

Traitement de données pour capteurs embarqués

Cyrille Henry, Thierry Coduys

La kitchen
78, avenue de la république
75011 PARIS

Cyrille.Henry@la-kitchen.fr
Thierry.Coduys@la-kitchen.fr

***Résumé** : Les capteurs sont de plus en plus utilisés lors du processus de création artistique, afin d'enrichir les possibilités de création et d'interaction d'un acteur face à un dispositif électronique. Les interfaces permettant de connecter des capteurs embarqués sur un acteur à un logiciel de synthèse sonore ou visuel sont de plus en plus nombreuses. Dans cet article, nous décrirons différents types de traitement applicable de façon générique lors de l'utilisation de capteurs pour la synthèse audio en temps réel. Nous porterons une attention particulière avec l'exemple d'un percussionniste équipé de capteurs. Nous verrons qu'il est possible d'utiliser l'intention musicale pour piloter une synthèse audio-visuelle : il est possible de capter les mouvements du percussionniste, afin de piloter un son avant la frappe, pour annoncer celle-ci, afin de modifier et d'enrichir le discours musical. Enfin nous verrons qu'une méthode générique basée sur les réseaux de neurones permet l'analyse et la reconnaissance gestuelle en temps réel.*

1. Introduction

Le développement des interfaces de captation gestuelle a permis la généralisation des relations corporelles avec la musique lors de performances en temps réel. Cependant cette relation n'est pas évidente, puisqu'elle n'est en général pas directe, un minimum de traitement pouvant être appliqué au signal. Cette relation est en général non linéaire (la réaction n'est pas proportionnelle aux valeurs des capteurs). Le traitement permet une analyse plus ou moins profonde du signal, permettant ainsi de l'exploiter au mieux. Cet article présente différents traitements qui peuvent être utilisés. Nous verrons plus particulièrement différents cas génériques (calibration/filtrage) et des traitements plus spécifiquement adaptés à l'analyse de gestes d'un percussionniste. Enfin nous décrirons une méthode basée sur les réseaux neuronaux pour l'analyse et la reconnaissance gestuelle en temps réel.

2. Traitement de données pour la captation gestuelle

2.1 Description du problème

Nous décrivons ici une chaîne complète au niveau du traitement entre le capteur et un actionneur (synthèse audio, vidéo, autre), pour des applications temps réel. Le terme temps réel est employé ici de façon impropre pour signifier une réponse quasi immédiates (immédiates pour les sens humain). Certaines personnes préfèrent utiliser le terme **temps interactif**.

La relation entre le capteur et l'actionneur est rarement directe. Cette relation est la base de l'interaction : trop simple, elle n'apporte pas d'informations supplémentaires au spectateur. Trop complexe, le lien n'est pas perçu. Nous cherchons donc à extraire un maximum de données pertinentes du geste, afin de permettre une relation entre les capteurs et les actionneurs, la plus précise et répétable possible, afin de permettre à l'artiste de modeler l'interaction en fonction de son

discours artistique.

Pour cela, nous décomposerons la « boîte noire » en différentes parties :

- La calibration,
- L'extraction de données,
- L'analyse de ces données.

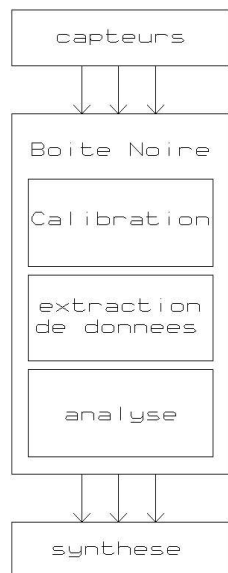


Figure 1 : la « boîte noire »

2.2 Dispositif nécessaire

Pour permettre une relation gestuelle naturelle et intuitive pour l'acteur, nous avons choisi de travailler à partir de capteur embarqué sur le corps. Afin d'avoir une précision et une résolution temporelle suffisantes pour le traitement des signaux, nous utilisons des interfaces de captation avec une précision de 10 bits minimum, les capteurs devant être actualisés à fréquence fixe, au moins toutes les 5 ms [Coduys 2003]. Le choix de capteur embarqué se justifie aussi ici : le prix du matériel de captation vidéo permettant de telles performances est prohibitif pour beaucoup d'applications. D'autre part, les caméras sont bien sur sensible uniquement à ce qu'elle « voient », ce qui engendre des limitations dans le jeu d'acteur pour le spectacle vivant par exemple. Ces limitations ne sont pas présente lors de l'utilisation de capteur embarqué : l'acteur peut être beaucoup plus mobile, voir même masqué du public etc. Nous utilisons le logiciel Pure data [Puckette 1996] (pd) comme environnement logiciel pour effectuer ce traitement. Notons que ce traitement est entièrement réalisable avec Max.

2.3 Calibration

La réponse de capteurs embarqués peut varier de manière significative en fonction de leur positionnement exact sur le corps mais aussi en fonction des lieux (exemple avec les capteurs magnétiques). Afin d'affiner la réponse du capteur, une calibration systématique peut être nécessaire. Cette étape permet une interaction précise avec le geste en toutes circonstances en établissant une bonne reproductibilité du traitement.

L'utilisation d'interfaces possédant une grande précision permet d'effectuer cette calibration de façon logiciel en repérant le minimum et le maximum du signal, afin de « mapper » ce signal entre deux bornes fixes. Cette calibration peut alors s'effectuer de façon extrêmement précise.

Dans le cas d'installation plastique où des capteurs sont utilisés pendant une longue période (plusieurs jours), leur réponse peut évoluer de façon significative, à cause des changements de température, d'humidité, etc. Il est parfois nécessaire de calibrer dynamiquement ces capteurs, en intégrant un système automatique permettant aux bornes extrêmes de la calibration de varier en fonction du temps.

2.4 Filtrage

Lors de l'utilisation de données bruitées, l'utilisation de filtres permet de réduire ce bruit. Cependant, un filtre engendre un retard, ce qui pose en général problème pour les applications en temps interactif. Afin de réduire au maximum cette latence, il convient d'adapter au mieux le filtre au signal provenant du capteur.

Par exemple, un capteur à ultrasons peut effectuer de fausses prédictions. Cela engendre des mesures aberrantes n'ayant aucun lien entre elles. Un filtre médian est bien adapté à ce genre de situations.

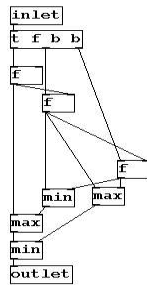


Figure 2 : exemple de filtre médian avec pd

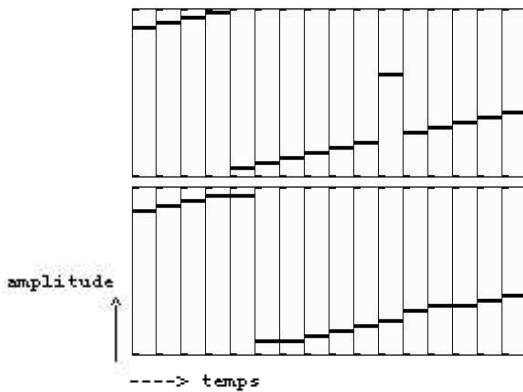


Figure 3 : réponse d'un filtre médian.

La figure 3 montre un exemple de sollicitation et la réponse d'un Médian (sur 3 échantillons) : on remarque que ce filtre a un retard correspondant à 1 échantillon. Les valeurs aberrantes sont supprimées. Un autre type de filtre couramment utilisé ce sont les filtres à réponse impulsionnelle infini (RII). Ils sont beaucoup utilisés à cause de leurs facilités à être implémenté.

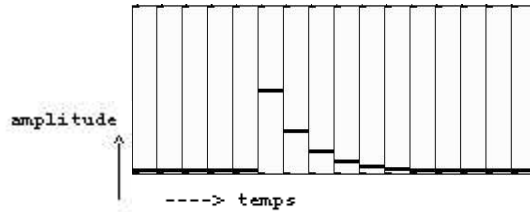
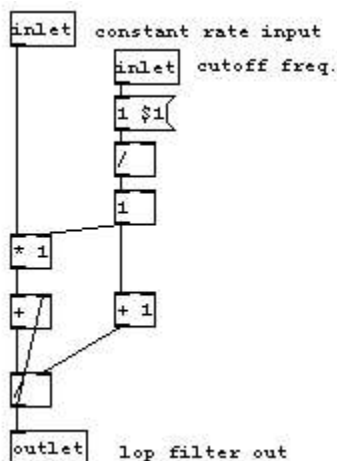


Figure 4 : exemple et réponse d'un filtre RII

3. Exemples de la percussion

La percussion est un instrument très intéressant à utiliser avec des capteurs, grâce à l'implication corporelle du musicien. Par ailleurs, la nature même du son percussif le rend très difficile à être traité en temps interactif : tout décalage supérieur à 10ms environ peut être perçu par l'oreille humaine pour ce type de son. Nous décrivons ici un moyen que nous avons mis en œuvre pour capter les gestes instrumentaux d'un percussionniste. Nous nous intéressons plus particulièrement à l'intention musicale : prédire la force de frappe avant que celle-ci n'ait eu lieu, afin d'adapter les traitements électroniques à la force de frappe. Nous verrons aussi comment doubler les sons acoustiques de la percussion par des sons électroniques, afin de modifier la perception sonore du son percussif.

3.1 Présentation du dispositif

Nous utiliserons des accéléromètres placés sur les baguettes du percussionniste. Ces capteurs analogiques sont bien adaptés à l'acquisition des gestes instrumentaux de la percussion. Ces capteurs sont branchés sur une interface de captation très précise (16 bits de résolutions). Cette interface est reliée en Ethernet sur un ordinateur afin de permettre un rafraîchissement complet de tous les capteurs en 5ms. L'ordinateur utilise un buffer audio de plusieurs millisecondes.

3.2 Résultats



Figure 5 : frappe simple

La figure 5 montre une frappe simple. La première

partie du graphique montre la préparation du geste, alors que le pic correspond au choc dû à la frappe instrumentale.

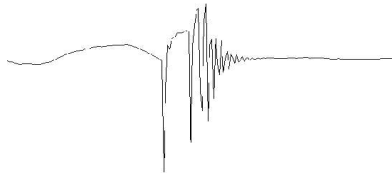
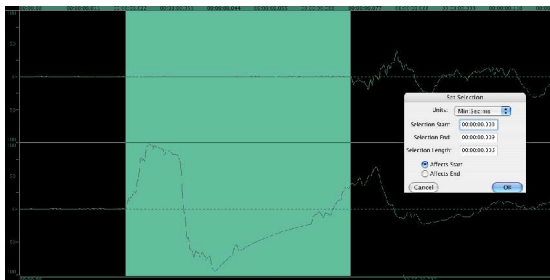


Figure 6 : frappe avec de multiples rebonds

La figure 6 montre de multiples rebonds, très rapides sur une membrane.

Les fréquences typiques des signaux provenant du choc et ceux correspondant aux gestes de l'instrumentiste sont très différents, il est aisé de séparer ces signaux par filtrage sans ajouter de



latence.

Figure 7 : enregistrement d'un son acoustique percussif et du son synthétisé

La figure 7 montre la différence de temps entre le son acoustique du jeu instrumental et le son synthétisé à partir des signaux captés par cette frappe. Une latence totale inférieure à 10ms peut être observée. Il est très difficile, même pour l'instrumentiste de discerner la latence du son électronique (latence due à la captation, l'analyse, la synthèse et la diffusion).

4. Reconnaissance gestuelle



basée sur les réseaux de neurones

4.1 Description

Les réseaux neuronaux sont généralement utilisés pour l'analyse et le traitement de signaux bruités, de signaux incomplets et pour l'interpolation. Nous avons implémenté dans pure data un réseau de neurone, pour plusieurs applications : l'analyse de système de capteur et la reconnaissance gestuelle. Les réseaux de neurones sont utilisés ici comme un des éléments de la « boîte noire », permettant de faire l'analyse et l'interprétation de signaux de captation en temps interactif.

Figure 8 : « boîte noire » basée sur un réseau de neurones

4.2 Analyse gestuelle

Lors de mouvements complexes, il est nécessaire d'utiliser plusieurs capteurs, pour arriver à connaître les déplacements d'un bras par rapport au corps. L'épaule étant une rotule pouvant librement tourner selon plusieurs axes, certains capteurs peuvent être liés à plusieurs axes de mouvement à la fois, sans réussir à séparer les différents degrés de liberté du système. De plus, la sortie désirée du

système n'est pas nécessairement en relation directe avec les coordonnées du mouvement. Il peut alors être intéressant d'utiliser un traitement intelligent, afin de faire correspondre à certaine position d'entrées, les résultats que l'on désire. Cela permet d'éviter d'avoir à définir directement la relation entre les données provenant des capteurs et les sorties de la « boîte noire », servant par exemple pour piloter une synthèse sonore. Les réseaux de neurones sont adaptés à ce genre de situation. Ils permettent d'adapter la sortie du système en fonction des paramètres d'entrées lors d'un processus d'apprentissage. De plus, ils peuvent interpoler entre les paramètres fournis au système lors de cette phase d'adaptation du système face au problème. Un autre intérêt des réseaux de neurones est leur robustesse face au bruit : il n'est pas possible d'effectuer exactement le geste plusieurs fois de suite. Une analyse basée sur les réseaux de neurones peut être fiable face à ce genre de situation [Fels 1993].

temps interactif des réseaux de neurone.

Références

[Coduys 2003] Coduys, T. and Henry, C. *Nouveaux matériels de captation haute précision* 10th 'Journée d'Informatique Musicale', Montbeliard, France, 2003.

[Fels 1993] Fels, S. S. and Hinton, G. E. *Glove-Talk: A neural network, interface between a data-glove and a speech synthesizer*. IEEE Trans. On Neural Networks, vol. 4, No. 1, 1993.

[Puckette 1996] Puckette, M. *Pure Data* International Computer Music Conference. San Francisco, 1996. <http://www-crcs.ucsd.edu/~msp/software.html>

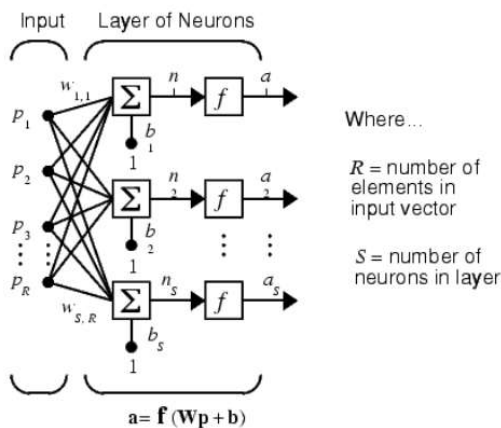


Figure 9 : structure d'un réseau de neurone.

4.3 Reconnaissance gestuelle

Un réseau de neurones peut aussi effectuer des reconnaissances de gestes. Pour cela, il est nécessaire de lui soumettre comme paramètres d'entrée, les valeurs des capteurs à différents instants (instant actuel mais aussi les valeurs provenant des 50ms précédentes environ). Concernant l'apprentissage, il est nécessaire de lui soumettre les gestes que l'on cherche à reconnaître, ainsi qu'une série conséquente de gestes non désirés, afin d'éviter les fausses détections.

Ces deux types de structure ont été implémentés dans pure data, afin de permettre une utilisation en