Université Pierre et Marie Curie
Mémoire de Stage de Master $2^{eme}$ année
Sciences de l'ingénieur mention ATIAM
2006-2007

Anthony Sypniewski

# Suivi du geste pour l'interaction musicale

Mars - Juin 2007

Laboratoire d'accueil : IRCAM - Equipe Interaction Musicale Temps Réel
Responsable : Frédéric Bevilacqua

# Table des matières

# Résumé

Le développement du violon augmenté au sein de l'équipe Interaction Musicale Temps Réel de l'IRCAM présente un fort potentiel créatif pour les compositeurs. Les informations délivrées par les différents capteurs placés sur l'archet du violon recèlent une grande quantité d'information sur l'interprétation du musicien. Mais cette information n'est pas directement accessible. C'est donc parallèlement à ce projet que le développement d'un outil de suivi et de reconnaissance du geste est mis au point au sein de la même équipe afin de pouvoir accéder à cette information. Plus précisément, la méthode adoptée, utilisant les Modèles de Markov Cachés, a pour but de pouvoir comparer automatiquement deux interprétations d'une même phrase musicale. Afin d'améliorer le modèle utilisé jusque là, nous en avons modifiés certains aspects en prenant en compte les variations d'interprétations du musicien. Les résultats que nous avons obtenus, que ce soit sur des signaux de synthèse ou des signaux provenant du violon augmenté, montrent une amélioration dans le suivi en temps-réel.

# Chapitre 1

# Introduction

## 1.1 Le violon augmenté

Le projet de violon augmenté, développé à l'IRCAM au sein de l'équipe Interaction Musicale Temps Réel (IMTR), s'inscrit dans un travail de recherche sur les nouvelles interfaces pour la création et l'interprétation musicale. Depuis l'apparition du Theremin jusqu'aux plus récents Méta-Instrument[1] et autres Reactable[2], cet axe de recherche se concentre essentiellement sur l'extension des possibilités de contrôle à l'aide de l'électronique et du traitement numérique. On peut distinguer deux approches. La première consiste à mettre au point une interface totalement nouvelle ne faisant pas directement appel à de précédents travaux. Bien qu'attrayante, plusieurs inconvénients peuvent apparaître une fois la surprise de la nouveauté passée. En effet, lors de son apparition, une telle interface ne bénéficie pas de langage gestuel suffisamment élaboré et d'interprètes suffisamment expérimentés afin de dépasser le stade de la démonstration. La profusion constatée ces dernières années dans l'apparition de nouvelles interfaces de ce type est un frein pour l'installation d'un véritable vocabulaire musical adapté et spécifique pour la plupart d'entre elles. De fait, très peu peuvent espérer une durée de vie conséquente.

La deuxième approche a été principalement développée au sein du *Hyperinstrument Group* du *MIT Media Lab* par Tod Machover. Il s'agit d'augmenter électroniquement les capacités de la lutherie traditionnelle existante. Le développement du violon augmenté s'inscrit dans cette approche [1]. Elle consiste à placer des capteurs sur l'instrument afin de pouvoir obtenir des informations en temps réel sur les mouvements du violoniste et, après analyse, sur son jeu. Contrairement aux nouvelles interfaces, cette approche bénéficie d'un langage gestuel qui s'est affiné durant près de cinq siècles et d'une importante communauté d'interprète de haut niveau. Les informations que l'ont

[1] www.pucemuse.com
[2] mtg.upf.edu/reactable

peut obtenir à partir d'un tel dispositif sont donc potentiellement très riches et précises, et peuvent être utilisées à des fins artistiques[3] ou pédagogiques[4].

### 1.1.1 L'architecture materielle

Pour analyser les mouvements du violoniste, il existe principalement deux types de captations ayant chacune leurs avantages et inconvénients : la vidéo ou les capteurs embarqués. C'est cette deuxième solution qui a été retenue, plus souple à mettre en place (une fois le dispositif électronique mis au point) et correspondant à une réelle augmentation de l'instrument.

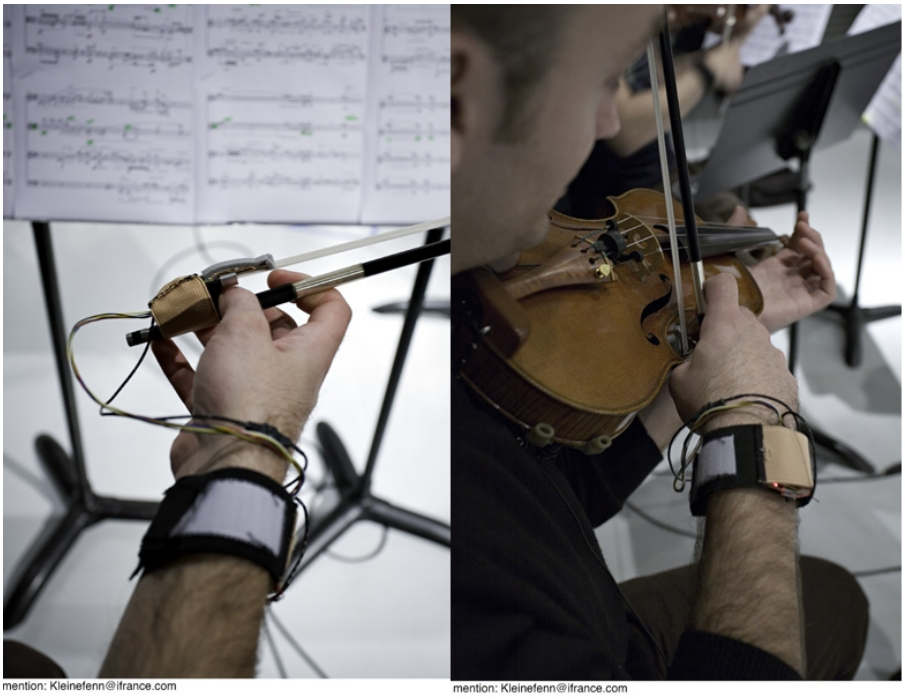


FIG. 1.1 – L'archet monté de capteurs d'accélération, de gyroscopes et d'un capteur de pression. Le bracelet autour du poignet du violoniste comporte l'alimentation et l'émetteur sans fil.

Une partie de la richesse d'un son produit par un violoniste provient de la pression et des mouvements donnés à son archet. C'est donc ces paramètres qu'il faut extraire. La hausse de l'archet se voit dotée d'un accéléromètre sur trois axes et d'un gyroscope sur deux axes (Fig. 1.1). Afin de mesurer la pression exercée par le musicien sur le violon, nous utilisons un dispositif développé par M. Demoucron au sein de l'équipe Acoustique instrumentale

[3] *Bogenlied* de Florence Baschet, pièce pour violon augmenté et électronique

[4] *I-Maestro*, environnement multimédia interactif pour l'éducation musicale (i-maestro.org)

de l'IRCAM [2]. Il s'agit de deux jauges de contrainte montées en opposition sur la mèche de l'archet du côté de la hausse. Lorsque la mèche se déforme, la tension aux bornes des jauges de contrainte change.

Les signaux ainsi récupérés sont numérisés sous forme d'un signal Open Sound Control (OSC) et transmis, sans fil, à une interface d'acquisition pour permettre le traitement. La batterie qui alimente le circuit conditionneur est fixé, à l'aide d'un bracelet, au poignet du violoniste. Le système, léger et sans fil, est très peu intrusif et ne modifie pas le jeu du violoniste.

### 1.1.2 Besoins et particularité de l'approche adoptée

L'information délivrée par les différents capteurs est de très bas niveau car elle nous renseigne uniquement sur les valeurs d'accélération ou de pression de l'archet. Il faut interpréter le signal afin d'obtenir des informations de haut niveau. En effet, une valeur d'accélération de l'archet est difficilement utilisable pour un musicien. Il faut alors traduire ces valeurs en langage musical. A partir des valeurs d'accélérations de l'archet, N. Rasamimanana a effectué une classification automatique des différents mode de jeu (*martelé*, *spiccato* et *détaché*) [3]. Ainsi, un musicien ou un compositeur peut directement utiliser l'information délivrée, dans un langage qu'il connaît.

D'autres informations peuvent être utiles, telles la vitesse d'exécution d'une phrase ou tout simplement, la reconnaissance d'une phrase exécutée. C'est le travail entreprit par l'équipe IMTR [4] [5]. Suivre un geste revient à aligner en temps réel la ou les courbes décrivant ce geste (accélération, position, pression ...) sur un geste enregistré au préalable. Reconnaître un geste revient à sélectionner l'exécution d'une phrase la plus semblable parmi un ensemble de phrases différentes enregistrés au préalable.

La reconnaissance d'une phrase nécessite un passage par une phase d'apprentissage des différentes phrases. Il y a plusieurs manière d'aborder cette phase. La démarche communément adoptée requiert un nombre d'exécutions important (de 20 à 100, selon la méthode de reconnaissance et le type de geste) rendant la phase d'apprentissage fastidieuse. L'approche présentement adoptée repose sur une seule exécution de chaque geste afin de réduire au minimum le temps d'apprentissage.

Parallèlement, les gestes effectués beaucoup plus lentement ou beaucoup plus rapidement que le geste de référence doivent également être reconnus et suivis.

## 1.2 Reconnaissance et suivi du geste : état de l'art

De nombreux travaux ont été entrepris sur la reconnaissance du geste. Au vu de sa nature non parfaitement reproductible, la modélisation du geste par un processus stochastique s'est imposé dans ce domaine. Provenant initialement des travaux effectués sur la reconnaissance de la parole [6], les Modèles

de Markov Cachés (MMC) sont utilisés dans la plupart des méthodes de reconnaissances formes ou de processus temporels (langage, écriture manuscrite, ...). Le modèle est assez souple pour pouvoir s'appliquer à de nombreux domaines mais nécessite d'être adapté selon le domaine considéré.

A partir d'une utilisation classique des MMC pour la reconnaissance du geste [7] [8], de nombreuses améliorations ont été proposées afin de prendre en compte la spécificité du phénomène observé. La plupart se concentrent sur une meilleure prise en compte des données d'apprentissage [9] [10] mais suppose toujours qu'il y a plusieurs réalisations du même geste.

A.D. Wilson propose une extension intéressante des MMC [11] [12] [13] afin de paramétriser le geste et donc de prendre en compte les déformations (linéaires ou non) qui peuvent apparaître dans l'exécution du geste lors de le reconnaissance et de pouvoir les mesurer. Mais la phase d'apprentissage reste fastidieuse : chaque geste servant à l'apprentissage doit être paramétrisé manuellement.

X. Ge propose également une alternative intéressante afin de trouver une forme donnée dans un flux d'information [14] [15], tout en prenant en compte de possible déformations dans le signal observé. Ce dernier est modélisé par une semi-chaîne de Markov [16] ce qui permet de mieux prendre en compte les exécutions de geste à des vitesses différentes mais ne bénéficie plus de la simplicité d'implémentation d'une chaîne de Markov classique.

D'autres pistes sont néanmoins proposées. Certains partent de l'algorithme de *Dynamic Time Warpping* (DTW) pour obtenir une mesure de similarité entre deux séquences [17] [18] ou pour effectuer un alignement [19]. La plupart de ces méthodes ne sont pas orientées temps-réel car l'algorithme de DTW ne peut s'effectuer qu'une fois le signal entièrement disponible. Récemment, des travaux ont été entrepris à l'IRCAM afin d'adapter cet algorithme à un alignement proche du temps réel (DTW à court-terme) [20].

# Chapitre 2

# Le suivi de geste

Pour un geste musical donné, quelque soit sont échelle temporelle (de la note à la phrase), il existe une multitude de traductions en terme de signaux numériques. Une traduction efficace préserve certaines caractéristiques communes pour une série de gestes identiques et permet de les discriminer devant d'autres gestes. Ainsi, les différents signaux récupérés depuis l'archet du violon augmenté nous permettent d'obtenir une bonne description des gestes effectués par le violoniste.

Les différences de geste constatés dans deux réalisations d'une même phrase musicale (que ce soit pour un même musicien ou pour deux musiciens différents) sont essentiellement provoquées par des différences d'interprétation. La finalité de la reconnaissance puis du suivi d'un geste est de pouvoir obtenir des informations sur ces différences. Pour ce faire, il nous faut, dans un premier temps, modéliser ces différences comme étant du "bruit gestuel" afin de ne garder que les caractéristiques discriminantes du geste en question (pour pouvoir le reconnaître puis le suivre). Une fois l'alignement effectué, l'information sur les différences d'intensité et de vitesse d'exécution peuvent être extraites.

Dans un premier temps, nous verrons comment appliquer le Modèle de Markov Caché pour reconnaître et suivre un geste. Ensuite, nous proposerons quelques améliorations sur un point précis du modèle afin de l'adapter à notre problématique.

## 2.1 Modèle de Markov Caché

La traduction du geste que nous avons à notre disposition se prête à une modélisation par un processus stochastique tel que le Modèle de Markov Caché. Nous verrons tout d'abord comment modéliser chaque geste constituant notre base de donnée gestuelle par une chaîne de Markov, puis la manière dont est traitée l'information sur le signal entrant (geste à reconnaître et à suivre).

### 2.1.1 Modèle de Markov du signal gestuel

Une chaîne de Markov est un processus stochastique décrivant un système comportant un nombre $N$ d'états, $S = \{S_1, S_2, S_3 \dots S_N\}$ (Fig. 2.1, où $N = 4$).

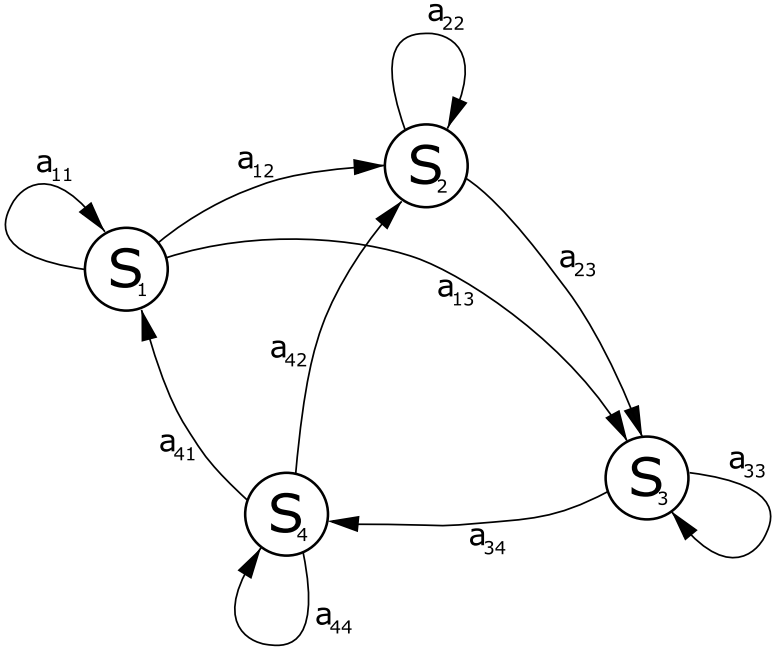


FIG. 2.1 – Chaîne de Markov à 4 états

A chaque instant $t$, $q_t$ représente l'état dans lequel nous nous trouvons. A l'instant $t + 1$, le système subit un changement aléatoire vers l'état $q_{t+1}$ en fonction d'une distribution des probabilités de passage entre les différents états de la chaîne. Cette distribution ne dépend que de l'état présent (chaîne de Markov d'ordre 1). La chaîne est dite homogène si ces probabilités ne dépendent pas du temps. La probabilité de transition de l'état $S_i$ vers l'état $S_j$ se note alors $a_{i,j}$. Nous obtenons la *matrice de transition* $A$ pour une chaîne de Markov à $N$ états :

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{bmatrix}$$

Nous avons

$$a_{ij} > 0$$

et

$$\sum_{j=1}^{N} a_{ij} = 1$$

Les signaux décrivant notre geste, échantillonnés à la période $T$, sont modélisés par une suite d'états représentant le déroulement temporel du geste. La période d'échantillonnage de la chaîne est $nT$ avec $n = 2, 3, \ldots$ (Fig. 2.2 pour $n = 2$). De plus, seuls sont permis les passages d'un état $S_i$ vers l'état suivant $S_{i+1}$ ou le même état $S_i$. Ainsi, nous en déduisons les probabilités de passages :

$$a_{i,j} = 0 \qquad si \qquad \begin{cases} j < i \\ j > i + 1 \end{cases}$$

$$a_{i,i} = 1 - \frac{1}{n} \qquad \text{et} \qquad a_{i,i+1} = \frac{1}{n}$$

Si la chaîne est circulaire (on autorise le passage de l'état $S_N$ à l'état $S_1$), nous obtenons la matrice de transition suivante pour $n = 2$ :

$$A = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 & \ldots & 0 \\ 0 & 0.5 & 0.5 & 0 & \ldots & 0 \\ 0 & 0 & 0.5 & 0.5 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 0.5 & 0.5 \\ 0.5 & 0 & 0 & \ldots & 0 & 0.5 \end{bmatrix}$$

Les valeurs d'accélérations et de pression récupérés des mouvements de notre archet sont notés $U = \{u_i\}$ avec $1 \leq i \leq N$ et $u_i$ un vecteur de dimension le nombre d'informations sur le mouvement de l'archet (noté $p$ ; nous ne prenons pas forcément en compte l'ensemble des signaux délivrés par l'archet augmenté). Modélisées par le processus stochastique, nous obtenons une chaîne sous-échantillonnée d'un facteur 2 par rapport au signal de référence (Fig 2.2).

La distribution initiale (au temps $t = 1$) de probabilité de présence parmi les différents états de notre chaîne est $\pi = \{\pi_i\}$, avec :

$$\pi_i = Prob(q_1 = S_i) \qquad 1 \leq i \leq N$$

Lorsque nous sommes certain de démarrer notre geste au niveau du premier état de la chaîne, nous avons :

$$\pi_i = \begin{cases} 1 & \text{si } i = 1 \\ 0 & \text{sinon} \end{cases}$$

Dans le cas opposé, si nous n'avons aucune information sur l'état de départ du geste (pour un geste cyclique par exemple), la probabilité de présence est uniformément répartie sur l'ensemble des états de la chaîne :

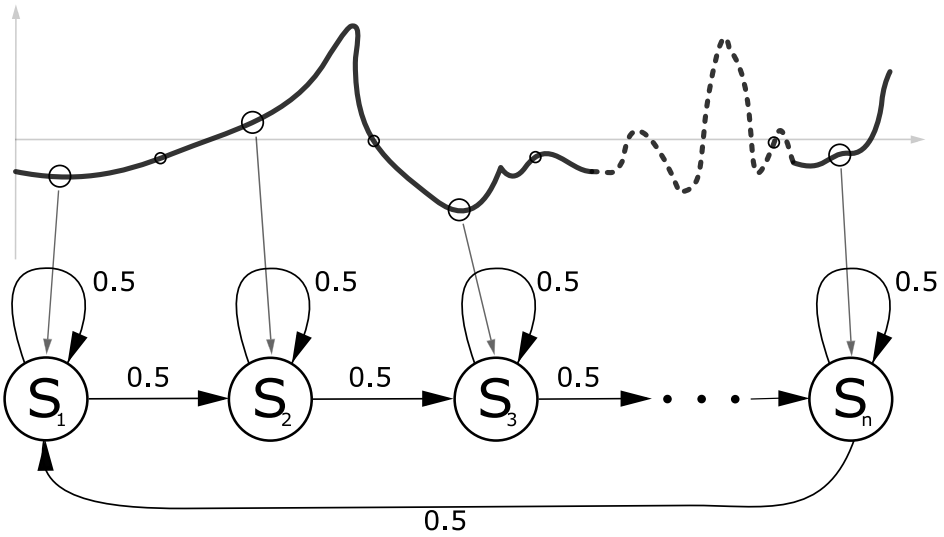


FIG. 2.2 – Chaîne de Markov circulaire du premier ordre à $N$ états, construite à partie de notre signal d'accélération.

$$\pi_i = \frac{1}{N} \qquad \forall i$$

La *matrice d'estimation* $E = \{e_1, e_2, \ldots, e_N\}$ qui nous donne la répartition des probabilités de présence sur la chaîne de Markov au temps $t$ est égale à $\pi$ au temps $t = 1$.

### 2.1.2 Emission

Bien entendu, lorsque nous observons le signal entrant, décrivant le geste à reconnaître et à suivre, nous n'observons pas directement l'état correspondant dans la chaîne modélisant ce geste, mais l'ensemble des valeurs d'accélérations et de pression de l'archet à chaque instant $t$, notée $V = \{v_1, v_2, \ldots, v_M\}$ ( $M$ étant le nombre d'échantillons du signal représentant le geste entrant). Les états sont *cachés*. Il nous faut donc introduire un deuxième processus stochastique entre l'observation $V$ et l'état de la chaîne $S$. Nous définissons alors la *matrice d'observation* $B = \{b_j(k)\}$ comme étant la probabilité d'observer la valeur courante $v_k$ sur l'état $S_j$ :

$$b_j(k) = Prob(\text{observer } v_k \text{ au temps } t \mid q_t = S_j) \qquad 1 \leq j \leq N,\ 1 \leq k \leq M$$

Nous considérons que les probabilités d'émission suivent une loi gaussienne :

$$b_j(k) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-dist(v_k, u_j)}{2\sigma^2}}$$

avec $dist(v_k, u_j)$ la distance Euclidienne entre les deux vecteurs $v_k$ et $u_j$ :

$$dist(v_k, u_j) = \sqrt{\sum_p (v_k - u_j)^2}$$

Nous obtenons, à chaque instant $t$, une distribution de probabilité d'émettre la valeur courante d'accélération (ou de pression) pour l'ensemble de la chaîne de Markov (Fig. 2.3).

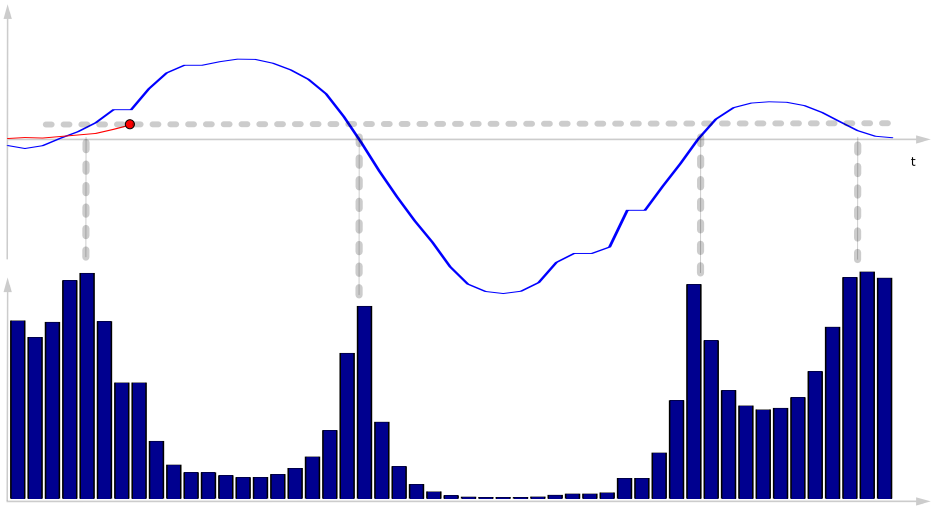


FIG. 2.3 – Le signal de référence (en bleu), le signal entrant (en rouge), ainsi que les probabilités d'émettre la valeur courante pour l'ensemble du signal de référence (histogramme). Ces probabilités sont maximales pour les valeurs d'accélérations du signal de référence égale à la valeur d'accélération entrante.

### 2.1.3 Déroulement de l'algorithme

Notre modèle est constitué, pour chaque geste de la base de donnée gestuelle, par :

- une chaîne de Markov à N états, $S = \{S_1, S_2, \ldots, S_N\}$, modélisant le signal du geste de référence
- une matrice de transition $A = \{a_{i,j}\}$ caractérisant les probabilités de passage entre les états $S_i$ et $S_j$ de la chaîne ($1 \leq i \leq N$ et $1 \leq j \leq N$)
- une matrice d'observation $B = \{b_j(k)\}$, pour chaque état de chaque chaîne de Markov ($1 \leq j \leq N$) ainsi que pour chaque valeur du signal entrant ($1 \leq k \leq M$, $M$ étant le nombre d'échantillons du signal représentant le geste entrant)
- une distribution initiale des probabilités de présence $\pi = \{\pi_1, \pi_2, \ldots, \pi_N\}$

– une matrice d'estimation $E = \{e_1, e_2, \ldots, e_N\}$ qui nous donne la répartition des probabilités de présence sur la chaîne de Markov

Le geste entrant est représenté par la distribution $V = \{v_1, v_2, \ldots v_M\}$.

Au temps $t = 1$, le choix de l'état initial $q_1 = S_i$ se fait en choisissant le maximum de la distribution $E$, soit le maximum distribution initiale $\pi$. Puis, pour chaque geste de notre base de donnée, tant que $t < M$ :

1. la matrice d'estimation $E$ est multipliée pas la matrice de transition $A$ : $E = E \times A$
2. la matrice d'estimation $E$ est multipliée, terme à terme, par la matrice d'observation $B$, obtenue à partir du signal entrant $v_t$ : $E = E.B$

Ainsi, à chaque instant $t$, l'état de la chaîne dans lequel nous avons le plus de chance de se trouver correspond au maximum de la matrice d'estimation $E$ (Fig. 2.4).

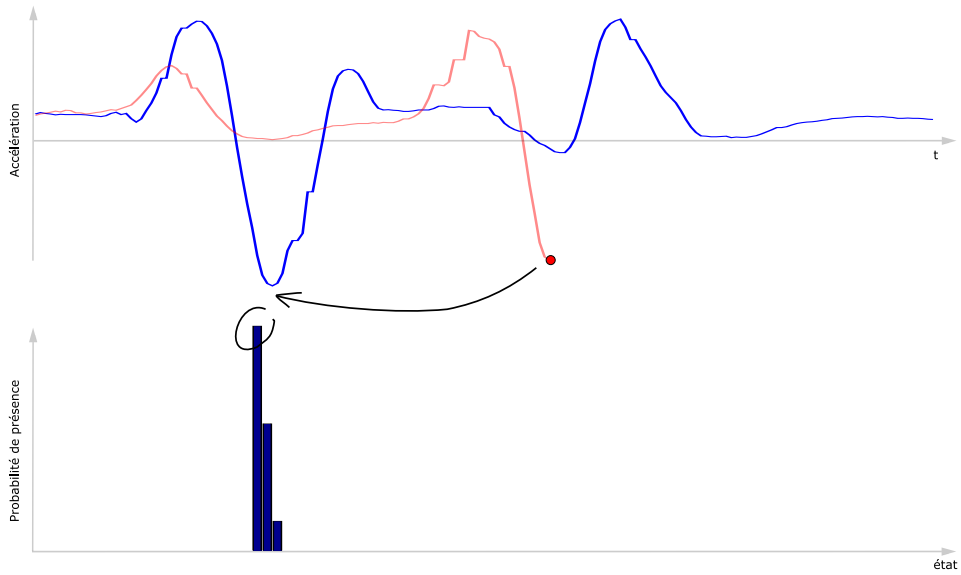


FIG. 2.4 – Le signal de référence (en bleu), le signal entrant (en rouge). L'état dans lequel la probabilité de se trouver à l'instant $t$ est la plus grande correspond au maximum de la matrice d'estimation $E$ (historgramme).

Afin d'extraire la meilleure suite d'état qui produit les observations, soit sélectionner le geste le plus ressemblant dans notre base de donnée gestuelle, nous calculons à chaque instant $t$ et pour chaque geste l'estimation globale de ressemblance $x_t$ :

$$x_t = \sum_{i=1}^{N} e_i$$

## 2.2 Amélioration de la phase d'observation

Dans le modèle précédemment décrit, la matrice d'observation $B$ est calculée à partir de la valeur courante du signal d'accélération. Pourtant, nous souhaitons pouvoir suivre un geste même si ce dernier est exécuté avec une amplitude différente et qui donnera donc des valeurs d'accélération différentes. En considérant uniquement la valeur courante d'accélération, nous pouvons obtenir des résultats ambiguës dans notre matrice d'observation. Dans l'exemple figure 2.5, les deux signaux représentent le même coup d'archet. Le signal entrant est situé vers la fin de ce coup d'archet alors que la matrice d'observation nous donne une plus grande probabilité de se situer vers le début du mouvement. En effet, la valeur de l'accélération du signal entrant est plus proche de la valeur d'accélération du début du signal de référence. Le suivi ne peut qu'en être perturbé et quelquefois bloqué si la matrice d'observation donne des valeurs d'émission trop faibles au niveau de l'état auquel nous nous trouvons.

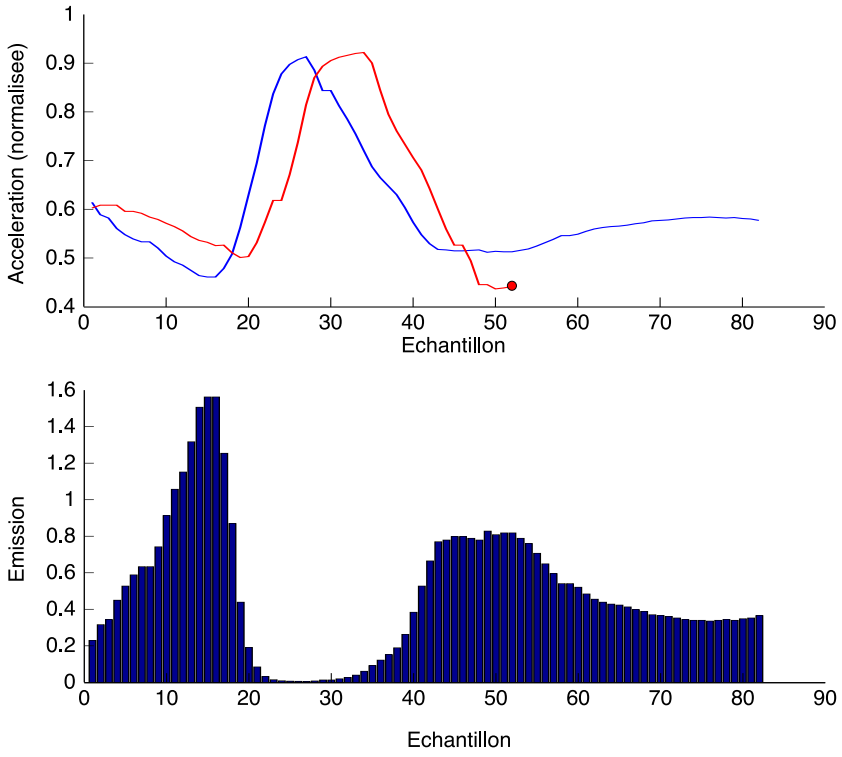


FIG. 2.5 – Le signal de référence (en bleu), le signal entrant (en rouge). Etant situé vers la fin du geste de référence, la matrice d'observation (histogramme) donne plus de probabilité de se trouver en début de geste. Ceci est du à une différence dans l'intensité d'exécution du geste.

Il en découle la nécessité d'avoir une observation du signal (soit une

mesure de ressemblance) prenant en compte le "bruit gestuel" afin de le "filtrer" pour permettre un suivi indépendant de l'amplitude du geste analysé. Il existe également une source de "bruit gestuel" dans la vitesse d'exécution d'un geste qu'il nous faut prendre en compte. Pour ce faire, nous modélisons notre signal entrant $V(t)$ tel :

$$V(t) = \epsilon_{intens2}(t).V_{geste}(f(t)) + \epsilon_{intens1}(t) + \epsilon_0$$

$V_{geste}$ caractérisant le geste en question, $f(t)$ représentant les variations temporelles dans sa vitesse d'exécution, $\epsilon_{intens1}(t)$ et $\epsilon_{intens2}(t)$ les variations d'intensité (offset et coefficient multiplicateur), et $\epsilon_0$ le bruit provenant des différents capteurs. Les trois premières méthodes présentées ci-dessous se concentrent sur la prise en compte de $\epsilon_{intens1}(t)$ et $\epsilon_{intens2}(t)$. La méthode n°4 se concentre sur la prise en compte de la fonction $f(t)$.

### 2.2.1 Méthode n°1 : fenêtrage

Dans l'exemple montré figure 2.5, il nous est évident que nous nous trouvons vers la fin du geste de référence car nous avons accès aux différentes valeurs d'accélérations qui ont précédés celle de l'instant présent. Nous pouvons reconnaître la variation d'accélération du geste entrant dans le geste de référence. Il nous faut donc prendre en compte non pas uniquement la valeur d'accélération courante mais également celles qui lui précèdent sur une durée déterminée.

Une première solution consiste donc à calculer la distance entre notre signal $V$ entrant fenêtré sur un intervalle $[k-\tau : k] : V_{k,\tau} = \{v_{k-\tau}, \ldots, v_k\}$ avec l'ensemble du signal de référence fenêtré lui aussi : $U_{j,\tau} = \{u_{j-\tau}, \ldots, u_j\}$. Le résultat est pondéré par une fenêtre exponentielle dissymétrique.

$$dist_1(V_{k,\tau}, U_{j,\tau}) = \sqrt{\sum_p \sum_{t=0}^{\tau} e^{-\alpha t}[v_{k-t} - u_{j-t}]^2}$$

Pour notre précédent exemple, en choisissant $\tau = 20$ échantillons, la matrice d'observation nous donne une distribution plus cohérente (histogramme Fig. 2.6) présentant son maximum vers la fin du geste. En effet, contrairement au minimum d'accélération au niveau du échantillon 16, les valeurs d'accélérations précédant celle de l'échantillon 52 atteignent un valeur d'accélération de 0.9, comme pour notre signal entrant. La distance entre ces deux forme est donc proche : la probabilité d'émission est donc plus forte.

Pourtant, ce calcul de distance reste dépendant des valeurs absolues d'accélération. Afin de ne comparer que la forme de notre signal fenêtré, nous pouvons retrancher la valeur de l'accélération $u_j$ du geste de référence sur l'ensemble du signal fenêtré $U_{j,\tau}$ ainsi que la valeur de l'accélération $v_k$ du geste entrant sur l'ensemble du signal $V_{k,\tau}$ :

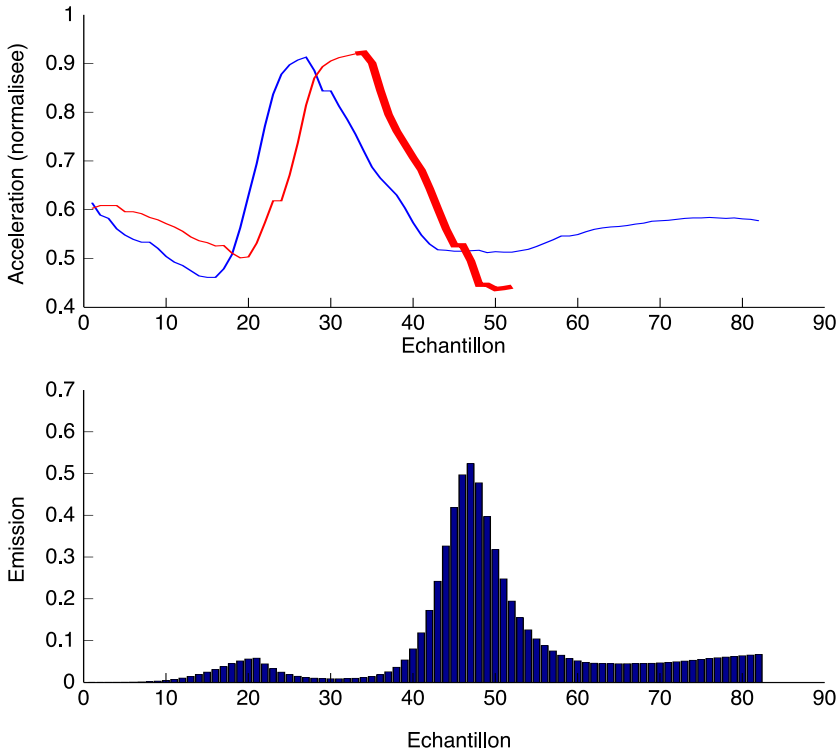


FIG. 2.6 – Le signal de référence (en bleu), le signal entrant (en rouge - en gras sur les 20 derniers échantillons). La distribution des probabilités d'observation (histogramme) place bien l'état courant vers la fin du geste.

$$dist_2(V_{k,\tau}, U_{j,\tau}) = \sqrt{\sum_p \sum_{t=0}^{\tau} e^{-\alpha t}[(v_{k-t} - v_k) - (u_{j-t} - u_j)]^2}$$

Ainsi, si la taille de la fenêtre est de 2 échantillons ($\tau = 1$), cela revient à observer la dérivée (discrète et pondérée) des différents signaux. Toutefois, une valeur de $\tau$ plus conséquente est conseillée afin d'avoir une information sur la forme de la fonction et non sur sa pente locale. Sur notre même exemple, nous pouvons comparer (Fig. 2.7) la différence que nous obtenons dans notre matrice d'observation où l'histogramme rouge représente la matrice d'observation obtenue avec la fonction $dist_1()$ et le bleu avec la fonction $dist_2()$.

Le maximum de la distribution des probabilités d'observation obtenue avec la fonction $dist_1()$ est un peu différent de celui obtenu avec la fonction $dist_2()$. Dans notre exemple, la distribution est également plus discriminante. En effet, au niveau des échantillons 10 à 30 (là où il y avait une ambiguïté en ne prenant en compte que la valeur d'accélération courante), les probabilités

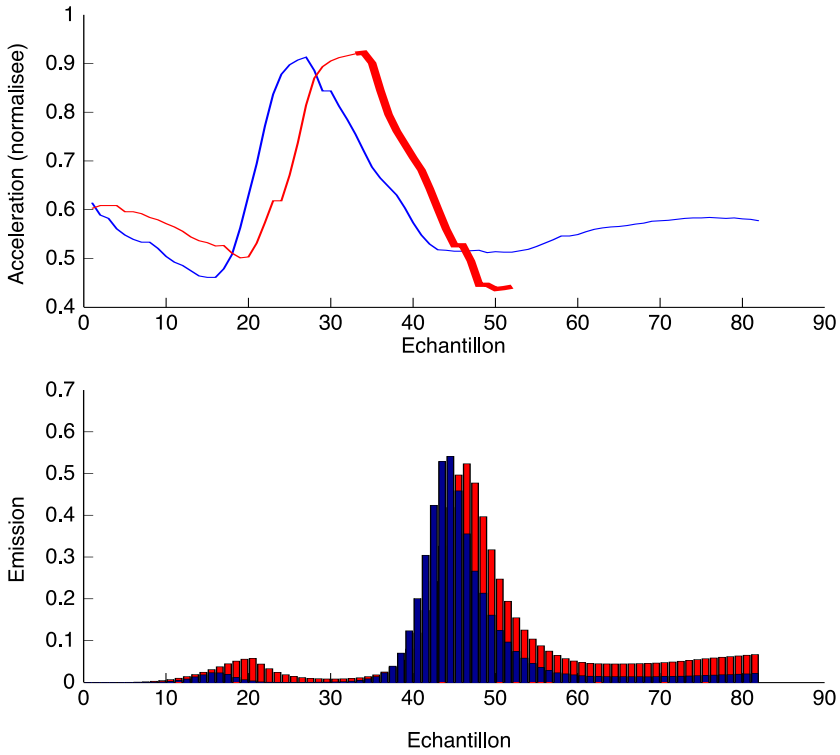


FIG. 2.7 – Le signal de référence (en bleu), le signal entrant (en rouge - en gras sur les 20 derniers échantillons). La nouvelle distribution des probabilités d'émission (histogramme en bleu) semble être plus discriminante que l'ancienne (histogramme en rouge).

d'émission sont plus faible. Quand nous effectuons la multiplication de la matrice d'observation $B$ avec la matrice d'estimation $E$, les probabilités de se trouver dans les états correspondants à ces échantillons sont donc amoindries.

Cette modification prend en compte le bruit gestuel modélisé par $\epsilon_{intens1}$ correspondant à un changement d'offset dans l'accélération.

### 2.2.2 Méthode n°2 : régression linéaire

La solution présentée ci-dessus n'est qu'une approximation dans le but d'obtenir un filtre sur notre bruit gestuel d'intensité. En effet, la méthode adoptée suppose que les différences d'interprétation de la part du violoniste ne sont constituées que de transposition de valeurs d'accélérations ce qui n'est bien évidement pas le cas. Une meilleure prise en compte de ce bruit consiste à supposer que l'accélération du geste subit une transformation linéaire.

Soit deux séquences $U = \{u_1, u_2, \ldots, u_N\}$ et $V = \{v_1, v_2, \ldots, v_N\}$. Elles ont une forte relation linéaire si $V \approx \beta_0 + \beta_1 U$. Afin de déterminer les valeurs des coefficients $\beta_0$ et $\beta_1$, et de mesurer la qualité de l'interpolation il faut

effectuer une analyse statistique de la distribution des couples de valeurs $(u_1, v_1), (u_2, v_2), \ldots, (u_n, v_n)$ dans l'espace $U - V$. Si les deux séquences ont une forte relation linéaire, la distribution des points se répartissent autour d'une droite appelée *droite de régression linéaire* (Fig. 2.8).

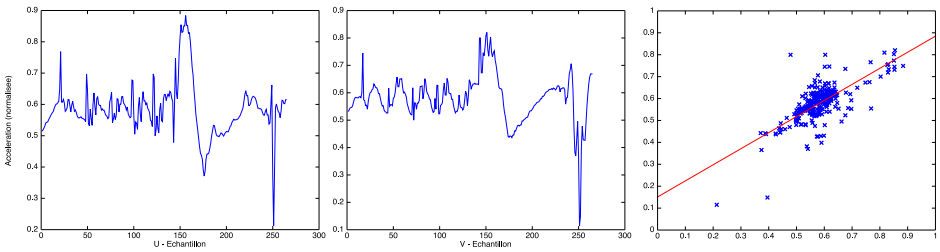


FIG. 2.8 – Deux séquences $U$ et $V$ ainsi que la répartition des couples de valeurs autour de la droite de régression linéaire dans le plan $U - V$.

Pour ce faire, nous partons du modèle $V = \beta_0 + \beta_1 U + \epsilon$ où $\epsilon$ est un terme d'erreur. Le but est de minimiser $\epsilon$ au sens des moindres carrés soit minimiser la fonction $Q(\beta_0, \beta_1)$ :

$$Q(\beta_0, \beta_1) = \sum_{i=1}^{n} \epsilon_i^2 = \sum_{i=1}^{n} (v_i - (\beta_0 + \beta_1 u_i))^2$$

soit

$$\frac{\partial Q}{\partial \beta_0} = 0 \qquad \text{et} \qquad \frac{\partial Q}{\partial \beta_1} = 0$$

Nous obtenons alors

$$\beta_0 = \overline{V} - \beta_1 \overline{U} \qquad \text{et} \qquad \beta_1 = \frac{\sum_{i=1}^{n} (u_i - \overline{U})(v_i - \overline{V})}{\sum_{i=1}^{n} (u_i - \overline{U})^2}$$

où

$$\overline{U} = \frac{1}{n} \sum_{i=1}^{n} u_i \qquad \text{et} \qquad \overline{V} = \frac{1}{n} \sum_{i=1}^{n} v_i$$

Afin de mesurer la qualité de l'interpolation, nous calculons le coefficient de détermination $R$ :

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \epsilon_i^2}{\sum_{i=1}^{n} (v_i - \overline{V})^2}$$

qui peut également être obtenu par :

$$R^2 = \frac{[\sum_{i=1}^{n} (u_i - \overline{U})(v_i - \overline{V})]^2}{\sum_{i=1}^{n} (u_i - \overline{U})^2 \sum_{i=1}^{n} (v_i - \overline{V})^2} \qquad \text{avec } R^2 \in [0, 1]$$

Plus le résultat est proche de 1, plus l'interpolation est bonne, donc $R^2(U, U) = 1$. La fonction est également symétrique : $R^2(U, V) = R^2(V, U)$[1].

[1]Ces calculs peuvent être généralisés sur $p$ variables (lorsque nous traitons plusieurs signaux délivrés par les capteurs de l'archet augmenté) avec la *régression linéaire multiple*

Nous utilisons la fonction $R^2$ pour mesurer la similarité des signaux fenêtrés sur un intervalle $[k-\tau : k]$. Il faut également prendre en compte la valeur du coefficient $\beta_1$. En effet, si ce dernier est éloigné de l'unité (pente à 45°), les courbes en question ne sont pas similaires. Il nous faut donc considérer la différence entre la pente de la droite de régression linéaire (de manière linéaire, donc en prenant l'arctangente du coefficient $\beta_1$) par rapport à la pente de 45°. Nous définissons alors :

$$dist_3(V_{k,\tau}, U_{j,\tau}) = (atan(\beta_1) - 1)^2$$

avec $\beta_1$ la pente de la droite de régression linéaire des séquences $U_{j,\tau} = \{u_{j-\tau}, \ldots, u_j\}$ et $V_{k,\tau} = \{v_{k-\tau}, \ldots, v_k\}$, ainsi que la nouvelle matrice d'observation $B$, qui prend en compte $\beta_1$ et $R^2$, avec :

$$b_j(k) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-dist_3(v_k, u_j)}{2\sigma^2}} . R^2(U_{j,\tau}, V_{k,\tau})$$

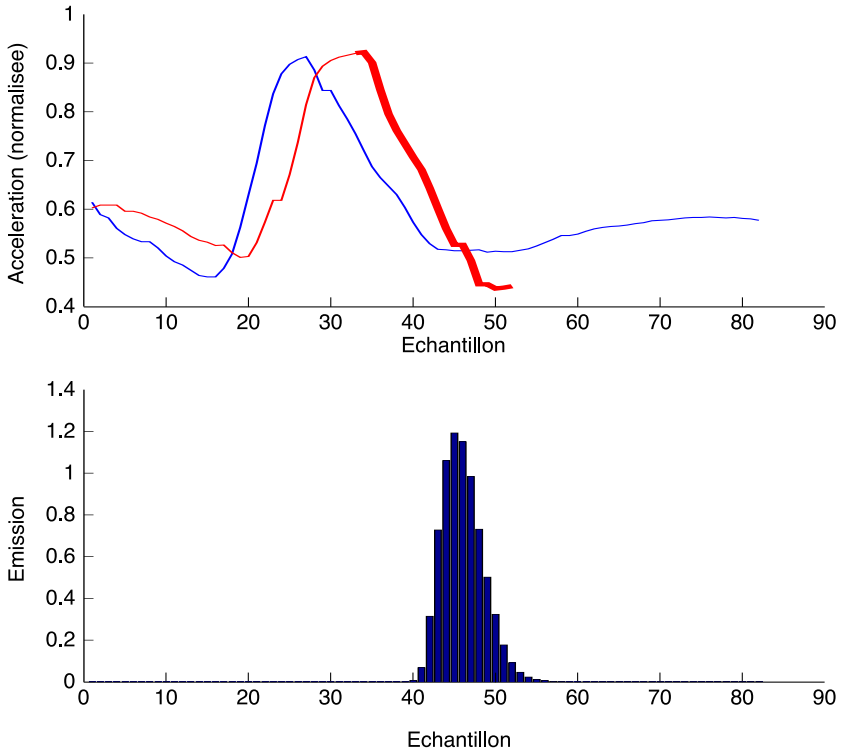


FIG. 2.9 – Le signal de référence (en bleu), le signal entrant (en rouge - en gras sur les 20 derniers échantillons). La distribution des probabilités d'émission (histogramme) ne donne que des valeurs significatives vers la fin du geste.

Sur notre exemple, avec $\tau = 20$, cela donne une répartition des probabilités d'observation telle que nous pouvons la voir figure 2.9. Sur la figure 2.10,

nous affichons la distribution des couples $\{(u_{j-\tau}, v_{k-\tau}), \ldots, (u_j, v_k)\}$ pour $j = max_{index}(B)$ sur le plan $U - V$ avec sa droite de régression linéaire.

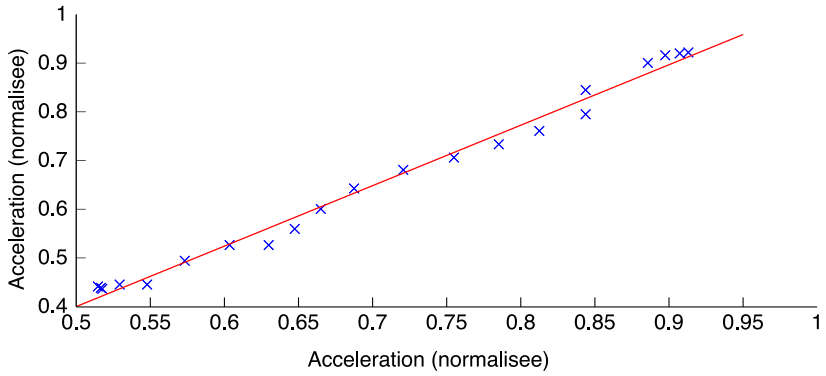


FIG. 2.10 – Au niveau de l'échantillon correspondant au maximum de la distribution des probabilités d'émission précédente, la relation entre les deux signaux fenêtrés sur 20 échantillons est fortement linéaire.

Nous pouvons constater que, sur notre exemple, la distribution obtenue est encore plus discriminante que les précédentes. Elle nous donne également directement une information sur la transformation linéaire modélisant la différence entre le geste entrant et le geste de référence (soit le bruit gestuel modélisé par $\epsilon_{intens2}$) grâce au coefficient $\beta_1$. Cette différence peut s'interpréter comme la différence d'intensité dans le jeu du violoniste (dans le cadre de notre modèle, qui considère ces différences comme des transformations linéaires dans l'accélération de l'archet).

### 2.2.3 Méthode n°3 : utilisation du Dynamic Time Warping

La méthode basée sur la régression linéaire suppose que les différentes intensités dans le jeu du violoniste correspondent à des transformations linéaires dans l'accélération de l'archet. L'ordre de l'interpolation peut être augmentée, mais ce modèle reste très réducteur par rapport aux réalités d'interprétation. Il nous faut prendre en compte les petites variations dans le mouvement de l'archet pour notre matrice d'observation. Pour ce faire, nous pouvons faire appel à l'algorithme de *Dynamic Time Warping* ($DTW$) qui réalise un alignement non-linéaire et qui nous permet de mesurer la ressemblance entre deux signaux présentant des différences d'exécution locales du geste. Il nous suffit d'utiliser cet algorithme de manière locale (sur une fenêtre de largeur $\tau$) dans notre modèle de suivi.

Nous repartons de nos deux séquences $U = \{u_1, u_2, \ldots, u_N\}$ et $V = \{v_1, v_2, \ldots, v_N\}$. Dans un premier temps, nous devons aligner ces deux séquences l'une sur l'autre à l'aide du $DTW$ afin de pouvoir en tirer une valeur de ressemblance. Un alignement est défini par une séquence $A =$

$\{a_1, a_2, \ldots, a_K\}$ dont chaque élément $a_k = (m_k, n_k)$ lie un échantillon $u_m$ à un échantillon $v_n$.

Plusieurs contraintes sont fixées :

- si $a_k = (m_k, n_k)$ et $a_{k-1} = (m_{k-1}, n_{k-1})$, alors $m_k \geq m_{k-1}$ et $n_k \geq n_{k-1}$ : l'alignement ne peut pas revenir en arrière.
- si $a_k = (m_k, n_k)$ et $a_{k-1} = (m_{k-1}, n_{k-1})$, alors $m_k \leq m_{k-1} + 3$ et $n_k \leq n_{k-1} + 3$ : le signal aligné ne peut pas aller trois fois plus vite que le signal de référence
- $a_1 = (1, 1)$ : l'alignement doit commencer sur les premiers échantillons des signaux

La distance locale $d(u_m, v_n)$ entre les deux échantillons $u_m$ et $v_n$ de nos deux séquences est :

$$d(u_m, v_n) = d(m, n) = \sqrt{(u_m - v_n)^2}$$

La première étape de l'algorithme consiste à calculer la matrice des distance augmentée $adm$ de taille $N \times N$, où $adm(m, n)$ représente la plus petite distance augmentée du point de coordonnées (1,1) au point de coordonnées (m,n). La valeur de la distance augmentée au point de coordonnées (m,n) est définie récursivement à partir de la distance locale $d(u_m, v_n)$ et des distances augmentées dans l'entourage du point concerné : $adm(i, j)$ avec $i < m$ et $j < n$ (Fig. 2.11). Ces distance sont pondérées en fonction de leurs provenances par les coefficients $[w_v, w_h, w_d]$. Les valeurs utilisées sont $[1, 1, 2]$.

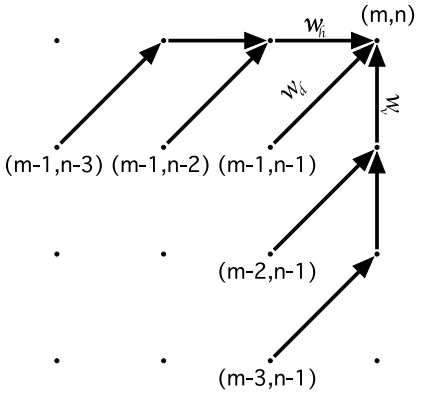


FIG. 2.11 – Calcul de la distance augmentée au point(m,n).

$adm(m, n)$ est défini par :

$$adm(m,n) = min \begin{cases} adm(m-1, n-1) + w_d.d(u_m, v_n) \\ adm(m-2, n-1) + w_v.d(u_m, v_n) + w_d.d(u_{m-1}, v_n) \\ adm(m-1, n-2) + w_h.d(u_m, v_n) + w_d.d(u_m, v_{n-1}) \\ adm(m-3, n-1) + w_v.d(u_m, v_n) + w_d.d(u_{m-2}, v_n) + w_v.d(u_{m-1}, v_n) \\ adm(m-1, n-3) + w_h.d(u_m, v_n) + w_h.d(u_m, v_{n-1}) + w_d.d(u_m, v_{n-2}) \end{cases}$$

Une fois la matrice construite, il suffit de trouver le chemin inverse (partant de la fin du signal vers le point de coordonnées (1,1) ; l'algorithme a donc besoin des signaux entier et ne peut être exécuté en temps réel) qui minimise la somme des distance augmentées rencontrées. Ainsi, nous obtenons la séquence $A$ qui aligne $U$ sur $V$. La ressemblance entre ces deux séquences est d'autant plus élevée que la somme des distance augmentée rencontrées sur le chemin sélectionné est faible. Pour mesurer la similarité entre nos deux séquences $U_{j,\tau} = \{u_{j-\tau}, \ldots, u_j\}$ et $V_{k,\tau} = \{v_{k-\tau}, \ldots, v_k\}$, nous définissons :

$$dist_4(V_{k,\tau}, U_{j,\tau}) = \sum_{i=1}^{K} adm(a_i)$$

Cette distance peut être calculé directement sur $V_{k,\tau}$ et $U_{j,\tau}$ ou sur leurs dérivées respectives. Les matrices d'observation que l'on obtient pour $\tau = 20$ sur notre exemple sont montrées figure 2.12

Sur notre exemple, la distribution de probabilité d'observation en effectuant un $DTW$ local sur le signal reste sensiblement la même que celle obtenue avec la méthode précédente. Par contre, la distribution obtenue à l'aide du $DTW$ local sur la dérivée des signaux donne de moins bon résultats. En effet, la méthode ici adoptée ne réalise pas à proprement parler de calcul de ressemblance mais tente, de manière locale, de trouver un alignement entre nos signaux fenêtré. De petits détails peuvent avoir de grosses conséquences dans ce que nous considérons comme la mesure de similarité entre nos deux signaux. Mais, en ayant accès à la séquence $A$, nous obtenons une information plus fine (car non-linéaire) sur ces variations dans les mouvement de l'archet.

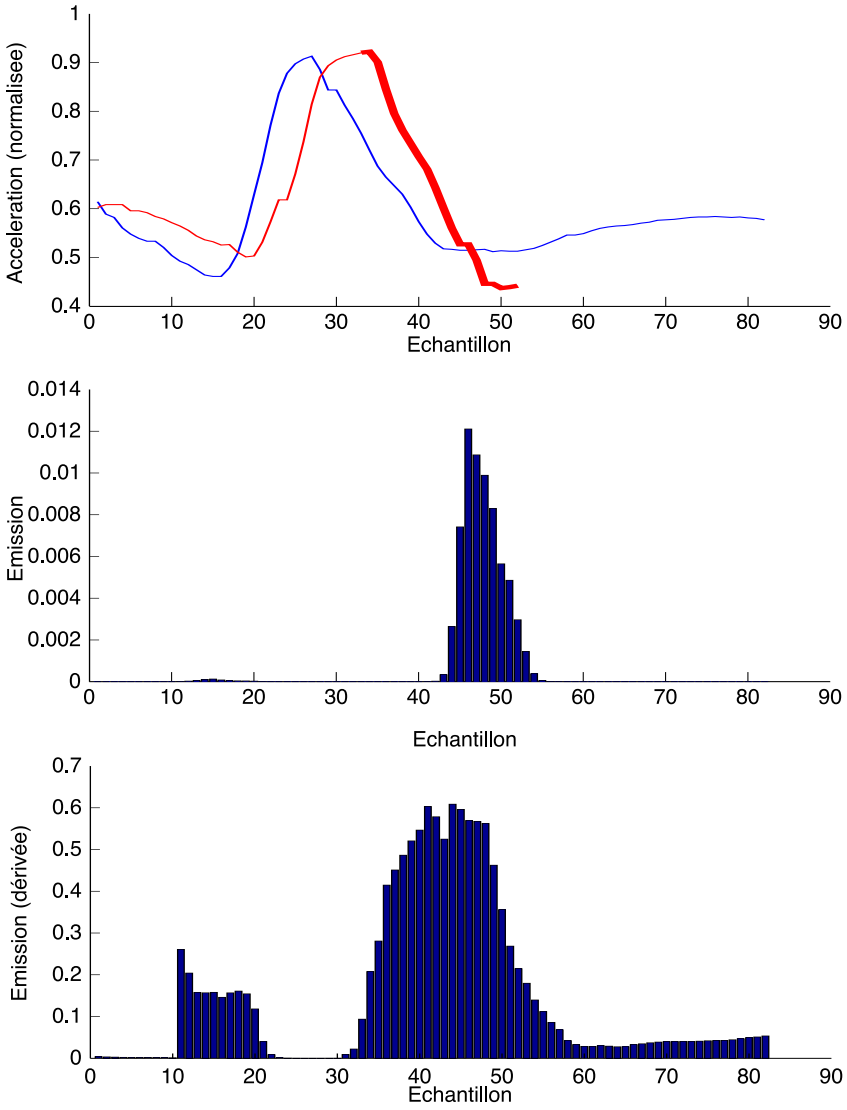


FIG. 2.12 – Le signal de référence (en bleu), le signal entrant (en rouge - en gras sur les 20 derniers échantillons). La distribution des probabilités d'émission obtenue à l'aide d'un $DTW$ sur le signal (histogramme 1) et celle obtenue sur la dérivée du signal (histogramme 2).

### 2.2.4 Méthode n°4 : ressemblance sur une base de fonctions d'échelles

Concentrons nous maintenant sur les variations d'exécution d'un même geste, soit la source de bruit modélisée par la fonction $f(t)$. Une première approximation consiste à considérer cette fonction comme étant linéaire localement :

$$V(t) = V_{geste}(\alpha(t)t) \qquad \text{avec} \qquad \alpha(t) = C^{ste} \text{ pour } t \in [k-\tau : k]$$

Il nous faut fixer des limites au facteur $\alpha$. En effet, un geste effectué dix fois plus rapidement qu'un autre ne peut plus vraiment être considéré comme un geste semblable, même si la fonction qui le représente a la même forme (contractée d'un facteur 10). Nous posons donc $1/2 \leq \alpha(t) \leq 2$.

Il nous faut donc comparer le signal de référence fenêtré sur $[j-\tau : j]$ (pour $j \in [1 : N]$) par rapport à notre signal entrant, fenêtré sur l'intervalle $[k-\tau : \tau]$ et déformé linéairement dans le temps d'un facteur $\alpha$. Pour construire la base des signaux déformés, nous interpolons linéairement les valeurs de notre signal entrant fenêtré (sur 20 échantillons figure 2.14). Nous utilisons 20 échelles différentes (entre $1/2$ et 2) de notre signal fenêtré afin de construire notre base.

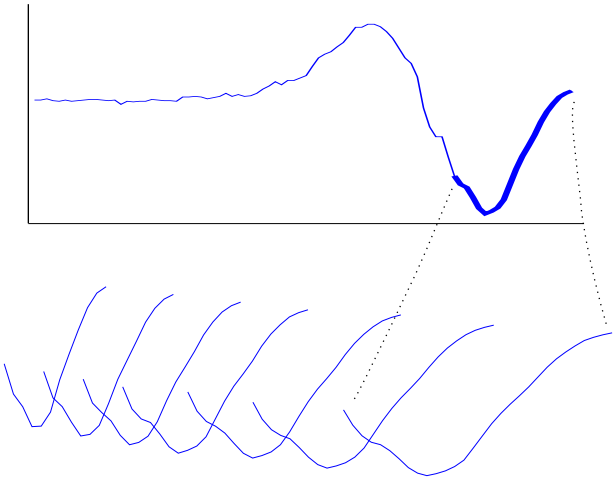

FIG. 2.13 – Le signal entrant (en bleu - en gras sur les 20 derniers échantillons) ainsi que quelques vecteurs de la base de fonctions d'échelle.

Il suffit désormais d'utiliser les fonctions d'observation précédemment définies afin de calculer la ressemblance entre les différents signaux fenêtrés du signal de base et, non plus la seule fonction du signal entrant fenêtré, mais l'ensemble de la base de ce signal déformé. Nous n'obtenons alors plus qu'une seule matrice d'observation mais le même nombre de matrice que de vecteurs de notre base (ici 20, Fig. 2.14).

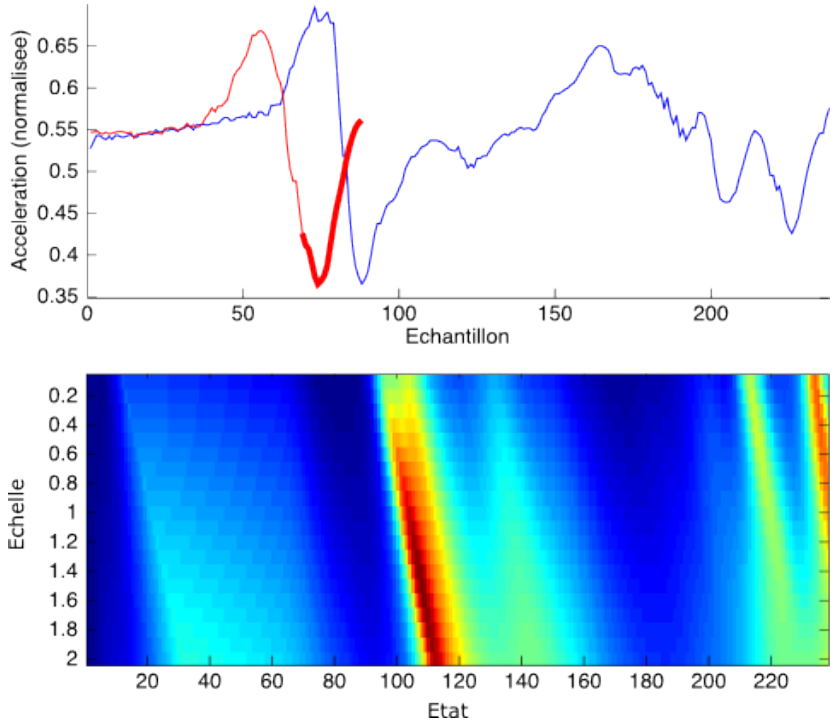


FIG. 2.14 – Le signal de référence (en rouge), le signal entrant (en bleu - en gras sur les 20 derniers échantillons) ainsi que les probabilités d'émission calculées à l'aide de la fonction $dist_1()$ en fonction de l'échelle.

Pour pouvoir faire la multiplication terme à terme avec la matrice d'estimation de probabilité de présence $E$, il nous suffit de récupérer la valeur maximale entre les 20 matrices représentant les différentes échelles, pour chaque état. Cette méthode nous permet d'obtenir à chaque instant une vitesse d'exécution relative locale : une fois l'état correspondant au maximum de la fonction d'estimation $E$ déterminé (état dans lequel nous avons le plus de chance de nous trouver), il suffit de récupérer l'index du maximum de la fonction d'observation $B$ au niveau de l'état en question afin d'en déduire la déformation locale la plus ressemblante de notre signal et donc la vitesse d'exécution locale.

# Chapitre 3

# Résultats

Nous présentons ici les différents résultats obtenus à partir des modifications apportées à notre modèle de suivi et de reconnaissance du geste ainsi que la méthodologie adoptée.

## 3.1 Méthodologie

Afin d'évaluer le suivi il nous faut un alignement de référence du geste suivi sur le geste de référence. Un alignement manuel des différents signaux les uns sur les autres est trop fastidieux. Nous tentons de mettre au point un dispositif d'évaluation plus rapide.

### 3.1.1 Comparaison avec le $DTW$

Afin d'évaluer notre suivi, soit l'alignement en temps réel d'un signal entrant sur un signal de référence, il nous faut pouvoir le comparer à un alignement de référence. Ce dernier peut être réalisé au moyen de l'algorithme de $DTW$ précédemment présenté et qui est utilisé dans la plupart des applications demandant un alignement de courbes non-temps réel (Fig. 3.1). Cet algorithme est non-causal ; il demande à avoir la totalité du signal pour pouvoir réaliser un alignement. Il ne peut être utilisé pour un alignement en temps réel, c'est-à-dire au fur est à mesure que les données arrive, tel que nous le prenons en compte avec le Modèle de Markov Caché.

Ainsi, nous pouvons comparer les différences entre les séquences $A = \{a_1, a_2, \ldots, a_K\}$ délivrée par l'alignement en temps réel et $A^{dtw} = \{a_1^{dtw}, a_2^{dtw}, \ldots, a_K^{dtw}\}$ obtenue grâce au $DTW$ hors temps réel.

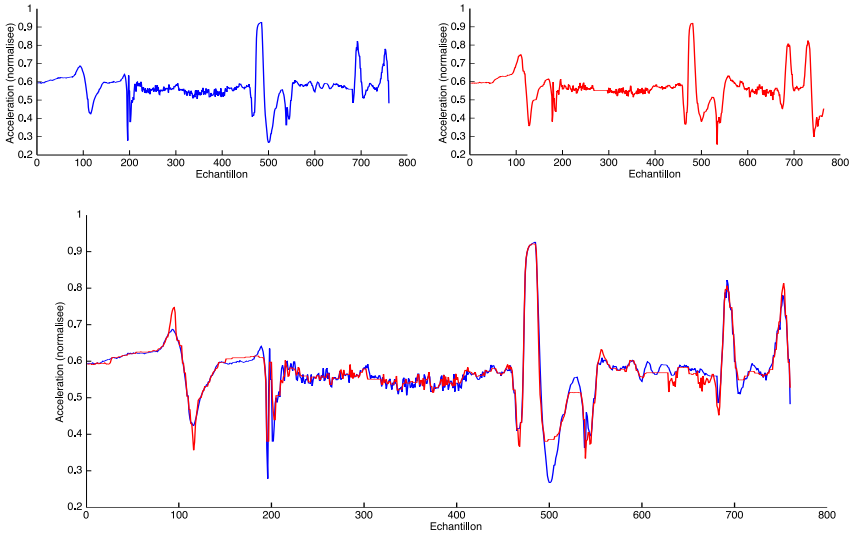


FIG. 3.1 – Réalignement du signal caractérisant un geste (en rouge) sur un signal caractérisant une autre exécution de ce même geste (en bleu) à l'aide de l'algorithme de $DTW$.

### 3.1.2 Signaux testés

Afin de mettre en avant les différentes transformations du geste qui sont prisent en compte par les modifications du modèle, nous les testons sur des signaux de synthèse présentant ces transformations.

– Deux signaux représentant un même geste. Seule l'amplitude change. Pour les méthodes n°1, 2 et 3. Fig. 3.2. $\alpha = 1.6$.

$$V_1(t) = V_{geste}(t) + \epsilon_{intens}(t) + \epsilon_0 \qquad V_2(t) = V_{geste}(t) + \alpha.\epsilon_{intens}(t) + \epsilon_0$$

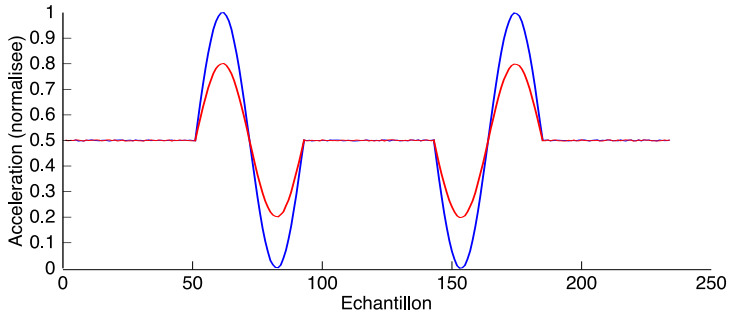


FIG. 3.2 – Deux signaux représentant un même geste où seule l'amplitude change.

– Deux signaux représentant un même geste. L'un est une version contractée de l'autre. Pour la méthode n°4. Fig. 3.3. $\alpha = 2$.

$$V_1(t) = V_{geste}(t) + \epsilon_0 \qquad V_2(t) = V_{geste}(\alpha.t) + \epsilon_0$$

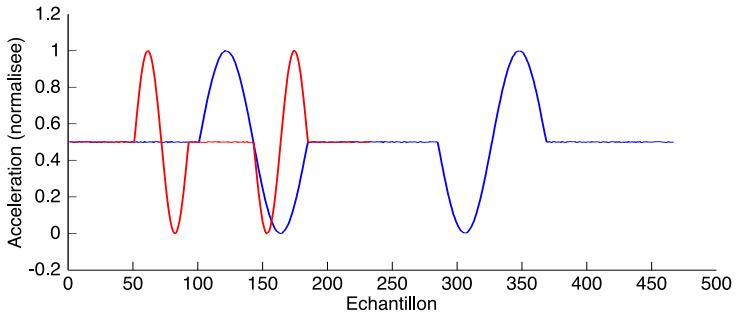


FIG. 3.3 – Deux signaux représentant un même geste où l'un est une version contractée de l'autre.

Les méthodes 1, 2 et 3 sont ensuite testées sur une base de donnée acquise lors de répétitions dans le cadre du projet pour quatuor augmenté de Florence Baschet. Il s'agit de différentes phrases musicales réalisées par deux violonistes différents.

## 3.2 Résultats

### 3.2.1 Modification de l'amplitude

Lorsque l'on applique l'algorithme de $DTW$ (hors temps réel) sur les signaux ayant pour différence l'amplitude du geste, nous obtenons le résultat figure 3.4. La séquence$A^{dtw}$ que nous obtenons à partir de cet alignement devient notre alignement de référence auquel nous allons comparer les alignements obtenus en temps réel avec les différentes fonction d'observation.

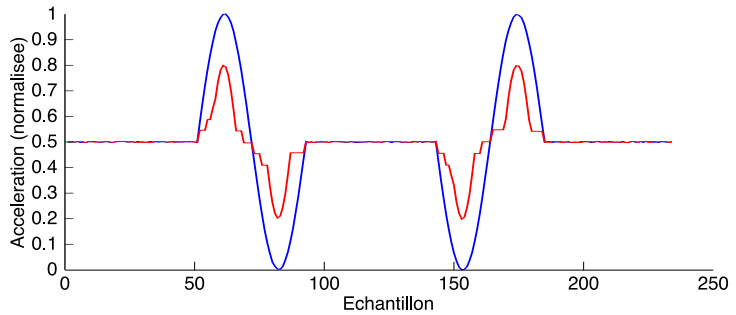


FIG. 3.4 – Alignement du signal entrant (en rouge) sur le signal de référence (en bleu) à l'aide l'algorithme de $DTW$ sur la dérivée du signal.

L'alignement en temps réel effectué à l'aide de la fonction $dist()$ (qui ne prend en compte que la valeur d'accélération courante) est présenté figure 3.5.

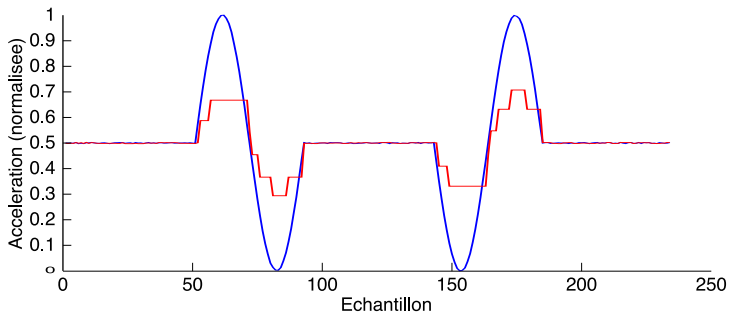


FIG. 3.5 – Alignement du signal entrant (en rouge) sur le signal de référence (en bleu) à l'aide de la fonction $dist()$

La comparaison entre les deux séquences $A$ produites par l'algorithme de $DTW$ hors temps réel et le suivi en temps réel avec la fonction $dist()$ est présentée figure 3.6. Nous pouvons y voir les écarts entre les deux alignements au niveau des changements d'amplitude entre les deux signaux (entre les échantillons 50 et 100, ainsi qu'entre 150 et 200). Le suivi fait alors des saut d'un état à un autre car l'amplitude du signal de référence est différente. Les grandes différences que l'on constate au début de l'alignement sont provoquées par deux conditions à $t = 0$ différentes : pour l'alignement de référence, nous imposons un départ commun entre les deux fonctions alors que pour l'alignement en temps réel, la distribution initiale de probabilité de présence est répartie de manière homogène sur l'ensemble des états du signal de référence. Le signal n'ayant pas de forme particulière en son début, l'état dans lequel nous nous trouvons peut se situer sur une des trois parties plate de notre signal. Le suivi en temps réel retrouve très rapidement l'alignement de référence.

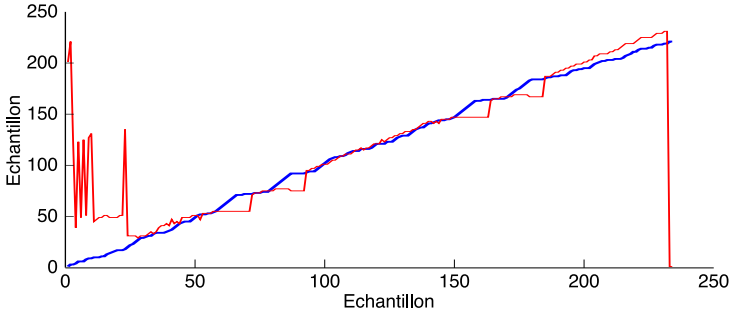


FIG. 3.6 – Séquence d'alignement de référence (en bleu) et séquence d'alignement obtenue à l'aide de la fonction $dist()$

En effectuant l'alignement sur ces mêmes signaux à l'aide de la fonction $dist_2()$ (qui prend en compte les valeurs d'accélération sur une fenêtre de 20 échantillons[1]) , nous obtenons le résultat figure 3.7. Le suivi est meilleur car il n'y a plus de saut au niveau des changements d'amplitude, mais nous pouvons remarquer un petit retard dans l'alignement sur le début du geste par rapport à la méthode précédente. Afin de pouvoir comparer les différentes

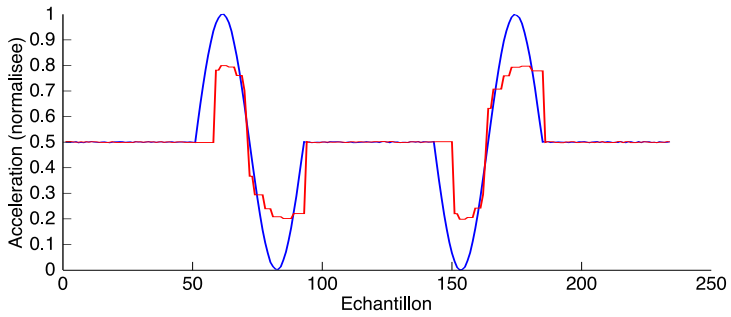


FIG. 3.7 – Alignement du signal entrant (en rouge) sur le signal de référence (en bleu) à l'aide de la fonction $dist_2()$

méthodes, nous affichons la différence entre la séquence $A^{dtw}$ de référence et la séquence $A$ obtenue par chaque méthode (en valeur absolue) :

$$A_{diff} = \{abs(a_1^{dtw} - a_1), abs(a_2^{dtw} - a_2), \ldots, abs(a_K^{dtw} - a_K)\}$$

Ainsi, l'alignement peut être considéré comme bon si les éléments de la sé-

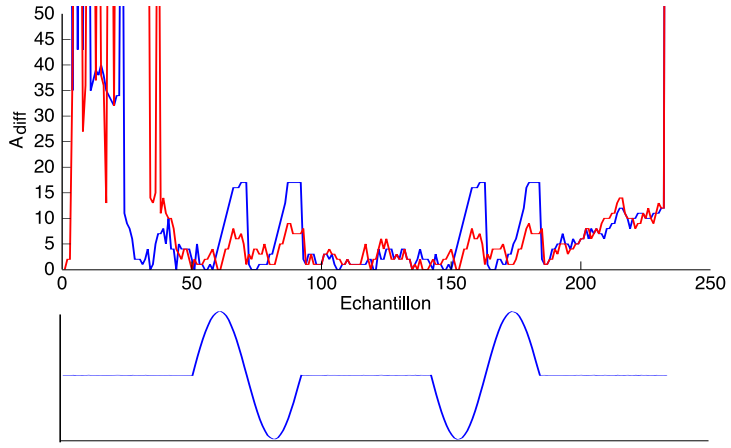


FIG. 3.8 – $A_{diff}$ de l'alignement obtenu à l'aide de la fonction $dist()$ en bleu et $dist_2()$ en rouge, ainsi que la fonction de référence en dessous en bleu.

[1]la taille de la fenêtre se choisie en fonction des signaux auxquels nous avons à faire : ici, 20 échantillons correspondent environ à une demi-période de sinusoïde de nos signaux de synthèse

quence $A_{diff}$ sont proches de zéro. Comparons les deux alignements obtenus avec la fonction $dist()$ et la fonction $dist_2()$ (Fig. 3.8). La fonction de référence est affichée en dessous afin de pouvoir mettre en rapport les problèmes d'alignement ($A_{diff}$ éloigné de zéro) avec le signal.

Nous pouvons remarquer que l'alignement effectué à l'aide de la fonction $dist_2()$ est plus robuste que celui effectué à l'aide de la fonction $dist()$ au niveau des changement d'amplitude dans le signal (entre les échantillons 50 et 100, ainsi qu'entre 150 et 200). Il en va de même mais de manière moins évidente entre la fonction $dist()$ et la fonction $dist_3()$ utilisant la régression linéaire (Fig. 3.9).

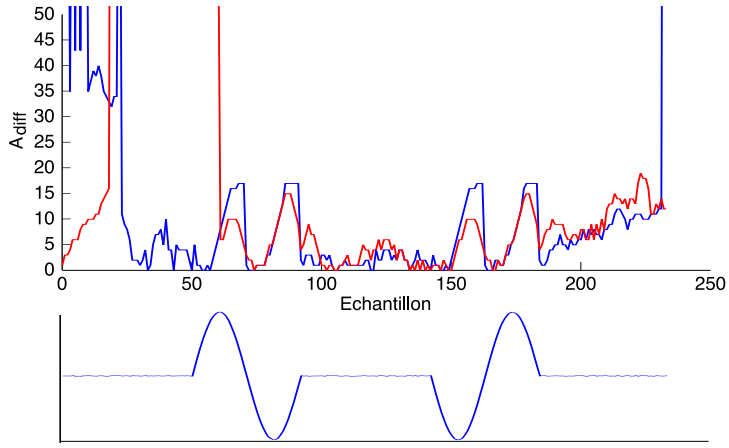


FIG. 3.9 – $A_{diff}$ de l'alignement obtenu à l'aide de la fonction $dist()$ en bleu et $dist_3()$ en rouge, ainsi que la fonction de référence en dessous en bleu.

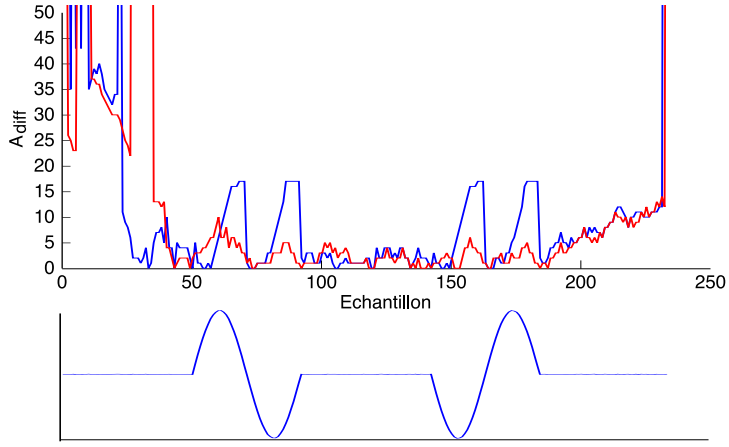


FIG. 3.10 – $A_{diff}$ de l'alignement obtenu à l'aide de la fonction $dist()$ en bleu et $dist_4()$ en rouge, ainsi que la fonction de référence en dessous en bleu.

La comparaison avec l'alignement effectuée à l'aide de la fonction $dist_4()$

(utilisant le $DTW$ local) est présenté figure 3.10. Le résultat est également plus robuste qu'avec la méthode standard (utilisation de la fonction $dist()$ qui ne prend en compte que l'accélération courante).

### 3.2.2 Modification de l'échelle temporelle

L'alignement réalisé avec l'algorithme de $DTW$ hors temps réel est montré figure 3.11. C'est notre alignement de référence.

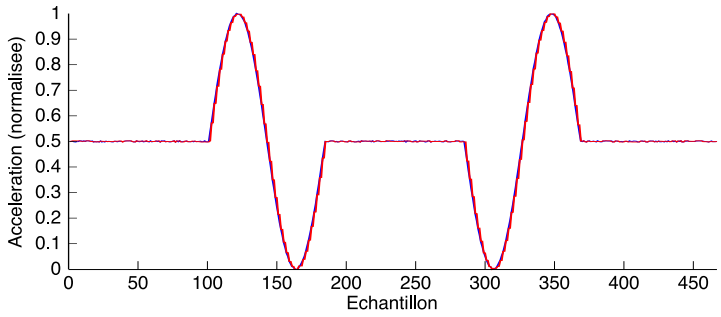


FIG. 3.11 – Alignement des deux signaux avec l'algorithme de $DTW$.

Comparons la séquence $A_{diff}$ (différence entre l'alignement de référence hors temps réel avec l'alignement temps réel) obtenues avec le suivi utilisant la fonction $dist()$ ainsi que celle obtenue avec le suivi utilisant la fonction $dist_1()$ avec la construction de la base de fonctions d'échelle (Fig. 3.12).

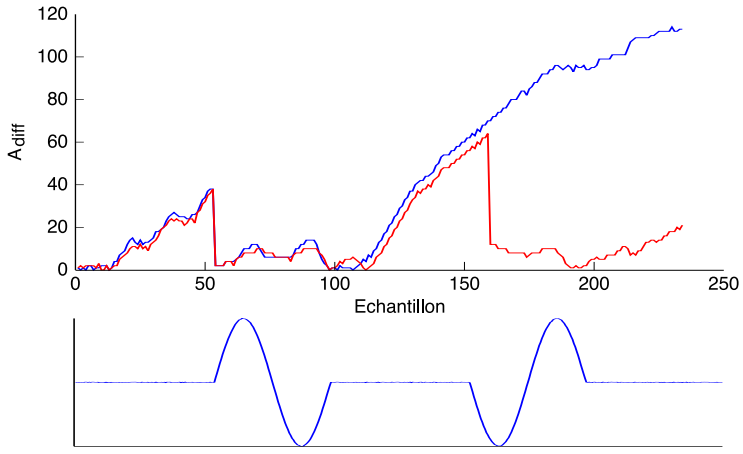


FIG. 3.12 – $A_{diff}$ de l'alignement obtenu à l'aide de la fonction $dist()$ en bleu et avec la fonction $dist_1()$ et la méthode n°4 en rouge, ainsi que la fonction de référence en dessous en bleu.

L'exécution du geste suivi est beaucoup plus rapide que le geste de référence. Les deux méthodes donnent un alignement semblable entre les échantillons 50 et 100. Mais quand le geste entrant effectue la seconde sinusoïde

(beaucoup plus rapidement que dans le geste de référence), le suivi utilisant la fonction $dist()$ (qui ne prend en compte que l'accélération courante) n'arrive pas à reconnaître la seconde sinusoïde du geste de référence. La différence entre l'alignement de référence et cet alignement croît. Par contre, grâce à la modification apportée par méthode n°4, le suivi arrive à reconnaître la version dilatée du geste entrant dans le geste de référence. C'est pour cela qu'on observe un saut dans le suivi peu après l'échantillon 150.

### 3.2.3 Signaux réels

Afin de tester nos modifications, nous choisissons une phrase ayant de grande différences dans la réalisation des deux exécutions. Les deux signaux à aligner en temps réel sont présenté figure 3.13 accompagné de l'alignement réalisé grâce à l'algorithme de $DTW$ hors temps réel.

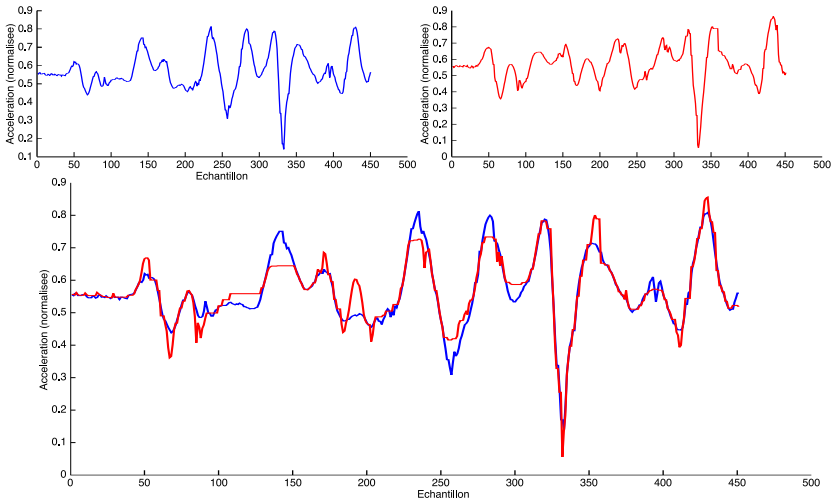


FIG. 3.13 – Réalignement du signal caractérisant une phrase (en rouge) sur un signal caractérisant une autre exécution de cette même phrase (en bleu) à l'aide de l'algorithme de $DTW$.

La comparaisons des séquences $A_{diff}$, entre l'alignement effectué à l'aide de la fonction $dist()$ et celui effectué à l'aide de la fonction $dist_2()$ est présentée figure 3.14.

Nous remarquons que le suivi est amélioré grâce à la modification du modèle : la séquence $A_{diff}$ obtenue avec l'alignement effectué avec la fonction $dist_2()$ entre les échantillons 100 et 160 ainsi qu'entre les échantillons 180 et 220. Il reste par contre de grandes différences entre les échantillons 160 et 180.

Reprenons l'alignement effectué avec la fonction $dist_2()$ et comparons le avec l'alignement effectué avec la fonction $dist_3()$ qui effectue une régression linéaire (Fig. 3.15).

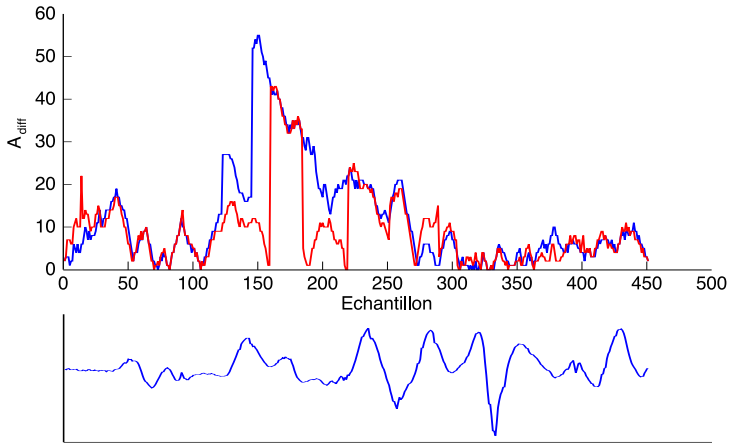


FIG. 3.14 – $A_{diff}$ de l'alignement obtenu à l'aide de la fonction $dist()$ en bleu et $dist_2()$ en rouge, ainsi que la fonction de référence en dessous en bleu.

Nous remarquons que le suivi est meilleur en utilisant la fonction $dist_3()$. Plus particulièrement au niveau des échantillons 160 à 190 où les deux précédentes méthodes faisaient une erreur conséquente sur l'état le plus probable. L'erreur est plus petite avec la méthode de la régression linéaire et il n'y a plus de sauts entre les différents états de la chaîne. Cette erreur peut donc s'interpréter comme un retard dans le suivi.

Reprenons l'alignement effectué avec la fonction $dist_3()$ et comparons le avec l'alignement effectué avec la fonction $dist_4()$ utilisant le $DTW$ local (Fig. 3.16).

Les deux alignements sont comparables. L'erreur entre les échantillons 160 et 190 est également évitée mais une nouvelle apparaît autour du $300^{eme}$ échantillon du signal de référence. En effet, à ce moment du suivi, la fonction d'observation utilisant le $DTW$ local a pu trouver une autre forme dans le signal de référence qui donnait une ressemblance forte avec le profil de notre accélération entrante sur sa fenêtre de 20 échantillon. Cela a provoqué un saut temporaire sur notre chaîne.

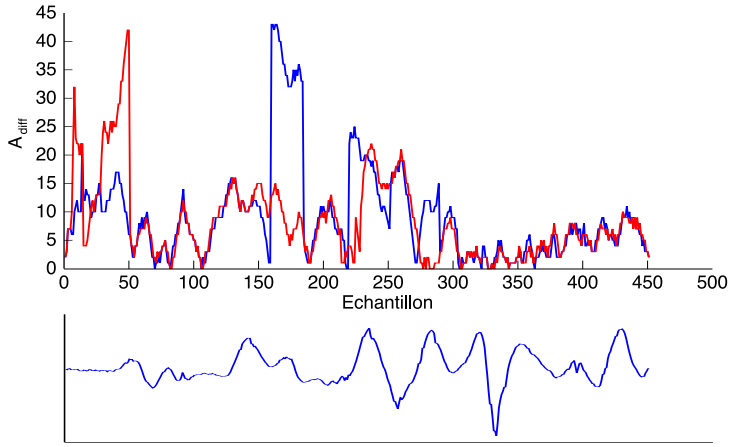


FIG. 3.15 – $A_{diff}$ de l'alignement obtenu à l'aide de la fonction $dist_2()$ en bleu et $dist_3()$ en rouge, ainsi que la fonction de référence en dessous en bleu.

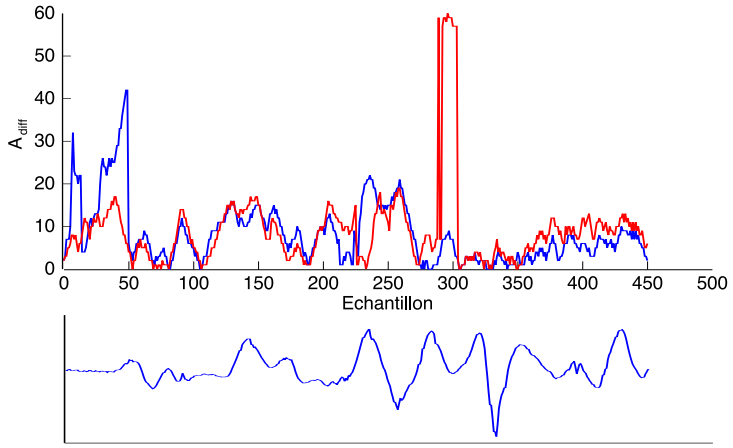


FIG. 3.16 – $A_{diff}$ de l'alignement obtenu à l'aide de la fonction $dist_3()$ en bleu et $dist_4()$ en rouge, ainsi que la fonction de référence en dessous en bleu.

### 3.2.4 Discussion des résultats

A partir de notre modélisation du bruit gestuel, nous avons apporté les modifications nécessaires au Modèle de Markov Caché afin d'améliorer le suivi du geste dans notre contexte. Les différentes améliorations apportées à la fonction d'observation améliorent le suivi du geste quand ce dernier présente de fortes variation dans son amplitude. Les calculs de distance prenant en compte le signal entrant sur une fenêtre $\tau$ sont rapides à mettre en place et offrent un suivi plus robuste s'adaptant à l'offset local constaté quelquefois dans les valeurs d'accélération de l'archet. La distance obtenue avec le $DTW$ local donne de bon résultats pour le suivi et une information riche sur les différences d'interprétation, mais son implémentation est lourde et son long temps d'exécution en font un mauvais candidat pour l'utilisation en temps-réel. Le meilleur compromis semble donc être la fonction d'observation utilisant la régression linéaire adapté aux modification linéaires d'intensité, qui consomme peut de ressource matérielle et donne de bons résultats lors du suivi.

La prise en compte des variations temporelles dans l'exécution d'un geste avec la méthode n°4 donne également de bon résultats sur nos signaux de synthèse. Il est à noter que la base construite à partir du signal entrant fenêtré à chaque instant $t$ n'est pas orthogonale ; il y a une redondance d'information dans les calculs de distance par rapport à l'ensemble des vecteurs de la base. Une meilleure prise en compte de la vitesse locale d'exécution d'un geste pourrai se traduire par une modification dynamique des coefficients de la matrice de transition de notre modèle. En effet, jusque là, les probabilités de transition d'un état à l'état suivant de la chaîne étaient toujours de 0.5. Si nous nous trouvons dans le cas d'une exécution plus rapide d'un geste, la probabilité de passer à l'état suivant doit être augmentée. Cette détection de modification de temps d'exécution peut être réalisée avec la méthode n°4. Ce travail ainsi que les tests de la méthode n°4 sur les signaux obtenus à partir du violon augmenté se ferons dans le second temps du stage.

# Chapitre 4

# Conclusion et perspectives

Nous avons tenté, tout au long de ce stage, d'apporter des pistes dans le cadre particulier du suivi du geste du violoniste. En effet, les contraintes que nous nous sommes fixées (suivi en temps réel, apprentissage du modèle réduit à son strict minimum, prise en compte de grandes variation dans l'interprétation d'un phrase musicale et quantification de ces variations) imposent quelques modifications dans l'utilisation de l'algorithme des Modèles de Markov Cachés, couramment utilisé dans le suivi du geste. Plus précisément, nous avons pris en compte ces modifications d'interprétation dans l'observation des différents signaux décrivant le geste, afin de ne garder que l'information "brute" du geste pour pouvoir faciliter le suivi. Cela permet, dans un second temps, de permettre de mettre de côté cette information "brute" afin de ne garder que l'information sur les différences d'interprétation.

Nous avons également mis au point une méthode d'évaluation du suivi au sein d'une interface, développée en Matlab, facilitant grandement l'implémentation et les tests des diverses modifications.

La modélisation des différences d'interprétation que nous avons proposé reste basique mais offre déjà des résultats intéressants, rendant le suivi plus robuste. L'amélioration du modèle passe par une meilleure prise en compte de ces différences. Cela peut se traduire par une évolution dans l'observation des signaux sur les différences d'échelles d'intensité et temporelle, entre le geste de référence et le geste à aligner, ainsi que par une modification dynamique du modèle en fonction des différences d'interprétation détectées.

# Remerciements

Ce stage a pu se dérouler dans un esprit de concertation et de partage des connaissances grâce à l'accueil bienveillant réservé par Norbert Schnell, responsable de l'équipe IMTR ainsi qu'au juste encadrement de Frédéric Bevilacqua, qui a su recadrer les orientations des différentes pistes de notre recherche, tout en me laissant jouir d'une autonomie certaine. Pour cela, je tiens à les en remercier.

# Bibliographie


[1] F. Bevilacqua, N. Rasamimanana, E. Fléty, S. Lemouton, and F. Baschet. The augmented violin project : research, composition and performance report. *Proc. NIME 06*, 2006.

[2] M. Demoucron, A. Askenfelt, and R. Caussé. Mesure de la "pression d'archet" des instruments à cordes frottées. In *Congrès Français d'Acoustique*, 2006.

[3] N. Rasamimanana. Gesture analysis of bow strokes using an augmented violin. Master's thesis, DEA ATIAM, 2004.

[4] F. Bevilacqua and E. Fléty. Captation et analyse du mouvement pour l'interaction entre danse et musique. Ircam.

[5] F. Bevilacqua, F. Guédy, N. Schnell, E. Fléty, and N. Leroy. Wireless sensor interface and gesture-follower for music pedagogy. *Proc. NIME 07*, 2007.

[6] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2) :257–286, 1989.

[7] J. Yang and Y. Xu. Hidden markov model for gesture recognition. 1994.

[8] A.D. Wilson and A.F. Bobick. Learning visual behavior for gesture analysis. *IEEE*, 1995.

[9] Y. Bengio and P. Frasconi. An input output hmm architecture.

[10] Aaron F. Bobick and Andrew D. Wilson. A state-based approach to the representation and recognition of gesture. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1997.

[11] A.D. Wilson and A.F. Bobick. Parametric hidden markov models for gesture recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1999.

[12] A.D. Wilson and A.F. Bobick. Nonlinear phmms for the interpretation of parametrized gesture. 1998.

[13] A.D. Wilson. *Adaptative Models for the Recognition of Human Gesture.* PhD thesis, Massachusetts Institute Of Technology, 2000.

[14] X. Ge and P. Smyth. Deformable markov model templates for time-series pattern matching. Technical report, Departement of Information and Computer Science, University of California, Irvine, 2000.

[15] X. Ge and P. Smyth. Hidden markov models for endpoint detection in plasma etch processes. Technical report, Departement of Information and Computer Science, University of California, Irvine, 2001.

[16] C. Decoux. Estimation de modèles de semi-chaînes de markov cachées par echantillonnage de gibbs. *Rev. Statistique Appliquée*, 1997.

[17] H. Lei and V. Govindaraju. Regression time warping for similarity measure of sequence. *IEEE*, 2004.

[18] Y. Yanagisawa and T. Saoh. Clustering multidimensional trajectories based on shape and velocity. *IEEE*, 2006.

[19] D. Chudova, S. Gaffney, and P. Smyth. Probabilistic models for joint clustering and time-warping of multidimensional curves.

[20] H. Kaprykowsky and X. Rodet. Globally optimal short-time dynamic time warping application to score to audio alignment. *IEEE*, 2006.