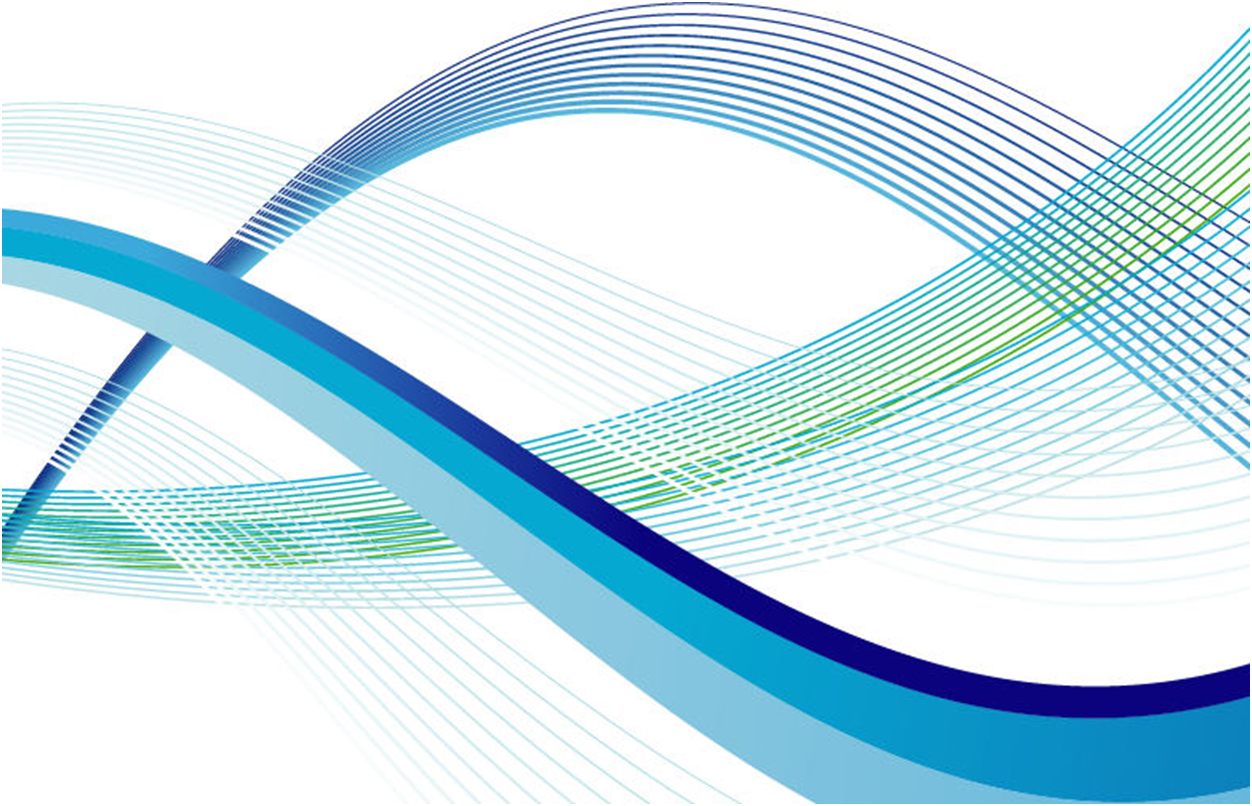
|  |
| --- |
| Polytech’ Nice Sophia Antipolis |
| Reconnaissance de mots |
| Conception d’une maquette d’un système simple de reconnaissance de mots |
|  |
| **Guénon Marie, Achard Jean-Paul, Favreau Jean-Dominique** |
| **24/1/2013** |

|  |
| --- |
|  |



Sommaire

[Introduction 3](#_Toc345505002)

[Plan prévu : 4](#_Toc345505003)

[Traitement du son 5](#_Toc345505004)

[1. Enregistrement du son 5](#_Toc345505005)

[2. Découpage du signal en temps 5](#_Toc345505010)

[3. Spectrogramme 6](#_Toc345505011)

[Fenêtre de Hamming 6](#_Toc345505012)

[Transformée de Fourier rapide 6](#_Toc345505013)

[Elimination des redondances 7](#_Toc345505014)

[Pondération des amplitudes 7](#_Toc345505015)

[Echelle de Mel et banc de filtre 7](#_Toc345505016)

[Reconstitution du spectrogramme 8](#_Toc345505017)

[Apprentissage et comparaison 10](#_Toc345505018)

[1. Ce qui existe **Error! Bookmark not defined.**](#_Toc345505019)

[Apprentissage 10](#_Toc345505020)

[Comparaison : Dynamic Time Warping 11](#_Toc345505021)

[2. Passage à la pratique 11](#_Toc345505022)

[Interface Homme Machine 12](#_Toc345505023)

[Conclusion 13](#_Toc345505024)

# Introduction

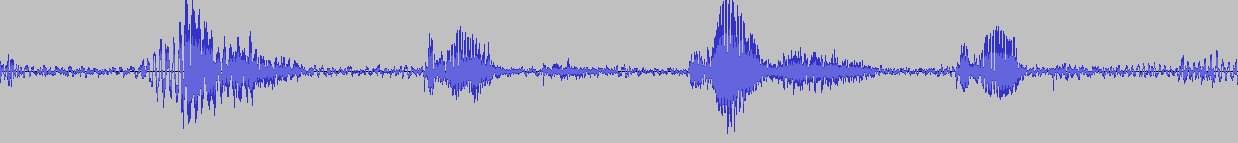
# Plan prévu :

* Traitement du son
  + Enregistrement du son (mots complets)
  + Auto-corrélation
  + Découpage du temps pas tranches de 20/30 ms, pour tout temps
  + Transformée de Fourier dans toutes les tranches et Echelonnage des fréquences : 20 points significatifs (Echelle de Mel : donne de l’importance à certaines fréquences)🡺 mis ensembles dans un spectrogramme (matrice)
* Apprentissage (problème de Bakis / Markov caché / Dynamic Time Warping🡪comparaison)
  + Pas forcement comparaison 1er avec le 1er, il peut y avoir des translations nécessaires ou des éléments ponctuels négligeables
  + Comparaison : « regarder le mot qui ressemble le plus », éventuellement si on a un grand vocabulaire commencer par un tri grossier  
    ex : 6/10  
    S-I-S 🡨 on supprime la partie identique  
    D-I-S  
    ↑on compare les parties pertinentes
* IHM
  + Interface graphique : afficher syllabe (et le spectrogramme)
  + « continuer » l’apprentissage : boutons « ok »/ « pas ok » et on rentre la syllabe dans la base
  + « Jeu » labyrinthe : on se dirige à la voix  
    / !\ apprentissage des mots à prononcer par le joueur pour le « jeu »

# Traitement du son

## Enregistrement du son

Avant même de traiter le signal sonore, il faut commencer par enregistrer le signal.  
Enregistrement du son lancé quand on attend une réponse et se poursuit *en continu ( ?)*, en enlevant les parties non intéressantes (sans paroles ou bruits)



## Découpage du signal en temps

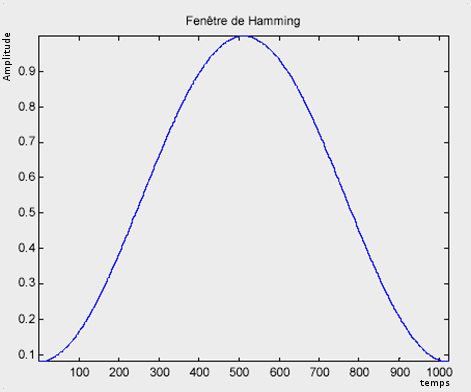
Découpage du temps pas tranches d’environ 30 ms :   
Pour tout temps t, on prend la tranche de signal qui s’étend de à.

Cependant, la transformée de Fourier rapide a besoin d’un nombre de points qui soit une puissance de 2 pour fonctionner beaucoup plus rapidement. Nous avons donc choisi de prendre des tranches de signal de taille n (avec n échantillons), où n est égal à une puissance de 2 et de telle manière que « n\*temps d’échantillonnage » soit le plus proche possible de 30 ms.

On peu voir clairement ici que la dernière tranche du signal ne sera pas nécessairement de la même taille que les autres échantillons, c’est pourquoi nous complétons la fin du signal avec des 0 et avoir ainsi un signal qui ait comme taille un multiple de la puissance de 2 considérée.

## Spectrogramme

### Fenêtre de Hamming

Une fois le signal découpé en tranches, nous lui appliquons une fenêtre de Hamming dans le but d’éviter les grosses discontinuités aux bords, et ainsi éviter d’obtenir des résultats incohérents après passage par la transformée de Fourier.

Formule de la fenêtre de Hamming :

Où T est la durée de la tranche de signal étudiée

### Transformée de Fourier rapide

Nous cherchons maintenant à appliquer à chaque tranche de notre signal la transformée de Fourier, qui s’écrit de la forme :

Où

Cependant, dans un but de rapidité, nous cherchons à utiliser la forme rapide de la transformée de Fourier, qui s’écrit :

Pour où

Pour

### Elimination des redondances

Toute transformée de Fourier est périodique et symétrique en 0. Etant donné qu’il n’est pas intéressant d’étudier deux fois la même chose (perte de temps), il suffit ici de ne conserver qu’une moitié du signal sur laquelle seront ensuite effectuées les comparaisons.

### Pondération des amplitudes

Nous cherchons ici à donner la même importance à toutes les fréquences, et éviter ainsi de donner trop d’importances aux pics présents dans les basses fréquences. Pour cela on utilise la formule suivante :

Echelle logarithmique pour Pondération pour réduire l’importance  
réduire la dynamique des basses fréquences

Où

### Echelle de Mel et banc de filtre

#### Echelle de Mel :

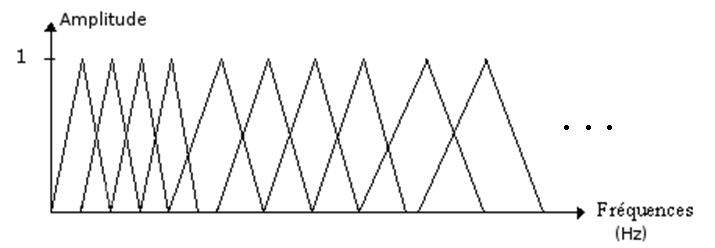
L’échelle de Mel a pour but de réduire l’importance des hautes fréquences qui sont moins finement perçues par l’oreille humaine. En effet, l'intensité de la perception d'un stimulus n'augmente pas linéairement en fonction de sa puissance mais de façon exponentielle.  
L’échelle de Mel permet donc de passer de la fréquence du signal d’entrée (en Hz) à une fréquence (en Mel) plus représentative de l’audition humaine et en suivant la formule :

Où F est la fréquence du signal d’entrée au point considéré.

Ceci donne la courbe suivante :

#### Banc de filtre :

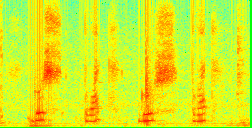
Le banc de filtre sert quant à lui à réduire le nombre de fréquences considérées à 20 valeurs possibles tout en respectant l’échelle de Mel.  
Pourtant l'utilisation de cette unité n'est pas suffisante. Pour avoir une largeur de bande relative qui reste constante, le banc de filtres est construit à partir de filtres triangulaires positionnés uniformément sur l'échelle Mel donc non uniformément sur l'échelle fréquentielle.  
Ce qui nous donne la courbe suivante :



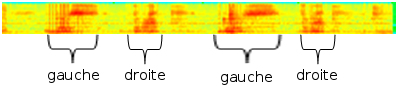
### Reconstitution du spectrogramme

La reconstitution du spectrogramme consiste à regrouper dans une matrice l’intégralité des transformées de Fourier obtenues et modifiées pour chaque instant du signal. Les lignes de la matrice représentent l’évolution en fréquence, les colonnes l’évolution en temps, et la valeur de chaque case représente l’amplitude de la transformée de Fourier au temps et à la fréquence donnée.

Spectrogramme obtenu avant filtrage par l’échelle de Mel :



Le même spectrogramme obtenu après filtrage par l’échelle de Mel :



Nous pouvons facilement voir ici la correspondance entre les deux spectrogrammes. De plus, nous pouvons voir que deux mots identiques on bien la même allure sur le spectrogramme.

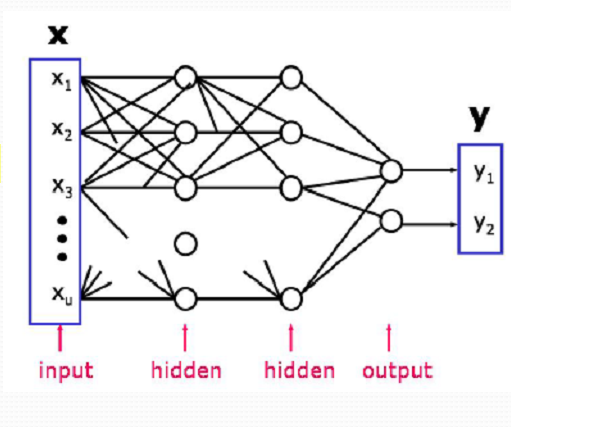
# Apprentissage et comparaison

## Les méthodes existantes

### Apprentissage

#### Réseaux de neurones artificiels

Un réseau de neurones est en général composé d'une succession de couches. Chaque couche est composée de Ni neurones, prenant leurs entrées sur les Ni-1 neurones de la couche précédente.  
À chaque [synapse](http://fr.wikipedia.org/wiki/Synapse) (connexion entre deux neurones) est associée un poids synaptique, de sorte que les Ni-1 sont multipliés par ce poids, puis additionnés par les neurones de niveau i.   
Mettre l'une derrière l'autre les différentes couches d'un réseau de neurones reviendrait à mettre en cascade plusieurs matrices de transformation et pourrait se ramener à une seule matrice, produit des autres.

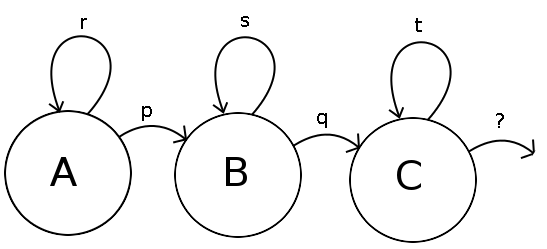


Pour ce système d’apprentissage, l’utilisateur n’a pas besoin d’adapter sa parole et il n’est pas non plus nécessaire d’entrainer spécifiquement le réseau de neurone à la voix de l’utilisateur.  
Cependant cette méthode nécessite grande base apprentissage représentative des cas qui seront rencontrés. De plus, dans le cas des éléments extrêmes, ils altèrent les poids donnés à chaque possibilité, ce qui rend le découpage de l’espace des possibilités instable.

Dans notre cas, nous avons un petit vocabulaire (d’une vingtaine de mots) basé sur un système mono-locuteur (on considère que lors d’une phase d’enregistrement une seul personne est amenée à parler). C’est pourquoi il n’est pas nécessaire d’utiliser ici des réseaux de neurones qui sont de plus lourds à mettre en place (d’un point de calculatoire) et il nous suffira de modifier la base de comparaison à chaque nouvel utilisateur.

#### Modèle de Markov caché

Une chaine de Markov est un automate contenant un certain nombre d’états, et pouvant à chaque instant passer d’un état à un autre avec une certaine probabilité.  
Sur notre exemple, nous avons 3 états : A, B et C. Si à l’instant i, l’automate est dans l’état B, il a alors la possibilité de passer à l’état C avec une probabilité q et de rester dans l’état B avec une probabilité s. (Notons ici que r+q=1 ; s+q=1 et t+ ?=1)   
Dans notre cas, nous avons supprimé les possibilités de retour en arrière car le temps est linéaire et A, B et C sont considérés comme des états successifs.



Le problème ici est de déterminer les différents coefficients de passage d’un état à un autre (p,q,r,s,t), pour ensuite pouvoir appliquer cet algorithme librement à la reconnaissance d’un mot.

Cet entrainement peut être basé sur l’algorithme forward-backward :   
Il commence par calculer l'ensemble des probabilités « en avant », qui donnent la probabilité d'obtenir k premières observations dans une séquence donnée, en terminant dans chaque état possible du modèle de Markov. Il calcule ensuite l'ensemble de probabilités « en arrière », qui représente la probabilité d'obtenir les autres observations étant donné un état initial.  
Ces deux ensembles de probabilités peuvent être combinées pour obtenir la probabilité d'être dans chaque état à un temps donné pendant l'observation de la séquence.

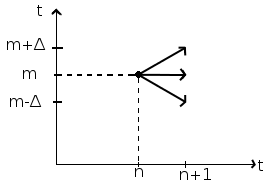
Cependant, il n’est pas intéressant dans notre cas d’utiliser le modèle de Markov caché, car il est trop complexe pour le petit vocabulaire que nous allons utiliser. Il nous suffit d’utiliser le Dynamic Time Warping qui est une simplification du modèle de Markov caché.

### Comparaison : Dynamic Time Warping (Déformation dynamique temporelle ou DTW)

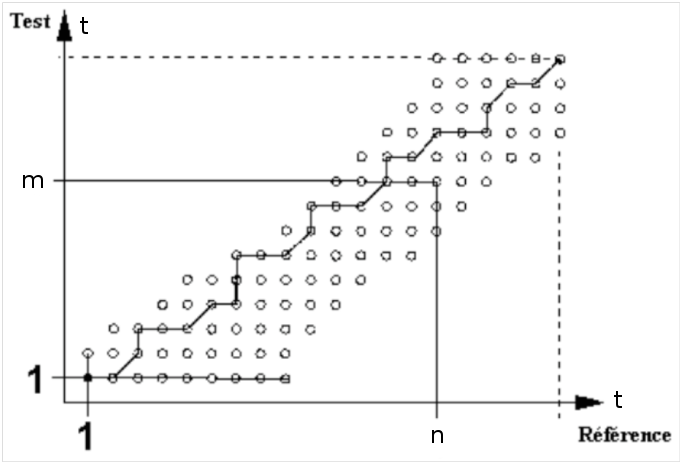
Le principe du DTW est de déterminer pour chaque élément d'une séquence, le meilleur élément correspondant dans l'autre séquence.  
Dans notre cas, il s'agit de comparer, à l'aide d'une distance euclidienne, les spectres d'un signal mesuré avec ceux des différents mots composant notre vocabulaire.

Principe de la comparaison :   
A chaque itération, nous comparons le nième spectre du signal mesuré avec les spectres des références de m- à m+ et on gardera le mième spectre le plus proche.

Le déplacement est donc de la forme :



Où les flèches représentent les déplacements possibles.  
Une fois tous les déplacements successifs mis ensemble, nous obtenons le chemin minimal entre les deux séquences. Voici un exemple de représentation du chemin le plus court entre une séquence testée et la séquence de référence :



Une fois la distance avec chaque mot établie, il suffit de récupérer la plus petite d’entre elles qui correspondra au mot de référence détecté.

## Passage à la pratique

# Interface Homme Machine

# Conclusion