le but était d'enregistrer les gestes 'naturels' d'un percussionniste professionnel, j'avais l'impression en regardant les images que le percussionniste cherchait à déclencher le son, quitte à en modifier son jeu. Cette attitude est également liée au fait que dans une situation de concert, le percussionniste cherche à déclencher les sons⁴.

6.4 Deuxième acquisition : plus proche du jeu naturel ?

Cette fois, nous avons enregistré uniquement des séquences de jeu réel, basées sur une partition d'exercices rythmiques (fig. 6.2). J'avais demandé préalablement à Clément Fauconnet de préparer une partition contenant tous les types de coup et le maximum de diversité dans l'enchaînement de coups. Pour aider à l'analyse, la séquence était jouée à trois tempi définis, afin de pouvoir comparer les gestes lors de mouvement rapides et lents, et faciliter la segmentation. Les séquences ont donc été jouées avec l'aide d'un métronome, aux tempi 60, 100, et 132. Le jeu au métronome constituait en soi un écart par rapport au jeu libre pratiqué par Clément en situation de performance réelle, mais représentait une aide certaine pour pouvoir observer l'influence du tempo d'exécution sur des gestes semblables.



FIG. 6.2 – La partition jouée à 3 nuances et tempi différents

⁴D'autant plus que l'écriture de C. Havel qui joue sur l'interaction entre les différents instruments, ne facilite pas toujours la perception de correspondance entre geste et son

6.4.1 Influence du retour auditif

J'avais aussi décidé de réaliser ces enregistrements sans le système de synthèse sonore, afin que le retour auditif ne vienne pas influencer le jeu. En effet, le déclenchement des sons étant encore très aléatoire, il me semblait qu'il valait mieux éviter ce retour auditif.

Le jeu de Clément ce jour là était sans doute plus proche d'un jeu naturel, mais il me confiait à la fin des mesures que l'absence de son pour les '*frisés*' lui rendait difficile le comptage des coups pendant ces roulements. Il est intéressant de voir que c'est pour les frappes où le percussionniste utilise le plus le rebond sur la peau, que le retour sonore devient une aide d'autant plus importante. Il semblerait que le percussionniste contrôle alors sa gestique davantage par le retour auditif que kinesthésique.

6.4.2 Influence des gestes d'accompagnement

L'hypothèse de l'influence des gestes ancillaires ou d'accompagnement dans la performance réelle se trouve confirmée dans la séquence de jeu, avec notamment les gestes consistant à **marquer la pulsation**. On voit sur la figure 6.3, que le percussionniste marque le temps avant de commencer un roulement, comme pour *prendre appui*. Remarquons que ce geste s'accompagne fréquemment d'une inspiration synchrone, comme pour préparer un plongeon en apnée, lors de traits instrumentaux rapides.

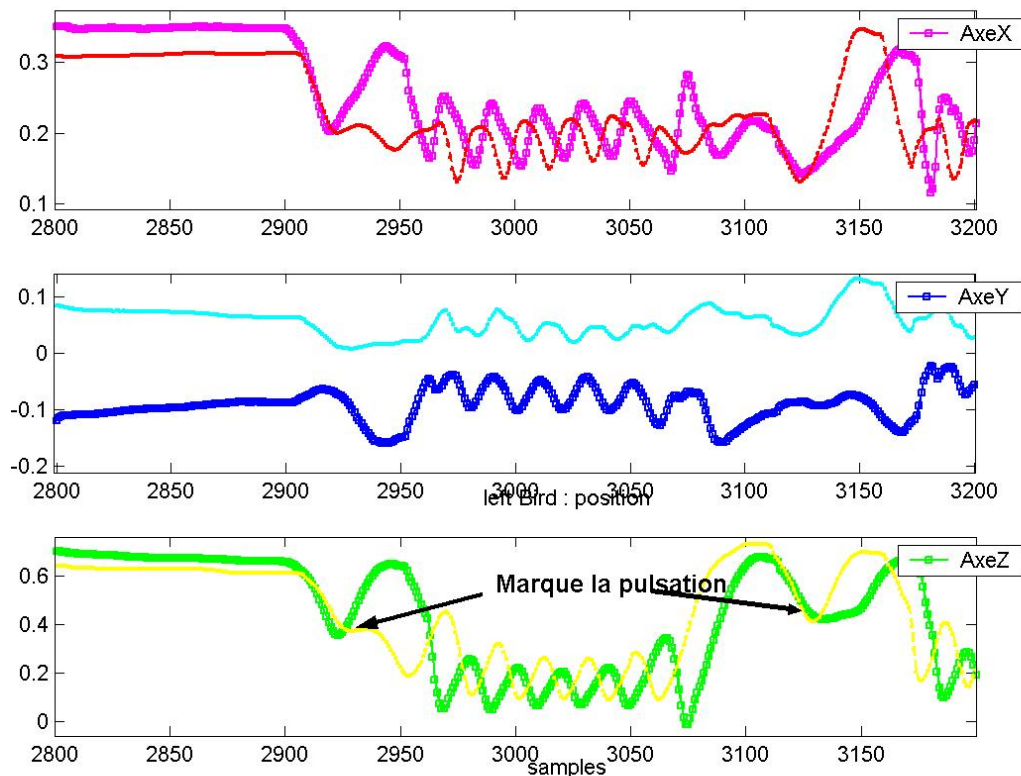


FIG. 6.3 – Les gestes d'accompagnements : inspiration sur la pulsation avant un frisé

On peut aussi remarquer que le mouvement des baguette suit une certaine symétrie tendant possiblement à établir un équilibre, une stabilité au niveau du reste du corps. Ainsi la baguette gauche aura tendance à remonter quand la droite descend et vice-versa.

6.5 Troisième acquisition : les mêmes séquences avec le son

Nous avons refait une séance de mesure sur la même partition, et avec les mêmes tempi et nuances, pour avoir davantage de données sur lesquelles établir des mesures statistiques. Mais suite aux remarques de Clément, les mesures furent faites avec le rendu sonore cette fois là. C'est en jouant un exercice rythmique très carré avec le métronome que le mauvais fonctionnement de la détection fût le plus flagrant. Le *tac* intransigeant du métronome, soulignait les retards aléatoires des déclenchements ; et le retour auditif des mauvais déclenchements sonores avait un effet très visible sur la difficulté du percussionniste qui faisait la grimace pour arriver à rester au tempo, en ayant dans les oreilles un son très décorrélé.⁵ Le retour auditif actif⁶ constitue donc une voie primordiale dans la chaîne d'un Instrument de Musique Digital, d'autant plus que le retour *tactile* est inexistant dans le cas de la percussion aérienne.

⁵Là encore, précisons que le percussionniste est relativement habitué à jouer avec un retour sonore décorrélé, car la plupart des 'Métamorphoses' de Christophe Havel sont des pièces dans lesquelles les gestes des instrumentistes influencent le son d'un méta-instrument davantage qu'ils ne produisent un son correspondant au son attendu d'un instrument traditionnel ; cependant il n'y a jamais dans ces pièces de tempo fixe comme celui d'un métronome, rendant la décorrélation moins perceptible et donc moins gênante.

⁶Retour *passif/actif* : Le retour *passif* désigne les bruits issus des caractéristiques physiques de l'instrument digital –inexistante dans le cas de la percussion aérienne–, tandis que le retour *actif* désigne le résultat sonore produit par une action particulière de l'instrumentiste. (Voir [Wan01] p17).

Chapitre 7

Analyse des signaux

7.1 Visualisation

Une des premières choses que j'ai eue à réaliser était un ensemble de programmes me permettant une visualisation aisée des différents paramètres, pour les différents fichiers.

7.1.1 Visualisation des données brutes et des paramètres

Un programme a été réalisé en Matlab permettant une représentation temporelle des données avec 3 fenêtres permettant l'observation simultanée des 3 axes de l'espace, ainsi qu'un menu pour choisir quelle donnée (position, angle, vitesse, accélération) afficher.

Position et angle

Les valeurs de position et d'angle, dans un espace (Oxyz) centré sur la base émettrice (Cf. fig7.1), sont renvoyée directement par la bibliothèque d'acquisition *libflock*, sous la forme de valeurs flottantes entre -1 et 1.

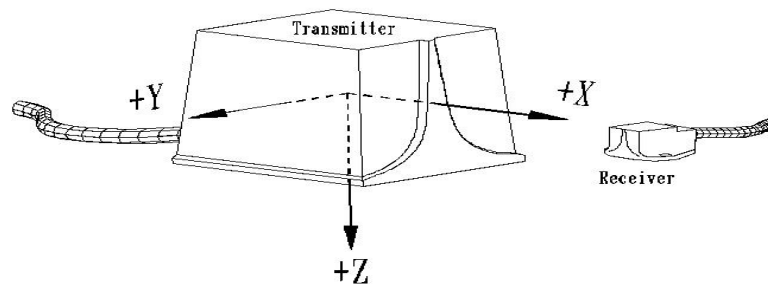


FIG. 7.1 – Le repère défini par rapport à la base émettrice

Vitesse, Accélération axiale et tangentielle

La vitesse est calculée pour chaque axe comme la différence entre la position reçue et la position précédente :

$$v(n) = p(n) - p(n - 1)$$

L'accélération est calculée pour chaque axe comme la différence entre la dernière vitesse calculée reçue et la vitesse précédente :

$$a(n) = v(n) - v(n-1) = p(n) - 2p(n-1) + p(n-2)$$

L'accélération tangentielle est calculée comme le produit scalaire entre l'accélération sur les 3 axes, et la *direction*. La direction est égale au vecteur vitesse divisée par sa norme.

$$a_{\text{tan}}(n) = \begin{cases} \vec{a}(n) \frac{\vec{v}(n)}{\|\vec{v}(n)\|} & \text{si } \|\vec{v}(n)\| > 0 \\ 0 & \text{sinon} \end{cases}$$

7.1.2 Visualisation des méta-données

Certaines données calculées de façon automatique à partir des enregistrements ont été enregistrées dans des fichiers séparés, afin d'éviter de toutes les re-calculer à chaque observation d'un enregistrement. Elle sont conservées dans un fichier de *méta-données*, comprenant les informations détaillées ci-après.

Pourcentage de prédiction

Afin de pouvoir étudier les courbes en tenant compte de la correction éventuellement apportée par l'algorithme de prédiction, le pourcentage de prédiction est représenté par des barres verticales (en jaune) au niveau de chaque échantillon.

Impacts détectés

L'algorithme de détection d'impacts rend compte des différentes étapes à franchir pour la détection effective d'un impact¹. Cet affichage permet d'étudier le comportement de l'algorithme de détection des coups face à la diversité des situations.

Notation manuelle des coups

A défaut d'avoir une détection parfaite des impacts et afin de pouvoir réaliser une analyse statistique, un programme permet l'indication manuelle des coups sur une séquence de données, par observation directe en cliquant sur les courbes. Cette segmentation manuelle a été utile pour une analyse statistique des paramètres de vitesse et d'accélération autour des positions d'impact (Cf. section 7.5)

7.1.3 Visualisation dynamique

Afin d'observer dynamiquement l'évolution des baguettes, un programme² permet de rendre une animation en trois dimensions des capteurs avec un repère pour indiquer leur angle par rapport à la base émettrice.

¹Cf. section 7.4

²Adapté d'un programme de la collection 'Bird I/O for Matlab' développé par Nathan Delson [Del01]

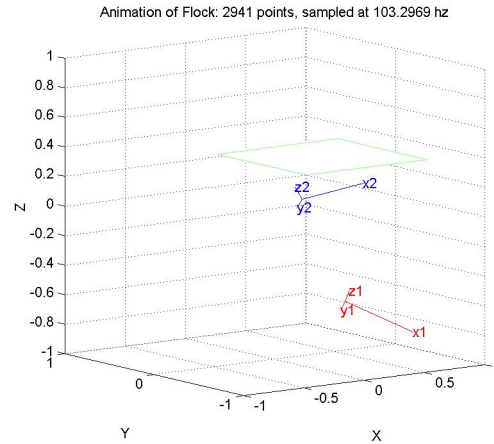


FIG. 7.2 – Animation en 3 dimensions : baguette gauche et droite

7.2 Valeurs erronées

7.2.1 Distorsion de l'espace

Les données de position et d'angle issues des capteurs sont calculées à partir de la valeur du champ électromagnétique au niveau des capteurs et suppose une certaine uniformité du champ statique. Cette condition qui n'est que partiellement respectée entraîne une déformation de l'espace représenté par les valeurs issues des capteurs par rapport à l'espace réel. Ces déformations posent un problème dans les applications de réalité virtuelle où l'utilisateur doit manipuler des objets dans le monde réel, dont la position est en décalage par rapport à ce qu'il peut voir sur un écran.

Cependant dans notre cas, le percussionniste ne voit pas les percussions et il s'adapte et compense ce décalage en testant 'auditivement' l'espace. Il est d'ailleurs assez raisonnable de penser qu'un percussionniste ne se sert que très peu de sa vue pour savoir où taper ; s'il on en juge le regard d'un percussionniste quand il joue, il sert généralement davantage à assurer un équilibre global du corps en se fixant un point d'horizon.

Lorsqu'on parle de mouvements complexes, le cerveau crée une 'plate-forme de guidage inertiel stabilisée' : la tête. La stabilisation est en rotation et l'angle que fait la tête avec l'horizon dépend de la direction du regard.[TrpOA93]

Une étude exhaustive des problèmes de calibration des capteurs électromagnétiques été menée par Kindratenko[Kin00], et des techniques de calibration permettent de corriger ces erreurs. Un étude sommaire a été réalisé pour mesurer cette distorsion (Cf. fig.7.3), consistant en la mesure de positions statiques du capteur sur un support quadrillé tous les 10cm. Cette mesure n'a pas montré de déformation suffisamment significative³ pour qu'un algorithme de calibration de l'espace (qui alourdirait encore le calcul sensé tourner en temps réel) soit implémenté.

³Inférieure à 5% sur une distance de 80cm

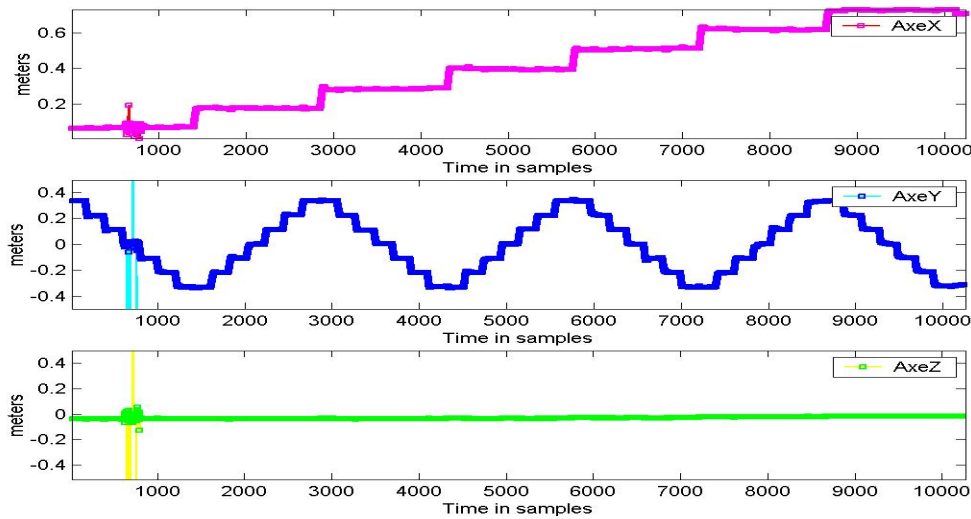


FIG. 7.3 – Étude de la distorsion statique de l'espace

7.2.2 Valeurs aberrantes

Un des problèmes majeurs des flocks of birds est leur grande fragilité et sensibilité aux perturbations électromagnétiques. Un certain nombre de problèmes apparaissent qui avaient été remarqués par les groupes précédents, mais qui n'avaient pas été corrigés. J'ai pu repérer principalement 5 types d'erreurs :

Erreurs dues à un retournement des axes

Ce premier type d'erreur n'est pas une erreur à proprement parler, mais une conséquence du fonctionnement des flocks par émission d'un champ électro-magnétique. Il se produit un retournement des axes Y et Z, lorsque l'on franchit le plan (oYZ). Ce défaut ne peut-être corrigé que dynamiquement, par un suivi de la trajectoire, car l'axe X, choisi comme axe de référence ne renvoie que des valeurs positives, des 2 cotés du plan (oYZ) : il n'est pas possible de savoir, pour une valeur fixe, dans quel sens sont orientés les axes.

Discontinuité des valeurs d'angle

De la même manière que pour le retournement des axes, les valeurs d'angle évoluent dans l'intervalle $[-1, 1]$, et présentent des sauts de modulo 2 lorsque l'on dépasse cette limite. Ce problème doit également être corrigé dynamiquement, par un suivi de la trajectoire.

Erreurs dues à une saturation des capteurs

On a pu observer un phénomène de blocage des données, apparaissant principalement aux instants de forte vitesse. Ce type d'erreur, facilement détectable, pose toutefois le problème d'entraîner rapidement le modèle de prédiction.

Une solution consisterait alors à déplacer les capteurs en amont de la baguette pour qu'il subissent des vitesses moins élevées. Il faudra alors que la position de l'extrémité de la baguette,

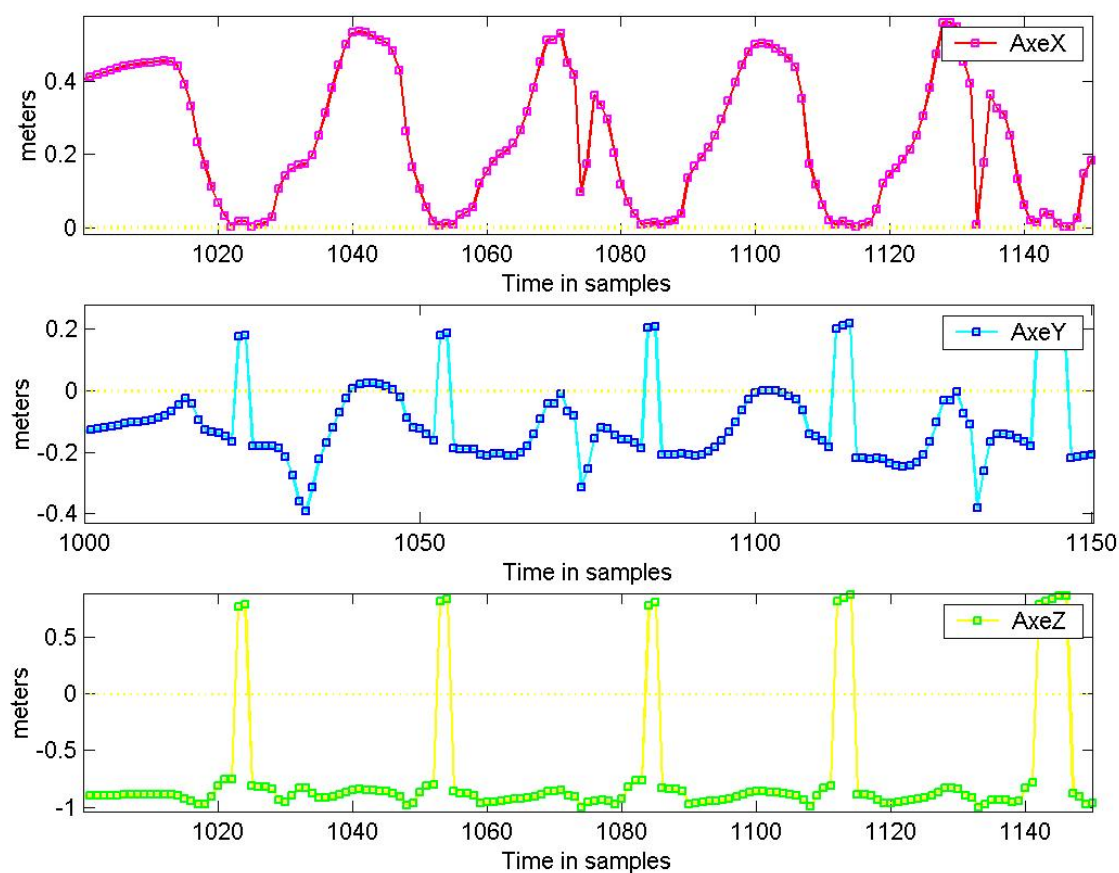


FIG. 7.4 – Retournement des axes par franchissement du plan (oYZ)

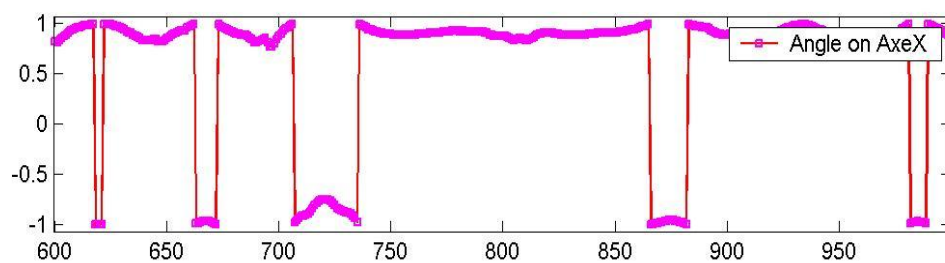


FIG. 7.5 – Discontinuité des valeurs d'angle

nécessaire pour connaître le point d'impact soit calculée en fonction des données d'angle et de position, mais cette méthode ajoute une erreur supplémentaire quant à la position exacte de l'extrémité en la rendant notamment dépendante de la fiabilité des données d'angle. Une position intermédiaire des capteurs est également gênante pour l'équilibre de la baguette.

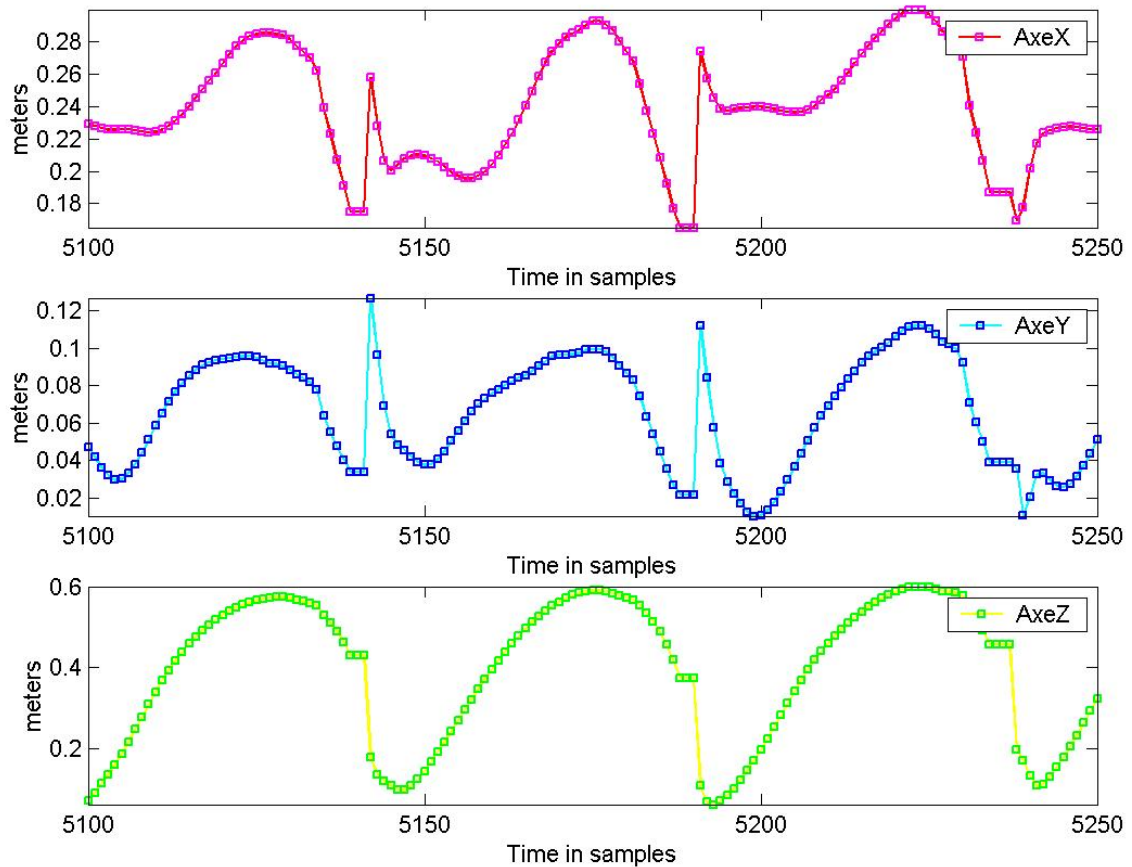


FIG. 7.6 – Les données bloquées dues à une saturation des capteurs

Erreurs dues à des perturbations électro-magnétiques

Un autre type d'erreur se présentant comme des perturbations aléatoires localisées dans le temps a été attribuée à des perturbations électromagnétiques.(cf fig. 7.7) La technologie des flocks of birds par champ électromagnétique pulsé les rend sensible à l'environnement magnétique dans lequel ils sont utilisés. Malheureusement, la présence d'ordinateurs, d'écrans cathodiques, et d'autres structures métalliques est assez fréquente dans le cadre de la percussion aérienne.

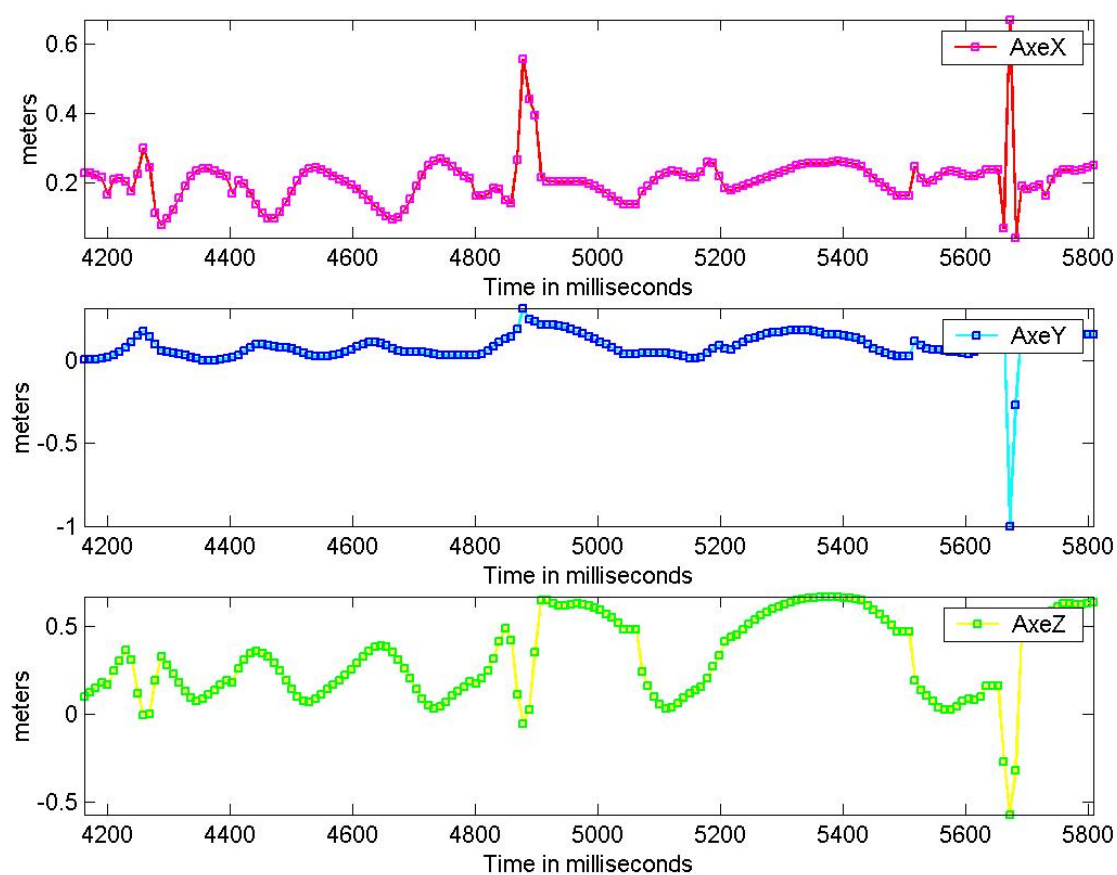


FIG. 7.7 – Influence de perturbations électromagnétiques sur les données

Les erreurs dues à des défauts de contact

Un autre inconvénient de ces capteurs est qu'ils nécessitent des fils reliant les 'birds' au flocks, et qui ne sont pas toujours adaptés aux mouvements violents des gestes de percussion.

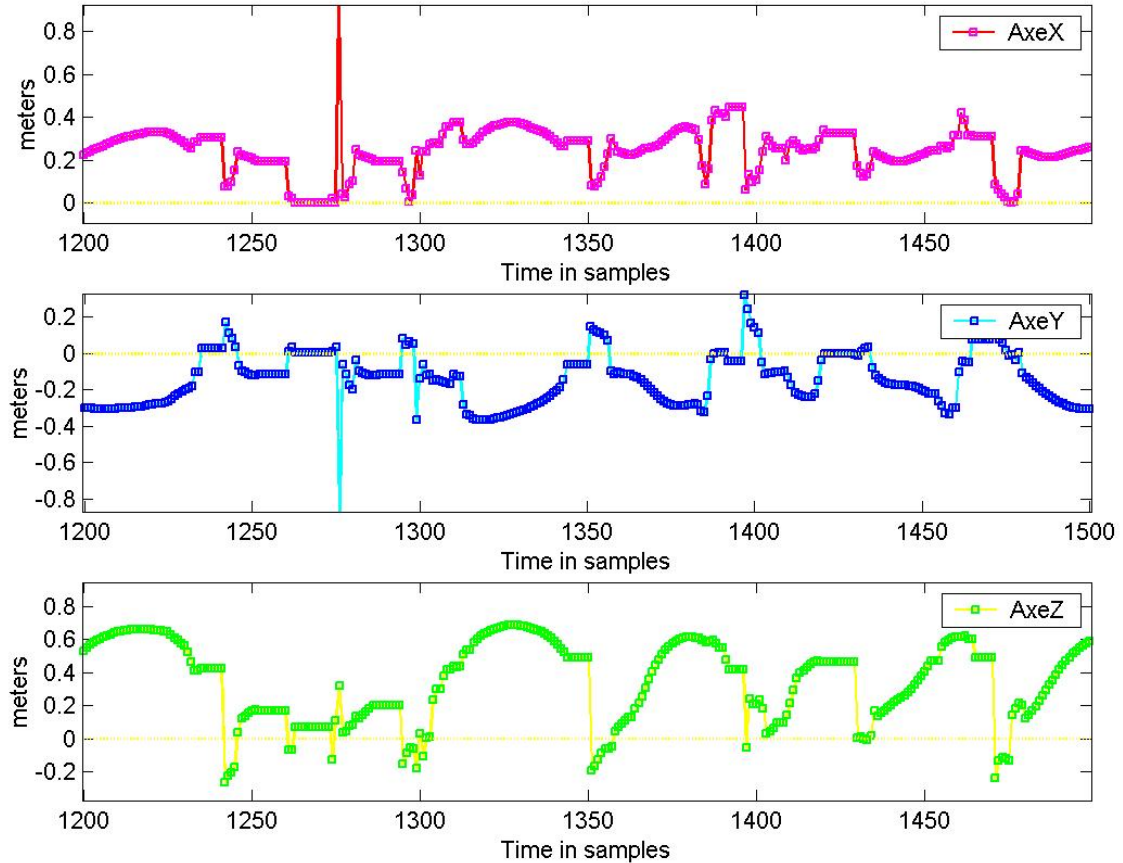


FIG. 7.8 – Problèmes de contact

La correction de ces erreurs a occupé une grande partie du stage car la reconnaissance de coup qui devait se faire sur une étude statistique d'un ensemble de données mesurées ne pouvait en aucun cas être réalisée sur des mesures comportant de telles aberrations qui fausseraient complètement les variances.

7.3 Algorithme de correction

Prediction is like driving a car when the only available view is the rear view mirror. The driver can see what happened in the past, but he has no direct view of the future.(...) To keep the car on the road, the driver must anticipate, or predict, where the road will be, based solely on the past observations and his knowledge of

roads in general. The difficulty of this task depends on the shape of the road and how fast the car is going. If the road is straight and remains so, then the task is easy. If the road twist and turns rapidly, the task may be nearly impossible.[Azu95]

L'algorithme de correction est basé à la fois sur des considération physiques et sur un modèle de prédiction linéaire (LPC). Nous allons d'abord présenter rapidement le principe de la prédiction LPC, puis l'algorithme de correction l'utilisant.

7.3.1 LPC et méthode de Burg

L'objectif initial de l'utilisation d'un modèle LPC était de prévoir le moment d'impact. Les modèles LPC ont été utilisé avec succès pour extrapoler des signaux audio[IKS01], et des trajectoires de partiels harmoniques[MLR03].

Nous supposons que les gestes étudié ici font partis d'une même classe de signaux gestuel que les premiers. On constate que la prédiction linéaire par méthode de Burg donne de très bon résultats pour des signaux suffisamment stationnaires (fig. 7.9).

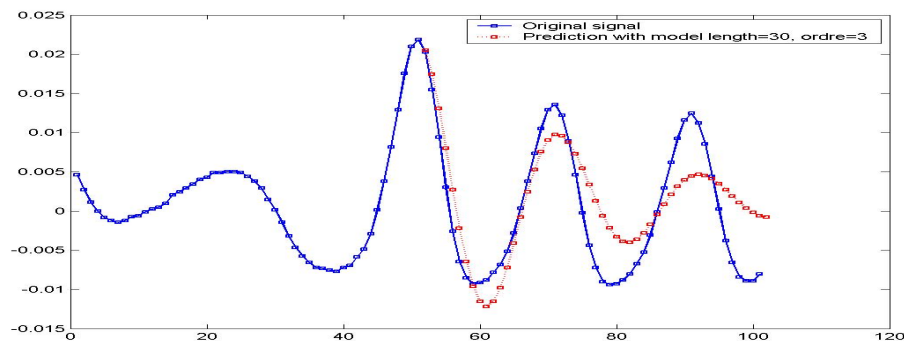


FIG. 7.9 – Prédiction d'un *frisé* par modèle LPC coefficients calculés par la méthode de Burg

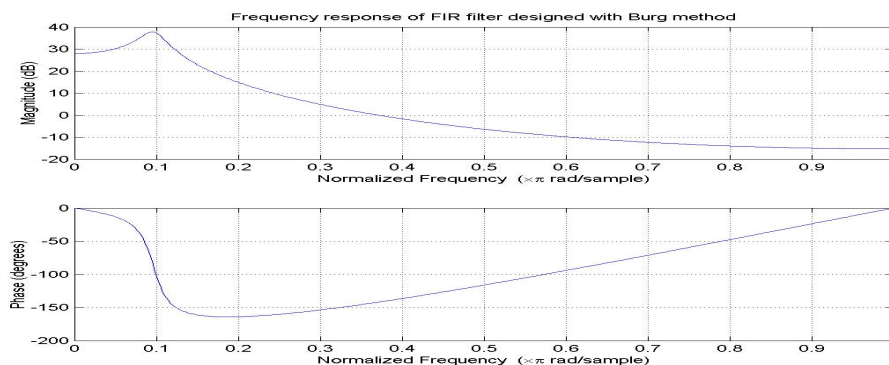


FIG. 7.10 – Réponse en fréquence du filtre pour la prédiction du frisé fig.7.9

Dans les modèles de prédiction linéaire (LP) autoregressifs (AR), l'échantillon courant $x(n)$ est approximé par une combinaison linéaire des échantillons passés du signal d'entrée. Nous nous intéressons alors à trouver un vecteur a de N coefficients, N étant l'ordre du modèle LP. La valeur prédite \hat{x} est alors calculée par filtrage FIR

$$\hat{x}(n) = \sum_{i=1}^N a_i x(n-i)$$

Parmi les différents algorithmes utilisés pour calculer les coefficients d'un modèle LP, l'algorithme de Burg a été choisi sur les conseils de Sylvain Marchand, en raison de son succès dans le suivi de trajectoires d'harmoniques, à produire des signaux synthétiques proche de signaux naturels. De plus, la méthode de Burg conduit à des filtres à minimum de phase (i.e. tous les pôles du filtre sont inclus dans le cercle unité), assurant une réponse rapide, et stable, ce qui n'est pas le cas pour la méthode par covariance.

La méthode de Burg est présentée dans ses aspects théorique et computationnel dans [FKZ00] et explicitée en annexe A. Pour ce qui nous concerne, les paramètres importants de l'algorithme de prédiction sont l'*ordre du filtre* et la *taille du modèle* utilisé pour estimer les coefficients LP. Les valeurs retenues ont été choisies après des considérations théoriques et empiriques.

L'évolution des signaux gestuels comprend des variations exponentielles (accélération et décélération) et des variations pseudo-sinusoïdales (frisés) : il faut donc un ordre minimum de 2. D'un autre côté, un ordre trop élevé entraîne des oscillations non désirables. À défaut de connaître l'intention du percussionniste, nous pouvons extrapoler son mouvement dans une certaine mesure, en l'amortissant progressivement vers sa position moyenne dans l'espace.

La taille du modèle doit être assez grande pour détecter la périodicité, et suffisamment courte pour ne pas être trop influencée par les évolutions passées. La période la plus courte des gestes *stationnaires*, relevée dans des *frisés* (Cf. fig. 7.9), est proche de 200ms (5 Hz), ce qui correspond à une longueur de 20 échantillons dans notre système. Heureusement [pour la musique], ou malheureusement [pour notre algorithme de prédiction], le jeu de percussionniste est loin d'être stationnaire...

Une étude a donc été réalisée sur une série de données exempte d'erreurs, pour diverses valeurs d'ordre et de taille du modèle, en cherchant à minimiser l'erreur quadratique de prédiction⁴, pour une prédiction des 10 échantillons suivants (100ms). On peut en effet supposer que l'inertie du corps entraîne un mouvement prévisible sur quelques dizaines de millisecondes⁵.

Un modèle de 30 échantillons, et un filtre d'ordre 3 semblent donner les meilleurs résultats.

7.3.2 Description de l'algorithme de correction

Correction des retournements d'axes et saut d'angle

La correction de ce problème assez facilement identifiable est faite en premier, en détectant le retournement d'axe ou le saut d'angle par comparaison entre la valeur brute reçue et sa

⁴i.e. la différence quadratique entre le signal prédit et le signal réel. Nota : le triangle supérieur droit de la figure 7.11 correspond à des ordres supérieurs à la taille du modèle, ce qui n'est pas réalisable (système sous-déterminé)

⁵Une étude de la spécificité des gestes et de leurs enchaînements pourrait peut-être repousser cette limite, mais le modèle LPC utilisé ici est agnostique

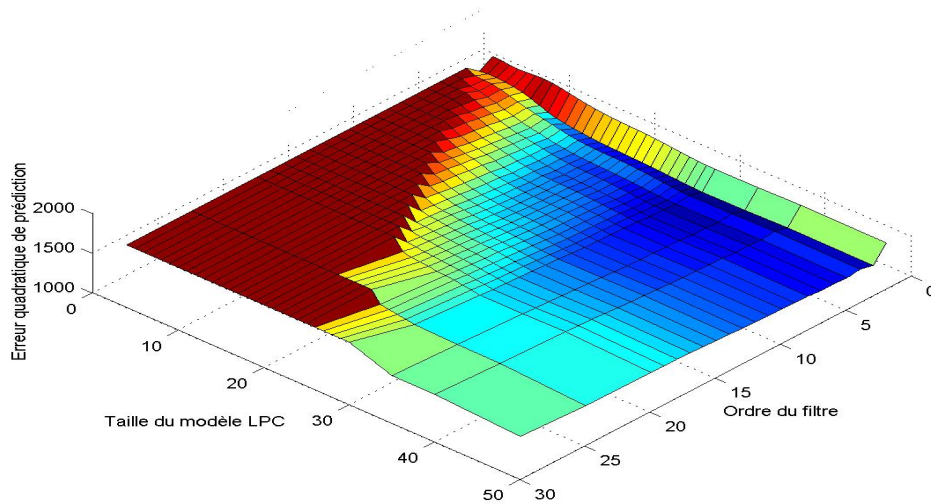


FIG. 7.11 – Erreur quadratique de prédiction des 100ms en fonction de l'ordre et la taille du modèle LPC

prédiction, puis en prenant l'opposé des valeurs dans le cas du retournement d'axe, et en ajoutant/retranchant le modulo [2] pour les valeurs d'angle.

Évaluation de la viabilité des données

L'algorithme de correction consiste ensuite, à calculer un coefficient de pondération, à chaque nouvelle valeur acquise, selon que l'on détecte :

- un blocage des données
- une valeur de l'accélération tangentielle, qui ne respecte pas certains seuils correspondant aux limites de la force physiques du percussionniste, ainsi que de la dérivée de cette accélération, limitée par le temps de réaction des fibres musculaires à la contraction⁶.

La vitesse corrigée est ensuite calculée comme la somme pondérée de la vitesse 'brute' et de la vitesse estimée par LPC. Puis, on remonte à la position, qui est calculée comme la somme pondérée de la position 'brute' et de l'intégration de la vitesse corrigée. Cette deuxième pondération permet d'éviter une dérive progressive de la position par accumulation d'erreur lors de l'intégration de la vitesse estimée durant une trop grande période.

Le retour aux valeurs réelles, une fois la zone de perturbation passée se fait progressivement, en diminuant le pourcentage de prédiction en fonction de la distance entre les valeurs corrigées et les nouvelles valeurs réelles reçues calculée comme suit :

$$Dist(n) = \|p_{\text{corrigée}}(n) - p_{\text{réelle}}(n)\|$$

⁶La dérivée de l'accélération tangentielle est la dérivée 3ème de la position, et son utilisation est très contestable vis à vis de la fréquence d'échantillonnage de 103Hz des flocks of birds. Cependant il est prévu d'améliorer prochainement la percussion aérienne en ajoutant des accéléromètres avec une fréquence et une fiabilité suffisante pour exploiter ce paramètre.

... et le pourcentage de prédiction :

$$\%Pred = \min(\max(0, \%Pred - 0.01), K \times Dist_{\text{valeurs réelle}})$$

Le code Matlab de cette fonction est présentée en annexe pour plus de détails.

Détermination des seuils

Le signal de position issu des capteurs renvoie l'image de l'extrémité de la baguette dans l'espace. Lors d'une frappe dans le vide, le spectre du signal gestuel dans le vide nous indique qu'il y a très peu d'énergie dans les fréquence supérieures à 200Hz pour le signal d'accélération tangentielle (Cf. [Ver04]).

Les valeurs limites de la vitesse, et de l'accélération tangentielle du capteur situé au bout de la baguette, à 20cm environ de l'articulation du poignet, ont été évaluées empiriquement en prenant le maximum dans les données acquises, pour un *frisé forte*, et le minimum dans une séquence *piano*.

Notons qu'au fur et à mesure des dérivations, ces valeurs sont de plus en plus difficiles à relier à des phénomènes physiques, car on se situe vraiment à la limite de résolution temporelle permise par la fréquence d'échantillonnage à 103Hz. Marrin Nakra évalue cette fréquence à 450Hz[Nak00]. Les résultats⁷ de ces valeurs extrêmes sont rassemblées dans le tableau 7.1.

TAB. 7.1 – Valeurs minimales et maximales des paramètres

Paramètre	Valeur min	Valeur max
Norme de la vitesse	$1m/s$	$7m/s^2$
Accélération tangentielle	$20m/s^2$	$160m/s^2$
Décélération tangentielle	$40m/s^2$	$180m/s^2$
Dérivée accélération tan.	—	$35000m/s^3$

7.4 Détection des impacts

La détection des impacts devait s'avérer être un compromis entre une approche physique, décrivant son approche en trois phase :

- 1 une accélération
- 2 un pic de vitesse
- 3 une brusque décélération
- 4 un changement de direction

... et tenant compte des contraintes matérielles – notamment la limitations de la fréquence d'échantillonnage à une valeur en deçà des valeurs maximales que peuvent prendre les données gestuelles ([Nak00]), et la validité des données.

Le même modèle LPC que celui utilisé pour la correction nous sert ici à anticiper l'impact, qui est reconnu lorsque la direction change de sens et passe quasiment .

⁷NB : les données exprimées ici sont à relativiser quant à l'exactitude du mètre...

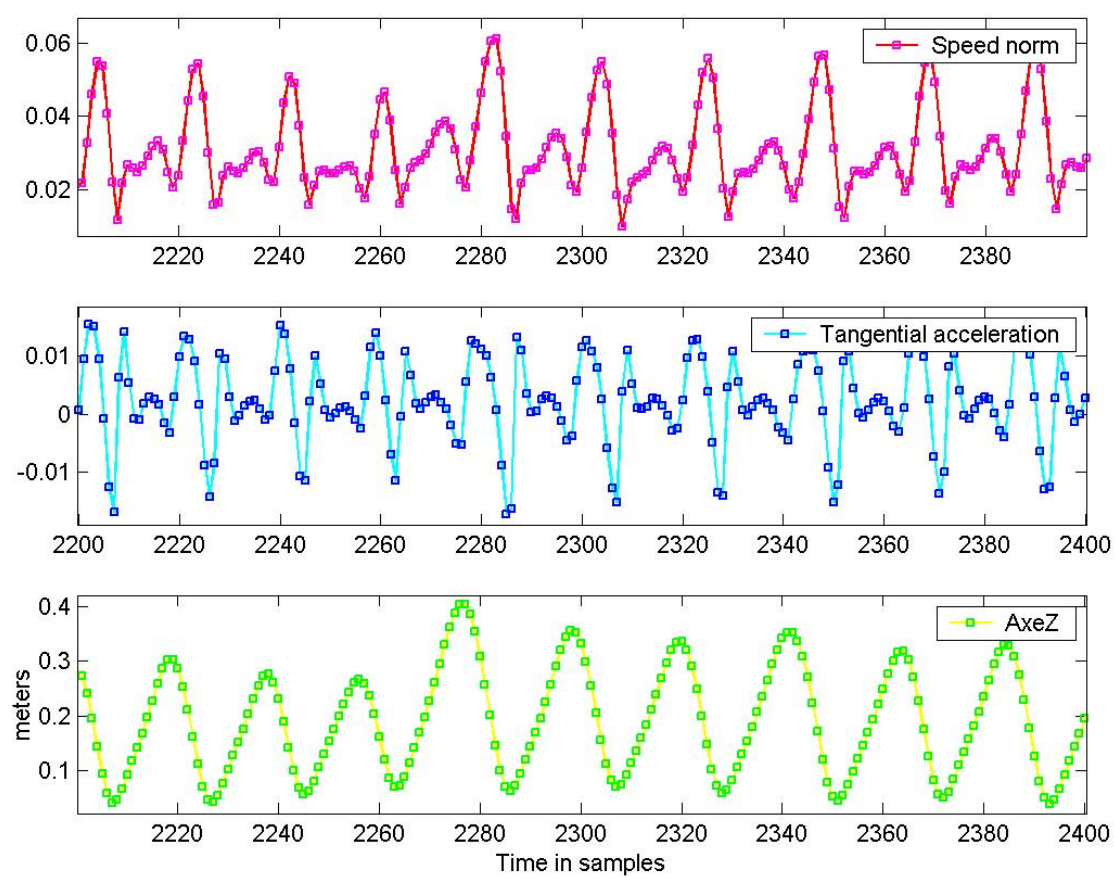


FIG. 7.12 – Valeurs maximales de l'accélération et de sa dérivée

7.5 Analyse des types de coups

Le compositeur souhaitait que les types de coups correspondant à la classification élaborée par François Dupin soient reconnus, avec l'idée de reconnaître par la suite une phrase rythmique, comme un enchaînement de ces coups de base.

La détection des types de coup pose le problème de savoir ce qui identifie ce coup. Plusieurs descriptions en avaient été faites dans les rapport précédent. Mais l'approche consistait en un modèle théorique qu'on voulait coller sur la réalité –virtuelle– au lieu d'analyser la réalité pour en déduire un modèle.

7.5.1 Extraction de statistiques de vitesse et accélération

L'analyse statistique de donnée *de jeu*, c'est à dire dans le contexte de mouvement corporel global du percussionniste jouant un enchaînement divers de frappes a été réalisé, afin de déterminer les valeurs significatives identifiant ces coups. Dans l'idéal, une segmentation automatique des enregistrement, grâce à la partition jouée (Cf. 6.2), aurait permis une analyse à grande échelle relativement facile. Malheureusement, la mauvaise qualité des données d'acquisition et leur imparfaite correction ne permettait pas une analyse aussi automatique, car les données aberrantes faussaient très fortement les statistiques.

J'ai donc réalisé un petit programme permettant de sélectionner les impacts visuellement, là où les données étaient correctes. Un programme d'analyse statistique servit ensuite à calculer les moyennes et variances de paramètres autour des points des points d'impacts. Les résultats sont montrés au chapitre suivant (fig. 8.5 et 8.6).

7.5.2 Extraction des types de coups par corrélation

Dans un deuxième temps, un programme a été réalisé pour évaluer, à chaque échantillon nouveau, la correspondance entre l'allure du mouvement et celles évaluées statistiquement pour chacun des types de coups. Pour faire cela, le 1er coefficient de corrélation est calculé pour chaque type de coup, sur les paramètres de vitesse et d'accélération préalablement normalisés en énergie. Cette normalisation permet de s'affranchir de l'amplitude du coup, afin de *ne comparer que les formes*.

$$Simil_{speed}(strike) = \sum_{i=1}^N v_{strike}(i) v_{normal\ N}(i)$$

$$Simil_{accel}(strike) = \sum_{i=1}^N a_{strike}(i) a_{normal\ N}(i)$$

Chapitre 8

Résultats

8.1 Correction des données

La correction des données n'a pas permis de fiabiliser à 100% le système, mais a permis au moins d'avoir des données de positions relativement continues, qui à défaut d'empêcher toutes les détections d'impacts liés aux perturbations, limite le déclenchement des coups grâce au filtrage par les instruments¹.

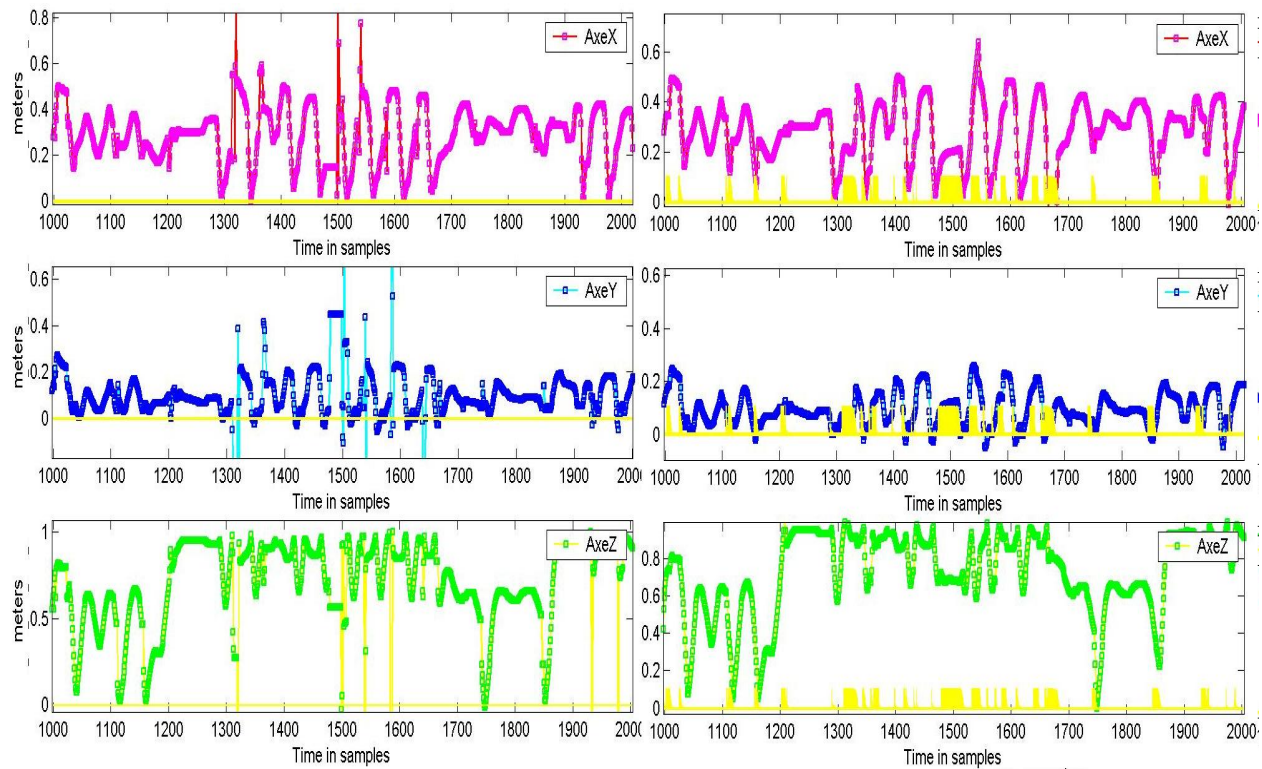


FIG. 8.1 – Position non corrigée, et corrigée. Les barres jaunes représente le pourcentage de prédiction.

¹Lorsqu'un impact est détecté, par exemple suite à une perturbation, on peut vérifier qu'il ait lieu *dans un instrument* avant de déclencher un événement.

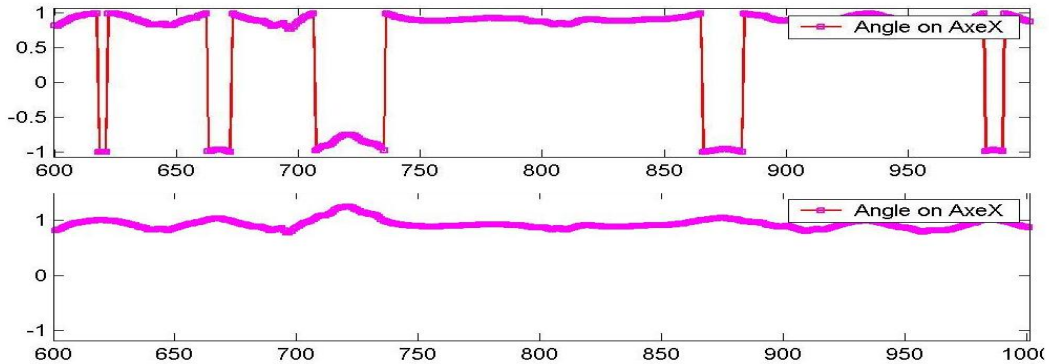


FIG. 8.2 – Correction des sauts d'angle.

8.2 Détection des impacts

Les impacts sont détectés comme le moment où la vitesse tangentielle subit un changement de direction d'un angle supérieur à la normale à la vitesse tangentielle calculée 2 échantillons (= 20ms) plus tôt. A l'aide de fichiers segmenté manuellement, on a pu estimer (Cf. tableau 8.1) la diminution du nombre de 'fausses alertes' à un rapport de 2 (sans compter le filtrage par l'instrument), et le nombre de coup non-détectés à un rapport de 4². La distance par rapport au moment précis de l'impact a été également fiabilisée

TAB. 8.1 – Comparaison nouvel algorithme / ancien algorithme

OK : Impact Détecté – FA : Fausse Alerte – ND : Non Détecté

σ : Ecart moyen par rapport l'instant d'impact (échantillons)

Fichier	Nb de coups	Avec modèle LPC				Algorithme précédent			
		OK	FA	ND	σ	OK	FA	ND	σ
seq.060.mf.01	67	87%	176%	13%	0.5	44%	337%	55%	0.7
seq.060.ff.01	99	89%	117%	11%	0.44	41%	212%	59%	0.63
seq.060.ff.02	76	92%	151%	8%	0.38	45%	289%	55%	0.55

Anticipation grâce à la prédiction

L'utilisation de l'algorithme de prédiction pour la correction des données permet d'anticiper la détection d'un coup, et de compenser ainsi le retard dans la transmission au logiciel de synthèse sonore (cf fig.8.3 et 8.4).

²Il ne faut cependant pas prendre les valeurs du tableau 8.1 comme ayant une signification absolue : en effet les frappes indiquées manuellement ne l'ont été que pour les endroits où le signal était propre ; ce tableau est donc essentiellement là pour illustrer un ordre de grandeur pour la comparaison avec l'algorithme précédent.

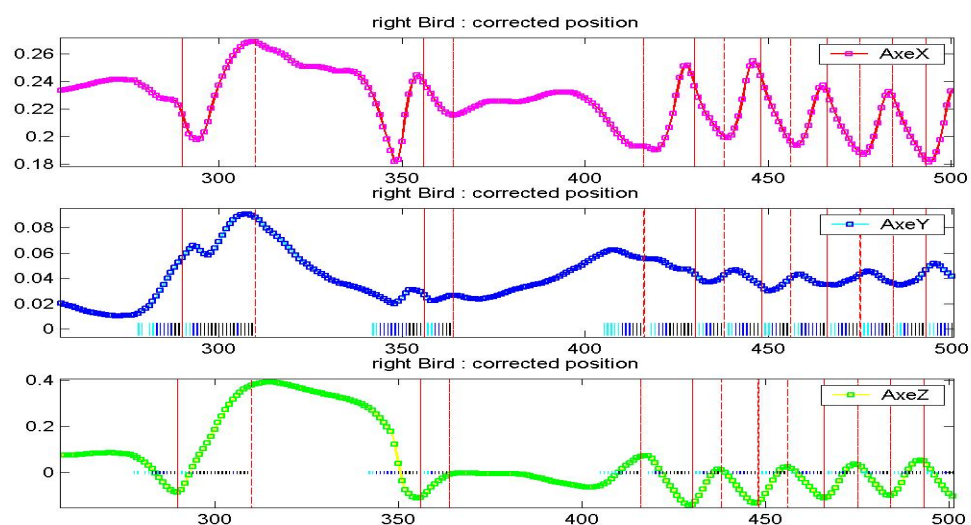


FIG. 8.3 – Détection de l'impact sans anticipation

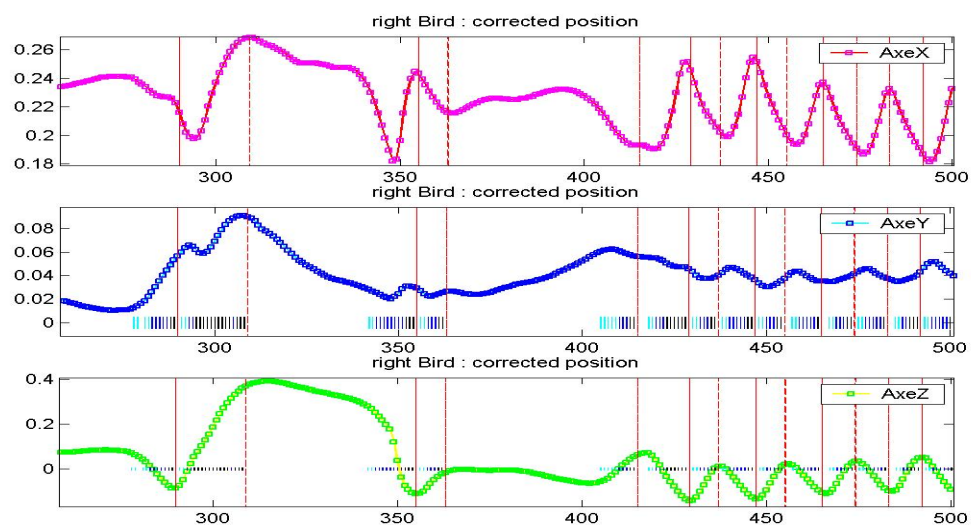


FIG. 8.4 – Détection de l'impact avec anticipation

8.3 Analyse de types de coups

Cette analyse est en cours d'élaboration et les résultats présentés ici sont assez sommaires, mais permettent d'ores et déjà de dégager des formes caractéristiques.

8.3.1 Analyse des paramètres

L'analyse des types de coups se base sur une analyse statistique des paramètres de vitesse et d'accélération. Il s'agit de voir si ces paramètres permettent une discrimination des différents types de coups.

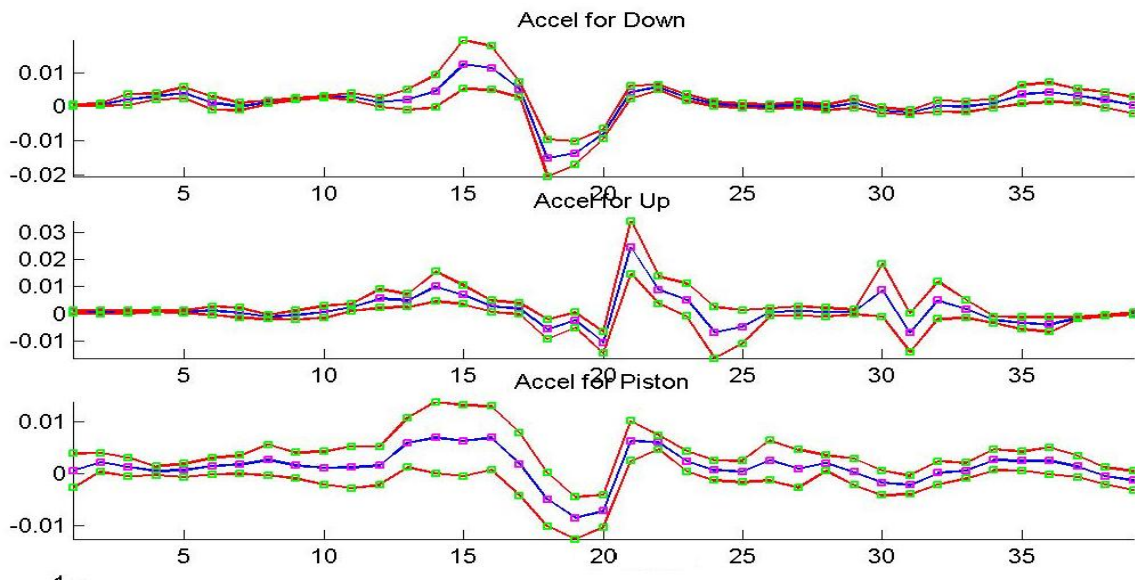


FIG. 8.5 – Statistiques pour l'accélération tangentielle autour de l'impact

8.3.2 Extraction des types de coup

On observe une discrimination satisfaisante des types de coups ³ grâce au calcul de la similarité *sur le paramètre de vitesse*, au moyen du premier coefficient de corrélation des fenêtres temporelles normalisées en énergie : au moment de l'impact, on a bien une similarité plus forte –même si la valeur n'est que très légèrement supérieure– avec le coup réalisé.

Remarquons que pour des données assez 'propres', la similarité pondérée par l'énergie du signal donne une assez bonne estimation de la probabilité d'apparition du coup, en donnant une importance aux gestes de moindre énergie.

Le calcul de la similarité *sur le paramètre d'accélération* n'a pas donné de discrimination satisfaisante, en étant trop soumis aux perturbations.

³Souvenons nous que les valeurs de similarité présentée fig. 8.7 sont calculées sur une fenêtre de 40 échantillons *autour de l'impact*. Cette méthode ne permet donc pas ici de discriminer *en temps-réel* les types de coups comme on le souhaiterait, c.à.d. avec les seules données *précédant l'impact*.

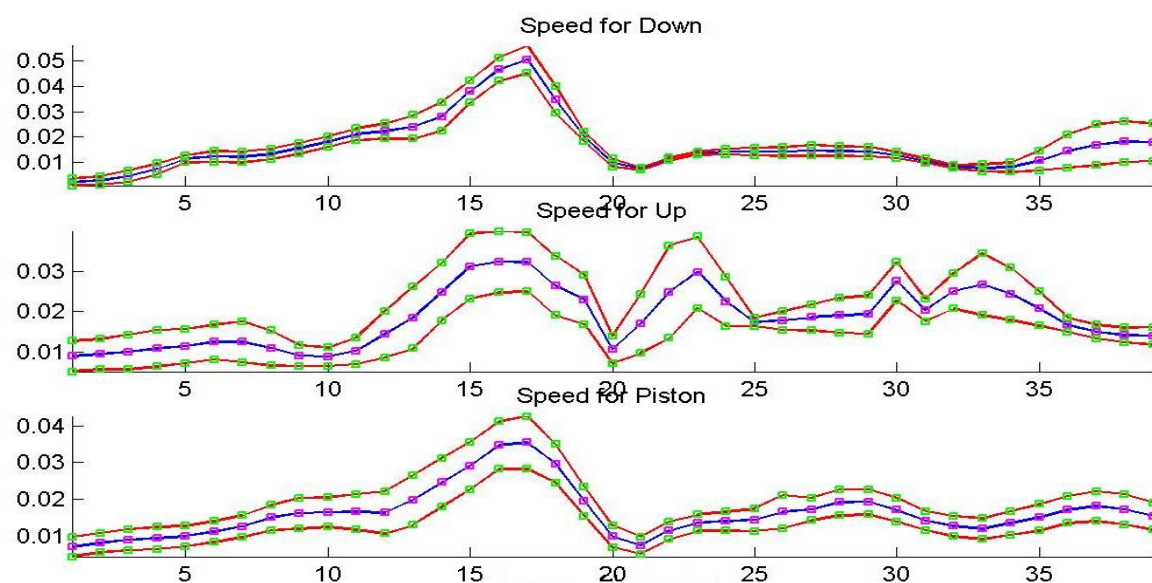


FIG. 8.6 – Statistiques pour la norme de la vitesse autour de l'impact

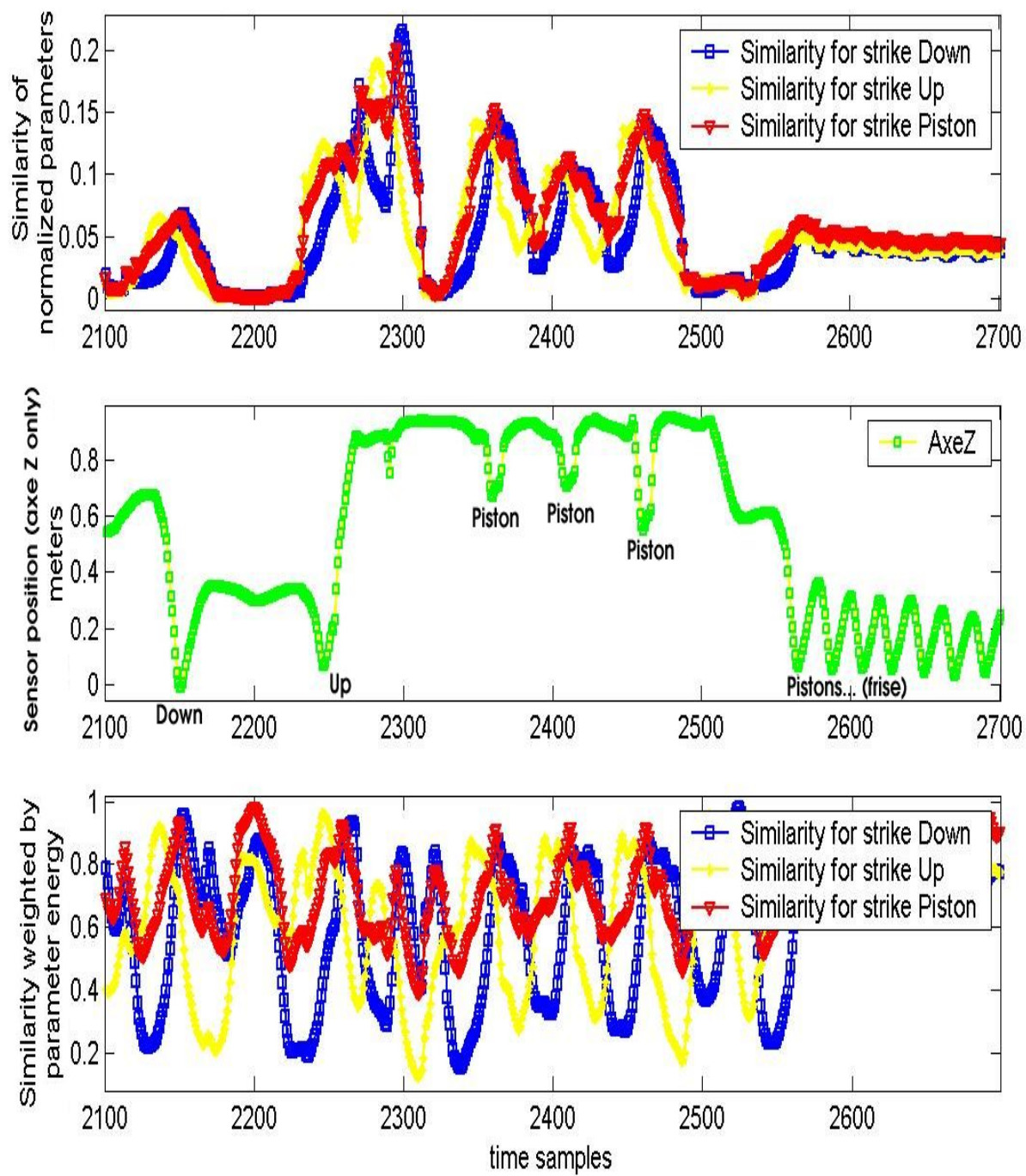


FIG. 8.7 – Similarité pour un enchaînement Down, Up, 4 Pistons – puis frisé

Haut : Similarité multipliée par l'énergie – *Milieu* : position du capteur sur l'axe Z – *Bas* : Similarité

Chapitre 9

Bilan et perspectives

9.1 Bilan

On a présenté ici un algorithme de correction de données basé sur un modèle LPC, adaptatif en fonction de la validité de données d'entrée au regard des possibilités physiques du percussionniste, et fonctionnant en temps-réel.

L'utilisation d'un modèle prédictif a permis de compenser le retard qu'entraîne l'attente d'un changement brusque de direction pour la détection d'impacts.

Une analyse statistique des paramètres de vitesse et d'accélération, dans une situation de jeu réelle, a été réalisée concernant la possibilité d'identification de primitives gestuelles du jeu de percussion classique.

Des considérations ont été notées sur les différentes influences qui peuvent s'exercer sur le jeu de l'instrumentiste, dans le cadre du jeu avec un instrument virtuel, qu'elles soient liées à des problèmes techniques, au protocole de mesure, ou encore à des réactions face à la décorrélation entre le monde réel et virtuel.

On a également proposé une ébauche générale du design de la percussion aérienne en essayant d'explicitier les différents choix qu'entraîne la relation désirée entre le geste, la synthèse sonore, et la composition formelle. Enfin, des remarques ont été apportées concernant le jeu du percussionniste, sur l'organisation spatio-temporelle de ses mouvements, ouvrant la voie vers des investigations plus fines.

On est donc encore loin du résultat souhaité, et l'extraction des types de coup qui permettrait d'accéder à une composition formelle en temps-réel n'est pas encore opérationnelle. Les résultats présentés dans ce mémoire semblent tout de même assez encourageants, et laissent penser qu'un affinage des paramètres permettrait d'atteindre cet objectif.

9.2 Perspectives

Portage en C : utilisation temps réel

Le développement des algorithmes sous Matlab va être porté en langage C pour une utilisation en temps-réel.

Évolution du matériel

Le dispositif d'acquisition 'flock of birds' utilisé lors de ce stage a causé de nombreux problèmes techniques comme il a été montré dans ce rapport. Une interface d'acquisition plus fiable et avec une meilleure résolution temporelle permettrait une considérable amélioration de l'analyse, notamment au niveau de l'étude statistique.

Reconnaissance des coups et des enchaînements

La reconnaissance des enchaînements présentée ne permet pas à l'heure actuelle de discriminer les types de coup *avant l'impact*. Une étude plus approfondie de cette partie du geste, en étudiant notamment la coordination de la main par rapport à l'avant bras, pourrait donner de meilleurs résultats. En effet, nous pensons que si le retour de la baguette est effectivement pensé avant l'impact, il est fort probable que cette anticipation soit visible au niveau de l'articulation du poignet.

Dans le cas où les analyses statistiques ne permettraient pas une robustesse suffisante, une segmentation des signaux gestuels par des algorithmes d'apprentissage comme les réseaux de neurones (Cf. [ACH04]) ou les HMMs¹ (Cf. [Mul04]) permettrait de comparer de manière agnostique les différents coups.

L'extraction de descripteurs moins évident pourrait être envisagée de manière automatique par des algorithmes génétiques (Cf. [Zil04])

Analyse de jeu d'autres percussionnistes

Toutes les mesures ont été réalisées avec le percussionniste Clément Fauconnet qui travaille en étroite collaboration avec Christophe Havel. Il sera très utile de refaire ce travail d'analyse avec d'autres percussionnistes, afin d'avoir une vision plus générale des caractéristiques de la gestique des percussionnistes, qui est de toute évidence très diversifiée.

Analyse de style

Il serait intéressant de mener une étude de l'évolution du geste en mesurant les positions des différentes articulations du bras. La préparation du geste est en effet amorcée au niveau du corps (torse / épaule) et se propage, *en se déroulant le long du bras*. L'étude de la dynamique et de l'interaction de ces différentes parties du corps pourrait permettre une meilleure anticipation des gestes.

Enfin, une fois la reconnaissance de coups fiabilisée, il sera sûrement intéressant d'étudier les motifs rythmiques qui se dégagent du jeu d'improvisation, à l'aide des outils de calcul symbolique et d'apprentissage², et d'ouvrir la voie pour la percussion à toutes les formes d'interaction avec l'ordinateur qui ont déjà été expérimenté avec les instruments MIDI notamment.

¹Hidden Markov Model : Modèle de Markov Caché

²Comme *OpenMusic* de l'IRCAM, le *continueur* de Sony CSL, ou bien encore le *BolProcessor* de Bel & Kippen

Annexe A

Méthode de Burg

Soient $e_k^f(n)$ et $e_k^b(n)$ les erreurs de prédiction à droite ('forward') et à gauche ('backward') pour un certain ordre k :

$$e_k^f(n) = x(n) - \sum_{i=1}^k a_i x(n-i) \quad e_k^b(n) = x(n) - \sum_{i=1}^k a_i x(n+k+i)$$

La méthode de Burg permet de calculer les coefficients du filtre de prédiction linéaire, en minimisant la moyenne de la puissance de l'erreur de prédiction à droite et à gauche sur un support fini, de manière récursive. C'est à dire, pour obtenir les coefficients $a(k)$ du filtre LP, on minimise :

$$\epsilon_k = \frac{1}{2}(\epsilon_k^f + \epsilon_k^b)$$

ou

$$\epsilon_k^f = \frac{1}{(N-k)} \sum_{n=k}^{N-1} |e_k^f(n)|^2 \quad \epsilon_k^b = \frac{1}{(N-k)} \sum_{n=0}^{N-1-k} |e_k^b(n)|^2$$
$$a_k(i) = \begin{cases} a_{k-1}(i) + r_k a_{k-1}(k-i) & \text{pour } i = 1, 2, \dots, k-1 \\ r_k & \text{pour } i = k \end{cases}$$

On appelle r_k le coefficient de réflexion. Les équations précédente se combinent pour donner une expression récursive de l'erreur à droite et à gauche.

$$e_k^f(n) = e_{k-1}^f(n) + r_k e_{k-1}^b(n-1) \quad e_k^b(n) = e_{k-1}^b(n-1) + r_k e_{k-1}^f(n)$$

Pour trouver r_k , on différencie la k^{th} puissance de l'erreur de prédiction par rapport à r_k , et en forçant la dérivée à zéro, on obtient :

$$r_k = \frac{-2 \sum_{n=k}^{N-1} e_{k-1}^f(n) e_{k-1}^b(n-1)}{\sum_{n=k}^{N-1} |e_{k-1}^f(n)|^2 + |e_{k-1}^b(n-1)|^2}$$

Annexe B

Extraits du code Matlab

Nota : Les extraits du code présentés ci-après utilisent souvent des variables définies globalement dans l'environnement de Matlab par d'autres scripts, et afin d'alléger (sic!) l'écriture, tout n'a pas été reprécisé. De plus, les constantes modifications apportées au cours du stage font que le style n'est pas des plus propre, et ne facilite pas forcément la lecture, malgré les commentaires. . . Le portage en C de ces fonctions, prévu dans un futur proche rendra certainement plus de clarté.

B.1 Extraction des paramètres de la LPC

Ce programme permet de trouver l'ordre du filtre LPC, ainsi que la taille du modèle les plus satisfaisants en calculant pour une performance choisie U l'erreur quadratique de la prédiction sur 50 échantillons futurs, pour différentes valeurs de ces paramètres.

```
function LPC_extract_param(U)
%
% PLOT FOR VARIOUS ORDER, MODEL_LENGTH, AND PREDICTION SIZE
% THE PREDICTION ERROR POWER ON TIME SIGNAL 'U'

ordre = [2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,25,30];
model_length = [4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,...
                22,23,24,25,26,27,28,29,30,35,40,50];
% SIZE OF ESTIMATED DATA ON WHICH WE WILL COMPUTE ERROR
Nb_Pred = 50;
so = max(size(ordre))
sm = max(size(model_length))
pred_quad_err_mtx = zeros(sm,so);
% TEST SIGNAL LENGTH SHOULD BE BIGGER THAN MODEL LENGTH
M = max(model_length)

for mod_len_idx = 1 : sm
    % COMPUTE ONLY WHEN ORDER IS LESS THAN MODEL_LENGTH
    N_ordre = find(ordre < model_length(mod_len_idx));
```

```

for ordre_idx = N_ordre
    pred_quad_err = 0;

    for pos_idx = (M+1) : (length(U)-M-Nb_Pred)
        % TAKE A WINDIW AND SUBSTRACT ITS MEAN
        x = U(pos_idx:pos_idx + M);
        x = x - mean(x);
        model = x(M+1-model_length(mod_len_idx):M);
        % COMPUTE LPC COEFFFS WITH BURG METHOD
        a = arburg(model,ordre(ordre_idx));
        % PREVENT FROM NaN VALUES OF BURG COEFS IN CASE OF CONSTANT SIGNAL
        if length(find(isnan(a))) > 0
            model = model + eps*10*rand(length(model),1);
            a=arburg(model,ordre(ordre_idx));
        end
        % COMPUTE PREDICTION
        y = IIRextrapolate(x,a,Nb_Pred);
        % COMPUTE QUADRATIC PREDICTION ERROR
        pred_quad_err = pred_quad_err...
            + sum(abs(U(pos_idx+M+1 : pos_idx+M+Nb_Pred)
                - (y(1:Nb_Pred) + m)));
    waitbar(pos_idx/length(U) , hwait);
    end
    % STORE THE ERROR FOR THESE ORDER AND MODEL_LENGTH
    pred_quad_err_mtx(mod_len_idx, ordre_idx) = pred_quad_err;
    if ishandle(hwait), close(hwait), end
end
end
% FOR DISPLAY PURPOSE: SET IMPOSSIBLE (ORDER Vs MODEL_LENGTH) AT MAX ERROR
pred_quad_err_mtx(find(pred_quad_err_mtx==0)) = max(max(pred_quad_err_mtx));
% 3D PLOT
surf(ordre, model_length, pred_quad_err_mtx)
xlabel('Ordre du filtre');
ylabel('Taille du modèle LPC');
zlabel('Erreur quadratique de prédiction');

```

B.2 Correction des données

Cette fonction prenant les valeurs de position et d'angle issues des capteurs, corrige les données en détectant les dépassement du plan (oYZ), les sauts des valeurs d'angle, les données bloqués, et les données perturbées.

```

TRUE=1;FALSE=0;ON=1;OFF=0;
X=1;Y=2;Z=3;
MIN_TIME_BETWEEN_BUMPS = 7; % minimum time (in samples) between 2 consecutive strikes;
[npoint, ndim, nbird] = size(pos_ar)

```

```

A_pred_ar = zeros(npoint,nbird);           % array with amount of prediction
bumps_ar = zeros(npoint, nbird);           % value to check bump detection
corrected_speed_ar = zeros(npoint, ndim*2+1,nbird); % corrected data's speed

% VARIABLES RELATIVE TO DATA CORRECTION
mem = 2 * model_length; % memory size we keep estimated and raw values in;
                        % here we keep twice as many values as necessary to
                        % build the model
max_count_crash = 15; % Max number of consecutive estimated values
max_count_agg = 5;   % Max number of consecutive agglutinated points
accel_tan_max = 0.025; % max possible value of acceleration
                  % (limited by percussionist force)
accel_tan_min = -0.025; % max possible value of deceleration
                  % (limited by percussionist force)
d_accel_tan_max = 0.05; % max possible value of acceleration derivative
                  % (limited by muscles bending time)
speed_max = 0.10;      % max value of speed (limited by NRJ=force*distance and
                  % sensor position on stick)
max_norm_corrected_accel = 0.04; %max value of norm of acceleration
                  % (used to come slowly back to raw values)
max_norm_speed = 0.8; % max value of norm of acceleration
                  % (used to come slowly back to raw values)
coef_dist = 2;        % coefficient controlling how fast we get back to real values
                  % the more the coef, the slowest the return to real values
                  % coef_dist=2 is not bad

% VARIABLES RELATIVE TO IMPACT DETECTION
accel_interval = 20; % max time interval (samples) for acceleration
deccel_interval = 10; % max time interval (samples) for deceleration
speed_interval = 10; % max time interval (samples) for speed above threshold

for ibird = 1:nbird
    % COUNTING HOW MANY CRASHED POINTS HAVE BEEN FOUND IN A ROW
    count_agg = 0; count_crashdata = 0;
    % INITIALIZE THRESHOLDS
    speed_min = 0.01; pic_accel = 0; pic_speed = 0; pic_deccel = 0;
    last_impact_pos = 0; raw_accel_tan = 0; last_accel_tan = 0; A_pred = 0;
    % MAXIMUM SPEED REACHED DURING STRIKE TRIGGER
    max_speed = 0;
    % INITIALIZE ARRAYS
    arburg_coeffs = zeros(ndim*2, ordre + 1);
    % CONCATENATE TO TREAT POSITION AND ANGLE TOGETHER
    data = [pos_ar(:,X:Z,ibird), ang_ar(:,X:Z,ibird)];
    % RAW VALUE OF THE INCOMING SIGNAL [ERRONOUS VALUES INCLUDED]
    raw_data_mem = zeros(mem, ndim*2);
    raw_data_mem(mem-model_length+1:mem,:) = data(1:model_length, :);
    % CORRECTED DATA = RAW VALUE + SOME ESTIMATED VALUES

```

```

corrected_data = zeros(npoint, ndim*2);
corrected_data(1:model_length, :) = data(1:model_length, :);
corrected_speed_ar(1:model_length, 1:ndim*2, ibird) = ...
    [zeros(1, ndim*2); diff(corrected_data(1:model_length, :))];
estimated_data = zeros(1, ndim*2);
corrected_accel_tan = 0;

for idx = (model_length + 1) : npoint
    % UPDATE LAST TURN MEMORY VALUES
    last_raw_accel_tan = raw_accel_tan;
    last_corrected_accel_tan = corrected_accel_tan;
    raw_data_mem = [raw_data_mem(2:mem, 1:ndim*2); data(idx, 1:ndim*2)];
    % HANDLE AXES INVERSION (WHEN SENSOR CROSS OYZ PLANE)
    % BY COMPARING INCOMING RAW DATA WITH THEIR PREDICTION
    if sum(abs(corrected_data(idx, X:Z) + raw_data_mem(mem, X:Z))) < 0.1 ...
        & sum(abs(corrected_data(idx, X:Z) + raw_data_mem(mem, X:Z))) > 0.1
        raw_data_mem(mem, X:Z) = - raw_data_mem(mem, X:Z);
        disp(['Axes inversion at position: ', num2str(idx)]);
    end
    % ANGLE PHASE UNROLL: LOOK FOR BIG JUMPS AND ADD/SUBSTRACT MODULO 2
    for iang = 4:6
        if (corrected_data(idx, iang) - raw_data_mem(mem, iang)) > 0.9 ...
            & corrected_data(idx, iang) * raw_data_mem(mem, iang) < 0
            raw_data_mem(mem, iang) = raw_data_mem(mem, iang) ...
                - 2*sign(raw_data_mem(mem, iang));
        end
    end
    raw_speed = raw_data_mem(mem, 1:ndim*2) - raw_data_mem(mem-1, 1:ndim*2);
    % COMPUTE TANGENTIAL ACCELERATION FROM RAW VALUES
    if norm(raw_speed(X:Z)) ~= 0
        raw_dir = raw_speed(X:Z)/norm(raw_speed(X:Z));
    else
        raw_dir = zeros(1, ndim);
    end
    raw_accel = raw_data_mem(mem, X:Z) ...
        - 2*raw_data_mem(mem-1, X:Z) ...
        + raw_data_mem(mem-2, X:Z);
    raw_accel_tan = raw_accel * raw_dir';
    % COMPUTE TANGENTIAL ACCELERATION FROM CORRECTED VALUES
    if corrected_speed_ar(idx, 7, ibird) ~= 0
        corrected_dir = corrected_speed_ar(idx, X:Z, ibird) ...
            / corrected_speed_ar(idx, 7, ibird);
    else
        corrected_dir = zeros(1, ndim);
    end
    corrected_accel = raw_data_mem(mem, X:Z) ...

```

```

        - 2*corrected_data(idx-1,X:Z) ...
        + corrected_data(idx-2,X:Z);
    corrected_accel_tan = corrected_accel * corrected_dir';
% COMPUTE DISTANCE FROM RAW VALUE TO USE IT LATER FOR A SMOOTHER TRANSITION
    dist_from_truth = sum(abs(corrected_data(idx, X:Z) - raw_data_mem(mem, X:Z)));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% BEGIN CORRECTION ANALYSIS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% BLOCKED DATA: NEW DATA IS SAME AS LAST CORRECTED_DATA ON ALL AXIS
    if (raw_data_mem(mem, :) == raw_data_mem(mem - 1, :))
        if count_agg < max_count_agg
            count_crashdata = 0;
            count_agg = count_agg + 1;
            A_pred = 1;
        else
            count_crashdata = 0;
            count_agg = 0;
            dist_from_truth = sum(abs(corrected_data(idx, X:Z) ...
                                     - raw_data_mem(mem, X:Z)))
            A_pred = min(max(0,A_pred-0.01), coef_dist*dist_from_truth);
        end
    else % RAW TANGENTIAL ACCEL FITS PHYSICAL BOUNDS
        if ((raw_accel_tan > accel_tan_min)...
            & (raw_accel_tan < accel_tan_max)...
            & (abs((raw_accel_tan-last_raw_accel_tan)) < d_accel_tan_max))...
            | ... % AVOID DRIFTING TOO LONG
            (count_crashdata>max_count_crash) ...
            | (count_agg>max_count_agg)

            if (count_crashdata > 0) | (count_agg > 0)
                A_pred = min(max(0,A_pred-0.01), coef_dist*dist_from_truth);
                count_crashdata = 0;
            else
                A_pred = 0;
                count_agg = 0;
            end
        else % CRASHED DATA !
            count_crashdata = count_crashdata+1;
            count_agg = 0;
            A_pred = min(1, coef_dist*dist_from_truth);
        end
    end
    A_pred_ar(idx, ibird) = A_pred;
    corrected_speed_ar(idx,1:ndim*2,ibird) = A_pred*estimated_data(1:ndim*2) ...
        + (1-A_pred)*raw_speed(1:ndim*2);

```



```

% LIMIT IT TO A PHYSICALLY POSSIBLE VALUE
corrected_speed_ar(idx, 7, ibird) = norm(corrected_speed_ar(idx,X:Z,ibird));
if corrected_speed_ar(idx, 7, ibird) ~= 0
    corrected_speed_ar(idx,1:ndim*2,ibird) = ...
        corrected_speed_ar(idx,1:ndim*2,ibird)...
        * min(corrected_speed_ar(idx, 7, ibird), max_norm_speed)...
        / corrected_speed_ar(idx, 7, ibird);
end
% RECOMPUTE CORRECTED_ACCEL TO GET A PHYSICALLY POSSIBLE VALUE LIMITED
corrected_accel = corrected_speed_ar(idx, X:Z,ibird) ...
    - corrected_speed_ar(idx-1,X:Z,ibird);
% LIMIT IT !
norm_corrected_accel = norm(corrected_accel(X:Z));
if norm_corrected_accel ~= 0
    corrected_accel= corrected_accel ...
        * min(norm_corrected_accel,max_norm_corrected_accel)...
        / norm_corrected_accel;
end
% RECOMPUTE SPEED ACCORDING TO LIMITED ACCEL_TAN
corrected_speed_ar(idx,X:Z,ibird)= corrected_speed_ar(idx-1,X:Z,ibird) ...
    + corrected_accel;
% COMPUTE POSITION FROM CORRECTED SPEED
corrected_data(idx,1:ndim*2) = A_pred *(corrected_data(idx-1,1:ndim*2)...
    + corrected_speed_ar(idx, 1:ndim*2, ibird))...
    + (1-A_pred)*raw_data_mem(mem,1:ndim*2);
% LIMIT POSITION TO RANGE [-1,1 FOR EACH AXIS
% NB: DO NOT DO IT FOR ANGLE BECAUSE OF PHASE UNROLL
corrected_data(idx,X:Z) = min(max(corrected_data(idx,X:Z),-1), 1);
% COMPUTE SPEED PREDICTION FOR NEXT TURN
for axe = 1 : ndim*2
    temp = arburg(corrected_speed_ar((idx-model_length) : idx, axe, ibird), ordre);
    % AVOID NaN's IN CASE OF CONSTANT SIGNAL BY TAKING PREVIOUS COEFS
    if sum(isnan(temp)) == 0
        arburg_coeffs(axe, :) = temp; %else: keep previous ones
    end
    % ... COMPUTE NBPRED PREDICTED VALUES (HERE: 1)
    estimated_data(axe)= IIRextrapolate(corrected_speed_ar((idx-model_length):idx,...
        axe, ibird), arburg_coeffs(axe, :), 1);
end
% RECOMPUTE CORRECTED_ACCEL_TAN TO GET A CORRECT
% LAST_CORRECTED_ACCEL_TAN IN NEXT TURN
norm_corrected_speed = norm(corrected_speed_ar(idx,X:Z,ibird));
if norm_corrected_speed ~= 0
    corrected_dir = corrected_speed_ar(idx,X:Z,ibird)/norm_corrected_speed;
else
    corrected_dir = zeros(1, ndim);
end
end

```

```
corrected_accel = corrected_speed_ar(idx, X:Z, ibird) ...
    - corrected_speed_ar(idx-1,X:Z, ibird);
corrected_accel_tan = corrected_accel * corrected_dir';
```

B.3 Détection des impacts

Cette fonction appelée après la phase de correction des données détecte les impacts, en repérant un pic d'accélération suivi d'un pic de vitesse, suivi d'une décélération et d'un changement de direction

```
% TRY TO PREDICT NEXT POSITION VALUE SO THAT WE CAN DETECT BUMPS
% JUST BEFORE A CHANGE IN DIRECTION
corrected_speed_ar(idx+1, 1:ndim*2, ibird) = estimated_data(1:ndim*2);
corrected_data(idx+1, 1:ndim*2) = corrected_data(idx,1:ndim*2) ...
    + corrected_speed_ar(idx+1, 1:ndim*2, ibird);

if (2 <= pic_accel) & (pic_accel < accel_interval)
    if (2 <= pic_speed) & (pic_speed < speed_interval)
        if (1 <= pic_deccel) & (pic_deccel < deccel_interval)
            if ((corrected_speed_ar(idx+1,X:Z,ibird) ...
                * corrected_speed_ar(idx-1,X:Z,ibird)' < 0.0001)...
                & (idx - last_impact_pos) > MIN_TIME_BETWEEN BUMPS)
                % IMPACT DETECTED!
                bumps_ar(idx, ibird) = 10;
                last_impact_pos = idx;
                pic_accel = 0; pic_deccel = 0; pic_speed = 0; max_speed = 0;
            else
                pic_deccel = pic_deccel + 1;
                bumps_ar(idx, ibird) = 3;
            end
        else
            if (corrected_accel_tan < -0.0015) & (pic_deccel < deccel_interval)
                % TRY TO ANTICIPATE WITH PREDICTED VALUE
                if ((corrected_speed_ar(idx+1,X:Z,ibird) ...
                    * corrected_speed_ar(idx-1,X:Z,ibird)' < 0.0001)...
                    & (idx - last_impact_pos) > MIN_TIME_BETWEEN BUMPS)
                    % IMPACT DETECTED!
                    bumps_ar(idx, ibird) = 10;
                    last_impact_pos = idx;
                    pic_accel = 0; pic_deccel = 0; pic_speed = 0; max_speed = 0;
                else
                    pic_deccel = pic_deccel + 1;
                    bumps_ar(idx, ibird) = 3;
                end
            else
                max_speed = max(max_speed, norm_corrected_speed);
```

```

        pic_deccel = 0;
        pic_speed = pic_speed + 1;
        bumps_ar(idx, ibird) = 2;
    end
end
else
    if (norm_corrected_speed > 0.015) & (pic_speed < speed_interval)
        pic_speed = pic_speed + 1;
        bumps_ar(idx, ibird) = 2;
    elseif (corrected_accel_tan > 0.001) & (pic_accel < accel_interval)
        pic_speed = 0;
        pic_accel = pic_accel + 1;
        bumps_ar(idx, ibird) = 1;
    else % was just a movement: untrigger everything
        pic_speed = 0;
        pic_accel = 0;
        bumps_ar(idx, ibird) = 0;
    end
end
else
    if (corrected_accel_tan > 0.0015) & (pic_accel < accel_interval)
        pic_accel = pic_accel + 1;
        bumps_ar(idx, ibird) = 1;
    else % rest state
        pic_accel = 0; pic_deccel = 0; pic_speed = 0; max_speed = 0;
        bumps_ar(idx, ibird) = 0;
    end
end
end
end
end
end

```

B.4 Etude statistique

Le but est ici d'extraire la moyenne et la variance des paramètres de vitesse et d'accélération autour des points d'impacts, et par type de coup. Les types de coups ont préalablement été indiqués à la main directement sur les séquences de mesure. NB. : La variable `filenames` contient les noms de toutes les performances enregistrées et segmentées manuellement.

```

% STRIKES CODING
down=1;up=2;piston=3;etouffe= 4;
s.strike_names = {'Down','Up','Piston','Etouffe'};
s.nstrike = length(s.strike_names);
% PARAMETERS CODING
s.param_names = {'Speed', 'Accel'};
s.nparam = length(s.param_names);
s.stat_param_names = {'Mean', 'Std dev'};

```

```

s.nstat_param = length(s.stat_param_names);
% ANALYSIS WINDOW AROUND IMPACT
win = 20;
% KEEP GLOBAL STATISTICS IN strike_stat(:, :, :, :, :, nfile+1)
strikes_stat = zeros(2*win+1, s.nstrike, s.nparam, s.nstat_param);
% THE AVAILABLE PERFORMANCES ARE IN filenames
nfile = length(filenames);

for istrike = 1: s.nstrike
    % ARRAY IN WHICH WE STORE SPEED AND ACCEL WINDOW AROUND STRIKE
    strike_param = zeros(2*win+1, 1, s.nparam);
    for ifile = 1 : nfile
        % LOADING PERFORMANCE DATA
        [pos_ar, ang_ar, s] = loaddata(filename, s);
        [bumps_ar, A_pred_ar, strikes_ar, s] = loadmetadata(filename, s);
        [npoint, ndim, nbird] = size(pos_ar);
        % COMPUTE SPEED AND ACCEL PARAMETERS
        [pos_speed_ar, ang_speed_ar, pos_accel_ar, ang_accel_ar] = ...
            compute_param(pos_ar, ang_ar, s);
        for ibird = 1:nbird
            % FIND TIME POSITION OF THE STRIKES OF THE TYPE BEING ANALYSED
            idx_strikes = find(strikes_ar(:, ibird) == istrike);
            if length(idx_strikes) ~= 0
                for idx_str = idx_strikes'
                    % STORE SPEED AND ACCEL PARAMETER ON A WINDOW AROUND IMPACT
                    strike_param(:, end+1, 1) = pos_speed_ar(idx_str-win:idx_str+win, ...
                                                                4, ibird);
                    strike_param(:, end, 2) = pos_accel_ar(idx_str-win:idx_str+win, ...
                                                            4, ibird);
                end
            end
        end
    end
end
% COMPUTE STATISTICS ON ARRAYS
idx_strikes = find(strikes_ar(:, ibird) == istrike);
for idx_win = 1 : 2*win+1
    strike_stat(idx_win, istrike, 1, 1) = mean(strike_param(idx_win, :, 1));
    strike_stat(idx_win, istrike, 1, 2) = std(strike_param(idx_win, :, 1));
    strike_stat(idx_win, istrike, 2, 1) = mean(strike_param(idx_win, :, 2));
    strike_stat(idx_win, istrike, 2, 2) = std(strike_param(idx_win, :, 2));
    waitbar(idx_win / 2*win+1, hwait);
end
% PLOT STATISTICS RESULTS
draw_stat2(strike_stat, 'titre', s);
end

```

B.5 Reconnaissance des coups

Cette fonction calcule la similarité entre le paramètre de vitesse, sur une fenêtre temporelle de 40 échantillons.

```
% STRIKE CODING
down=1; up=2; piston=3; etouffe= 4;
s.strike_names = {'Down','Up','Piston'};%,'Etouffe'};
s.nstrike = length(s.strike_names);

s.param_names = {'Speed', 'Accel'};
s.nparam = length(s.param_names);
s.stat_param_names = {'Mean', 'Std dev'};
s.nstat_param = length(s.stat_param_names);
% LOAD PERFORMANCE DATA, METADATA, AND STATISTICS FOR STRIKES
[pos_ar, ang_ar,s] = loaddata(filename,s);
[npoint, ndim, nbird]=size(pos_ar);
[bumps_ar, A_pred_ar, strikes_ar,s] = loadmetadata(filename,s);
load strike_stat;
[win, s.nstrike, s.nparam, s.nstat_param] = size(strike_stat);
% SIMILARITE
simil_coup = zeros(npoint, s.nstrike);
% COMPUTE SPEED AND ACCEL PARAMETERS
[pos_speed_ar, ang_speed_ar, ...
 pos_accel_ar, ang_accel_ar, ...
 pos_correl_ar, ang_correl_ar] = compute_param(pos_ar, ang_ar,s);
% NORMALIZE STRIKE STAT ENERGY
strike_stat_norm = zeros(win, s.nstrike, s.nparam);
for iparam = 1:s.nparam
    for istrike = 1:s.nstrike
        strike_stat_norm(:,istrike,iparam) = strike_stat(:,istrike,iparam,1) ...
            / sqrt(sum(power(strike_stat(:,istrike,iparam,1), 2)));
    end
end
strike_norm = zeros(win, s.nparam);

for idx = (0.5*(win-1)+1) : (npoint-(0.5*(win-1))-1)
    % NORMALIZE INCOMING DATA ENERGY
    E = sqrt(sum(power(pos_speed_ar(\textit{window}))))
    if E > 0
        strike_norm(1:win,1) = pos_speed_ar(\textit{window}) / E;
    else
        strike_norm(1:win,1) = zeros(win, 1);
    end
    E = sqrt(sum(power(pos_accel_ar(\textit{window}))))
    if E > 0
        strike_norm(1:win,2) = pos_accel_ar(\textit{window}) / E;
```

```

else
    strike_norm(1:win,2) = zeros(win, 1);
end

% COMPUTE SIMILARITY AS FIRST CORRELATION COEF
simil_coup(idx, istrike) = sum(strike_norm(1:win,1).* ...
                                strike_stat_norm(1:win,istrike,1,1));
end
% PLOT SIMILARITY WITH NORMALIZED PARAMETERS
figure
subplot(3,1,1)
a=plot(simil_coup(:, 1),'marker','square','markersize',4,...
        'markeredgecolor', 'b', ...
        'linestyle','-', 'color', 'b','linewidth',1);
hold on
b=plot(simil_coup(:, 2),'marker','*','markersize',4,...
        'markeredgecolor', 'y', ...
        'linestyle','-', 'color', 'y','linewidth',1)
hold on
c=plot(simil_coup(:, 3),'marker','v','markersize',4,...
        'markeredgecolor', 'r', ...
        'linestyle','-', 'color', 'r', 'linewidth',1)

legend([a,b,c], strcat('Similarity for strike Down'), ...
        strcat('Similarity for strike Up'),...
        strcat('Similarity for strike Piston'));
xlabel('time samples')
ylabel('Similarity of normalized parameters')
hold off

```

Bibliographie

- [Abb82] C. Abbott. Remembering performance gestures. In *Proceedings of the ICMC*, pages 188–193, San Francisco, U.S.A., 1982.
- [ACH04] Thierry Coduys Arshia Cont and Cyrille Henry. Augmented mapping : Towards an intelligent user-defined gesture mapping. In *Sound and Music Conference*, 2004.
- [Azu95] Ronald Tadao Azuma. *Predictive tracking for augmented reality*. PhD thesis, University of North Carolina - Chapel Hill, 1995.
- [BBS01] Nicolas Sceaux Bertrand Bocande and Johann Sebenne. Modélisation d'un percussionniste - rapport de projet de fin d'étude, juin 2001.
- [Cad88] Claude Cadoz. Instrumental gesture and music composition. *Proceedings of the ICMC*, 1988.
- [CH03] Thierry Coduys and Cyrille Henry. Nouveaux matériels de captation haute précision. In *Journées d'Informatique Musicale*, France, 2003.
- [DCH04] Myriam DeSainte-Catherine and Christophe Havel. *Modeling an Air Percussion for Composition and Performance*. *Proceedings of the NIME conference*, 2004.
- [Del88] François Delalande. *La gestique de Glenn Gould in Glenn Gould Pluriel*. Louise Courteau éditrice,inc, 1988.
- [Del01] Nathan Delson. Bird i/o in matlab. , 2001.
- [DSOB03] François Déchelle, Norbert Schnell, Nicolas Orio, and Riccardo Borghesi. Free software at ircam. <http://www.freesoftwares.ircam.fr/>, 2003.
- [Efr72] David Efron. *Gesture, race and Culture*. The Hague, Mouton and Compagny, 1972.
- [FKZ00] Daniel Arfib Florian Keiler and Udo Zolzer. Efficient linear prédiction for digital audio effects. In *Proceedings of the DAFx conference*, Verona, It, 2000.
- [Flo78] Jean-Loup Florens. *Coupleur gestuel rétroactif pour la commande et le contrôle de sons de synthèse*. PhD thesis, Institut National Polytechnique de Grenoble, 1978.
- [GM00] Marie-Claire Gaudin and Eric Menu. Modélisation d'un percussionniste - rapport de projet de fin d'étude, 2000.
- [God97] Hervé Godefroy. Etude et application d'un système de commandes d'instruments virtuels de percussion, septembre 1997.
- [IKS01] Jirki Kaupinnen Ismo Kaupinnen and Pekka Saarinen. A method for long extrapolation of audio signals. *JAES vol. 49, No12*, pages 1167–1180, 2001.
- [KA03] Loïc Kessous and Daniel Arfib. Bimanuality in alternate musical instruments. In *Proceedings of the NIME conference*, Canada, 2003.

- [KB] Jim Kippen and Bernard Bel. The identification and modelling of a percussion ‘language’, and the emergence of musical concepts in a machine-learning experimental set-up. *Computers and the Humanities*.
- [Kin00] Volodymyr V. Kindratenko. A survey of electromagnetic position tracker calibration techniques. In *Virtual Reality :Research,Development, and Applications*, volume 5, pages 169–182, 2000.
- [Las77] O. Laske. Toward a theory of interfaces for computer music systems. *Computer Music Journal*, 1977.
- [Leb03] Sébastien Lebreton. Percussionniste virtuel - rapport de projet de fin d’étude, 2003.
- [MBD01] F. Della Maestra T. Brouillet M. Chapis M. Boulanger, K. Dang and JC. Deledalle. Modélisation d’un percussionniste virtuel - rapport de projet de fin d’étude, mai 2001.
- [MLR03] M. Raspaud M. Lagrange, S. Marchand and J-B. Rault. Enhanced partial tracking using linear prediction. In *Proceedings of the DAFx conference*, London, UK, 2003.
- [Mul04] Rémy Muller. Human motion following system using hmm, 2004.
- [MWR98] N. Schnell M. Wanderley and J. Roven. Escher - modeling and performing ‘composed instruments’ in real-time. In *Proc. IEEE International Conference on Systems, Man and Cybernetics (SMC’98)*, pages 188–193, 1998.
- [Nak00] Teresa Marrin Nakra. *Inside the conductor’s Jacket : Analysis, Interpretation and Musical Synthesis of Expressive Gesture*. PhD thesis, Media laboratory, 2000.
- [Pai04] Dr Garth Paine. Gesture and musical interaction : interactive engagement through dynamic morphology. In *Proceedings of the NIME conference*, pages 80–86, 2004.
- [PZ03] Miller Puckette and David Zicarelli. Max/msp reference. <http://www.cycling74.com/>, 2003.
- [RBS89] Max Matthews Robert Boie and A. Schloss. The radio drum as a synthesizer controller. In *Proceedings of the ICMC*, pages 42–45, San Francisco, U.S.A., 1989.
- [SD02] W.A Schloss and Peter Driessen. New algorithms and technology for analysing gestural data. 2002.
- [SP02] Karim El Sawah and Christelle Pernon. Modélisation d’un percussionniste - rapport de projet de fin d’étude, juin 2002.
- [TrpOA93] Collectif d’auteurs Textes réunis par Odette Aslan. *Le corps en jeu*. C.N.R.S. Editions, 1993.
- [Van88] Christian Vander. *Un homme, une batterie*. Seventh-VHS, 1988.
- [Ver04] Charles Verron. Captation gestuelle pour une percussion augmentée, 2004.
- [Wan01] Marcello Wanderley. *Interaction musicien-instrument : application au contrôle de la synthèse sonore*. PhD thesis, Université Pierre et Marie Curie - Paris VI, 2001.
- [WB00] Marcello Wanderley and Marc Battier. *Trends in gestural control of music*. IRCAM - Centre Georges Pompidou, 2000.
- [WF97] M. Wright and A. Freed. Open sound control : A new protocol for communicating with sound synthesizers. In *Proceedings of the ICMC*, pages 101–104, Thessaloniki, 1997.
- [Zil04] Aymeric Zils. *Extraction automatique de descripteurs par traitement du signal et exploitation dans le cadre d’applications musicales*. PhD thesis, Université Paris 6, 2004.