|  |
| --- |
| Ecole Pratique des Hautes Etudes Commerciales Avenue du Ciseau 15 1348 Louvain-la-Neuve Technologie de l’informatique  ephec_660.png |
| Développement d'une application interactive homme-machine par reconnaissance et synthèse vocale |
| Rapport du travail de fin d’études |
| Pétroons Jérôme  3TL1 |
|  |
| **Année 2009-2010** |

Professeur : P. Ernotte

Année académique 2009-2010

Louvain-la-Neuve, le 25 aout 2010

# Remerciements

Avant de vous présenter mon rapport du travail de fin d’études, je désire remercier mon promoteur de stage, Mr. Ernotte, pour sa patience et pour m’avoir consacré du temps. Mais aussi tout particulièrement pour la rapidité de réponse à mes diverses questions ainsi que pour la rapidité de correction des diverses esquisses présentées.

Table des matières

[Remerciements 1](#_Toc270491248)

[Introduction – Définition du sujet 4](#_Toc270491249)

[La parole 5](#_Toc270491250)

[Le son 5](#_Toc270491251)

[Caractéristiques du son et de l’audition humaine 6](#_Toc270491252)

[La hauteur du son 6](#_Toc270491253)

[Intensité du son 6](#_Toc270491254)

[Durée du signal sonore 6](#_Toc270491255)

[Timbre du son 7](#_Toc270491256)

[La voix humaine 8](#_Toc270491257)

[Phonétique et phonologie 9](#_Toc270491258)

[Reconnaissance vocale 11](#_Toc270491259)

[Historique 11](#_Toc270491260)

[Techniques de reconnaissance vocale 11](#_Toc270491261)

[L’approche analytique 12](#_Toc270491262)

[Traitement du signal 12](#_Toc270491263)

[Méthode d’identification 12](#_Toc270491264)

[La méthode LPC 12](#_Toc270491265)

[Introduction 12](#_Toc270491266)

[Principe général 13](#_Toc270491267)

[Analyse LPC 14](#_Toc270491268)

[Les contraintes 16](#_Toc270491269)

[Le dictionnaire de mots 16](#_Toc270491270)

[Reconnaissance mono locuteur / multi locuteur 16](#_Toc270491271)

[Influence du micro dans la reconnaissance vocale 16](#_Toc270491272)

[Milieu sonore ambiant 17](#_Toc270491273)

[Comparaison dynamique 18](#_Toc270491274)

[L’algorithme DTW 18](#_Toc270491275)

[Mise en application 20](#_Toc270491276)

[Acquisition de la parole 20](#_Toc270491277)

[Acquisition des coefficients LPC 20](#_Toc270491278)

[Initialisation du système 21](#_Toc270491279)

[Apprentissage du système 22](#_Toc270491280)

[Reconnaissance d’un mot 22](#_Toc270491281)

[Estimation du signal acquis 23](#_Toc270491282)

[Synthèse vocale 24](#_Toc270491283)

[Historique 24](#_Toc270491284)

[Techniques de synthèse vocale 24](#_Toc270491285)

[Synthèse à partir d’un texte écrit 25](#_Toc270491286)

[Contraintes 25](#_Toc270491287)

[Mise en application 26](#_Toc270491288)

[Création des segments de la parole 26](#_Toc270491289)

[Programmation sous MATLAB 26](#_Toc270491290)

[Améliorations à apporter par la suite 29](#_Toc270491291)

[Reconnaissance vocale 29](#_Toc270491292)

[Synthèse vocale 29](#_Toc270491293)

[Conclusion 30](#_Toc270491294)

[Bibliographie 31](#_Toc270491295)

[Annexes](#_Toc270491296)

[Phonèmes du français](#_Toc270491297)

[Correspondance entre les noms employés pour MATLAB et les phonèmes](#_Toc270491298)

[Organisation de l’application](#_Toc270491299)

# Introduction – Définition du sujet

Dans la deuxième moitié du vingtième siècle, l’homme s'est assez vite intéressé au problème de la reconnaissance vocale dans le but d'améliorer la communication l’humain et la machine.

Dans le cadre de la dernière année de mon Baccalauréat en Technologie de l’Informatique à l’Ecole Pratique des Hautes Etudes Commerciales (EPHEC), j’ai choisi de m’intéresser à ce problème. Le sujet de mon travail de fin d’étude porte sur la réalisation d’une interface vocale entre l’homme et la machine. En effet, le but de mon travail est d’élaborer une interface appelée plus communément « Interface Homme-Machin » (IHM) par l’intermédiaire de la voix.

Un exemple assez connu du grand public d’une telle interaction sont les applications permettant de transformer un signal sonore en un texte écrit numérique ou également de permettre le processus inverse.

Dans le cas précis de ce travail, l’objectif est de concevoir une interface machine qui permettra à une personne d’interagir vocalement avec son environnement. Cette application permettra notamment à des personnes invalides ou à mobilité réduite de pouvoir interagir avec leur milieu environnant sans avoir recours à un quelconque déplacement physique.

On peut aisément illustrer cette interaction ainsi que son utilité avec l’exemple suivant. Prenons le cas d’une personne souhaitant effectuer une tâche quotidienne des plus simples comme allumer une lampe. Une action que quiconque effectue des nombreuses fois par jour sans que cela pose un quelconque. Mais cette simple action de tous les jours peut très vite devenir une épreuve pour certaines personnes à mobilité réduite. D’où l’utilité de concevoir une solution plus pratique et naturel.

Cette solution serait donc de permettre d’effectuer cette action, non pas avec leurs membres et donc de devoir se déplacer, mais bien avec leur voix afin d’éviter au maximum tout effort physique.

Il serait également intéressant de compléter cette interaction en offrant à l’utilisateur une réponse vocale de la part de l’ordinateur. Que ce soit pour informer la personne de la bonne compréhension de l’action demandée ou dans le cas contraire de demander de répéter l’action souhaitée. Cette réponse nécessitera donc une notion supplémentaire, un synthétiseur vocal. Celle-ci étant d’ailleurs utilisée par certaines applications afin de transformer un texte écrit en un fichier sonore.

# La parole

La reconnaissance vocale sera donc la technique informatique qui permettra d’analyser la phrase d’instruction énoncée par l’utilisateur et captée au moyen d’un microphone. Elle sera ensuite retranscrite en une instruction compréhensible et exploitable par la machine. Ensuite, celle-ci répondra de manière vocale pour signaler la bonne ou mauvaise compréhension de l’instruction au moyen de la synthèse vocale.

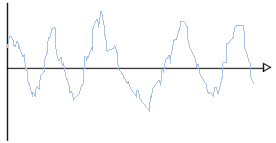
Ces notions de reconnaissance vocale et de synthèse vocale feront appel au domaine du traitement de signal. Mais particulièrement aux notions du son et de la parole, il n’est donc pas inutile de tout d’abord redéfinir celles-ci.

Dans le chapitre qui suit, je vais donc développer les notions physiques nécessaires à la bonne compréhension des futurs concepts utilisés lors d’une reconnaissance et d’une synthèse vocale.

## Le son

Première notion primordiale, le son. Le son est une vibration de l'air, c'est-à-dire une suite de surpressions et de dépressions de l'air par rapport à une moyenne, qui est la pression atmosphérique. C’est pourquoi une source sonore qui serait placée dans une cloche à vide n’émettrait plus un seul son, car il n’est plus entouré d’air. Pour exister, le son nécessite donc bien d’une vibration mécanique créant une onde se propageant sous forme d’une variation de pression dans un environnement compressible tel que l’air.

Ces ondes sonores peuvent être représentées sur un graphique comme les variations de la pression de l’air en fonction du temps. On obtient alors une représentation de la forme suivante :

t

## Caractéristiques du son et de l’audition humaine

Le son possède toute une série de caractéristiques qu’il sera important de prendre en compte dans la suite, tel que sa hauteur, son intensité, sa durée ainsi que son timbre.

### La hauteur du son

La hauteur d’un son correspond entre autres à sa vitesse de vibration. Il s’agit, en termes scientifiques, de la fréquence sonore (nombre de vibrations périodiques par seconde) que l’on exprime en hertz (Hz). C’est la fréquence qui permet de distinguer un son grave d’un son aigu. Les sons graves auront des fréquences basses (vibrations lentes) tandis que les sons aigus auront des fréquences élevées (vibrations rapides).

L’oreille humaine ne perçoit les sons que dans une certaine plage de fréquences située environ entre 16Hz et de 20kHz. Cependant, on estime que l’information phonétique se situe en réalité en dessous des 10kHz. C’est pour cela que dans le cas d’une communication téléphonique, les fréquences au-dessus de 3,5kHz sont coupées, mais la conversation reste compréhensible bien qu’un son très aigu tel que la prononciation du [S] soit toutefois mal rendu suite au filtrage.

### Intensité du son

L'intensité d'un son est la caractéristique permettant de distinguer un son fort d'un son faible.  Il s'agit, en termes scientifiques de l’amplitude de la vibration, qui se mesure en décibels (dB).

L’oreille humaine a également des bornes limites tout comme pour la fréquence. Si cette amplitude n’atteint pas une valeur minimale, appelée « seuil d’audibilité », le son ne sera pas perçu. Mais si au contraire, l’amplitude des vibrations sonores atteint une amplitude maximale cela devient extrêmement pénible pour l’oreille humaine, cela correspondant à la limite supportable ou « seuil de douleur ».

### Durée du signal sonore

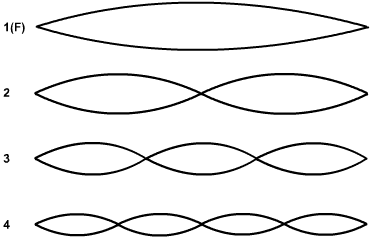
Comme tous les phénomènes perçus, le temps  joue lui aussi un rôle fondamental. Un son possède une certaine durée exprimée en seconde et correspond au laps de temps pendant lequel on perçoit ses vibrations.

### Timbre du son

Le timbre d'un son est en quelque sorte la couleur propre de ce son. Il varie en fonction de la source sonore, et ceci, indépendamment des trois premières caractéristiques. En effet, deux sons peuvent avoir la même fréquence fondamentale et la même intensité, mais ne peuvent jamais avoir le même timbre.

À identiques, les sons émis par deux instruments différents ne résonneront pas de la même manière. Cela se traduit par le fait qu’aucun son naturel n’est réellement simple. Il résulte de la combinaison d’un son fondamental, qui fixe la fréquence perçue par l’oreille et d’un grand nombre de ses harmoniques dont les pondérations relatives déterminent son timbre.

Lorsque l’on parle de fréquence fondamentale, on parle de la fréquence du premier harmonique du son considéré, que l’on désigne comme harmonique 1 ou harmonique fondamental. Les harmoniques naturels d’une note sont données par les fréquences multiples de la fondamentale.



Représentation de la fréquence fondamentale et ses harmoniques

## La voix humaine

Seconde notion primordiale, la voix humaine. La voix humaine est tout simplement l’ensemble des sons produits par le frottement entre l’air expiré par les poumons sur les replis du larynx. En effet, les poumons produisent un flux d’air suffisant permettant la vibration des cordes vocales. C’est cette vibration qui module très rapidement le débit d’air et qui crée un son.

Les muscles du larynx permettent en réalité d’ajuster la fréquence et dans une certaine mesure le timbre, c’est-à-dire le contenu harmonique, en enveloppant les cordes vocales. Celles-ci sont des muscles qui sous la pression pulmonaire forment un chenal appelé la « glotte » par laquelle s’évacue l’air vers les organes supérieurs. Ensuite la cavité buccale fournira l’articulation à l’aide de la langue, du palais et des lèvres.

|  |  |
| --- | --- |
|  |  |

Schémas représentant les organes vocaux ©Hachette

Deux types de sons peuvent résulter de ces étapes. Soit la glotte se resserre et l’air emprunte alors un chenal réduit mais ouvert ne mettant pas en action les cordes vocales ce qui produira des sons dits sourds ou non-voisés. Par opposition, dans le cas des sons dits sonores ou voisés, l’air doit écarter les parois de la glotte pour se frayer un passage, ce qui aura pour effet d’entrainer la vibration des cordes vocales et de produire un son.

Il est ensuite traité de manière différente suivant les cavités par lesquelles il passe. En effet, ces cavités agissent comme des résonateurs qui renforcent pour certains phonèmes les fréquences correspondant à leur fréquence de résonance. Ces fréquences renforcées sont appelées les « formants », et ce sont eux que l’on essaye de repérer dans un spectrogramme[[1]](#footnote-1) pour reconnaitre les phonèmes prononcés.

## Phonétique et phonologie

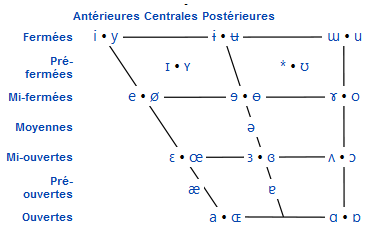
La phonétique est une branche de la linguistique qui étudie les sons utilisés dans la communication verbale. A la différence de la phonologie, qui elle, étudie comment sont agencés les phonèmes. Ces derniers sont souvent définis comme la plus petite unité phonique distinctive, c’est-à-dire celle qui permet de distinguer des mots les uns des autres.

Par exemple, l’opposition des mots « pain » et « bain » montre que [p] et [b] sont des phonèmes pour la langue français. Je précise, pour la langue française, car en effet cette notion de phonèmes est relative à une langue. Dans d’autres langues, comme l’espagnol latino-américain certains phonèmes de notre langue n’en seront pas pour eux, tel que [s] et [z] mais par contre [r] et [rr] en seront alors qu’ils n’en sont pas chez nous.

On peut classer les phonèmes en deux grandes catégories, les voyelles et les consonnes. (La liste des phonèmes de la langue française se trouve en annexe)

Les voyelles, généralement des sons voisés, sont des sons a durée assez longue et sont caractérisés par une assez grande constance de leurs propriétés fréquentielles dans le temps et peut être considéré comme périodique. La fréquence de vibration des cordes vocales détermine la fréquence fondamentale de la voix et est appelé le « pitch ».

Leurs formants sont horizontaux sur un spectrogramme. On classe souvent les voyelles au moyen d’un trapèze vocalique qui représente leur position dans un plan défini par les deux premiers formants et traduit la position de la langue lors de l’articulation.



Trapèze vocalique des voyelles

Tandis que les consonnes sont des phonèmes qui sont généralement beaucoup plus courts que les voyelles et beaucoup plus variable dans le temps, voir même chaotique. Elles peuvent être bruyantes ou sonantes, ce n’est que dans ce dernier cas qu’elles représentent des formants.

Afin d’illustrer quelques-uns des concepts formulés précédemment, voici ci-dessous un spectrogramme de la syllabe [asa], soit, une voyelle répétée deux fois et une consonne.



Spectrogramme de la syllabe [asa]

On peut y voir assez clairement l’horizontalité des formants de la voyelle [a]. La consonne [s] quant à elle est un bruit très aigu situé au-dessus de 5 kHz et assez clairement dépourvu d’un quelconque formant.

# Reconnaissance vocale

## Historique

Comme précisé dans l’introduction, les travaux sur la reconnaissance vocale datent de la deuxième moitié du xxe siècle. Le premier système pouvant être considéré comme faisant de la reconnaissance vocale date de 1952.

Ce système électronique développé par Davis, Biddulph, et Balashek aux laboratoires Bell Labs était essentiellement composé de relais et ses performances se limitaient à reconnaître des chiffres isolés.

La recherche s'est ensuite considérablement accrue durant les années 1970 avec les travaux de Jelinek chez IBM (1972-1993). C’est la société Threshold Technologies qui fut la première à commercialiser en 1972 un système de reconnaissance d'une capacité de 32 mots, nommé le VIP100.

Aujourd'hui, la reconnaissance vocale est un domaine à forte croissance grâce à notamment la déferlante des systèmes embarqués.

## Techniques de reconnaissance vocale

Il existe deux principales techniques pour effectuer la reconnaissance des mots, l’une est dite « globale » et l’autre « analytique ».

Dans l’approche globale, l’unité utilisée pour la reconnaissance sera le « mot ». Cette méthode consiste à donner au système une acoustique de chacun des mots qu’il devra identifier par la suite. Cette méthode nécessite donc une phase d’apprentissage durant laquelle chacun des mots sera répété plusieurs fois. Cette technique a comme désavantage d’être limitée en nombre de mots.

L’approche analytique tente plutôt de détecter et d’identifier les  phonèmes  ainsi que les  syllabes. Elle a comme réel avantage par rapport à l’approche globale que les mots ne sont pas mémorisés dans leur intégralité mais bien comme des suites de phonèmes. Et de ce fait la comparaison se fera par phonème.

Au cours de mes recherches pour ce projet, j’avais tout d’abord commencé par travailler avec une approche globale sur le programme « MATLAB [[2]](#footnote-2) ». Cependant elle représentait plusieurs désavantages au niveau de l’efficacité de la reconnaissance. En effet, cette technique fonctionnait assez bien pour des très petits mots, tels que  « un, deux, trois, quatre,… ». Autrement dit, pratiquement que des mots à une syllabe. Une fois que je passais à des mots comportant plus d’une syllabe la reconnaissance devenait déjà bien plus approximative et le taux de réussite n’était alors plus suffisant que pour être pertinent. Lorsque par la suite, j’ai tenté de passer à un système multi-locuteur, les performances chutèrent de façon significative. C’est pourquoi j’ai dû abandonner cette approche pour passer à l’approche analytique.

## L’approche analytique

### Traitement du signal

Une fois que le son a été émis par le locuteur, il est capté par un microphone et sera ensuite numérisé à l’aide d’un convertisseur analogique-numérique.

Comme la voix humaine est constituée d’une multitude de sons, souvent répétitifs, il faut en tout premier lieu traiter le signal acoustique pour y effectuer un tri sur les informations disponibles et de n’en retenir que les plus pertinentes au point de vue phonétique. Comme dit précédemment, la parole étant caractérisée par une amplitude, une fréquence et une forme du signal sonore, il faudra être en mesure de quantifier ces trois données.

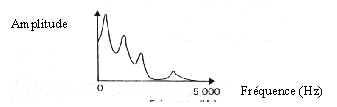
### Méthode d’identification

Elles sont fondées sur une connaissance des mécanismes de production telle que le conduit vocal. En effet, il a été dit que pour produire un son, l’homme utilisait son appareil respiratoire. Un groupe de méthode d’identification est fondé sur la connaissance de ce mécanisme de production. La plus utilisée est celle basée sur le « Codage Prédictif Linéaire », appelée LPC.

## La méthode LPC

### Introduction

L’hypothèse de base est que le canal buccal est constitué d’un tube cylindrique de section variable. L’ajustement des paramètres de ce modèle permet de déterminer à tout instant sa fonction de transfert qui fournit une approximation de l’enveloppe du spectre du signal à l’instant d’analyse.



Spectre lissé obtenu par prédiction linéaire (LPC)

On visualise alors aisément les fréquences des formants qui ne sont autres qu’en réalité les fréquences de résonance du conduit vocal. Celles-ci correspondent au « pic » d’amplitude dans le spectre.

La caractérisation des sons est importante dans le cadre de l’analyse LPC afin de savoir par le mode d’excitation et sa fréquence si le son est voisé.

### Principe général

Le LPC utilise le modèle source-filtre de la parole. Cette méthode se décompose en deux parties principales : l’analyse et l’encodage de la parole puis la synthèse de la parole ou le décodage.

Le codage LPC traite le signal de la parole en faisant une distinction entre les parties voisées et non voisées du son. Les parties voisées présentent une certaine périodicité, ce qui permet de trouver la fréquence fondamentale. Les parties non voisées n’étant aucunement périodiques, elles n’auront donc pas de fréquence fondamentale.

Les paramètres renvoyés par cette analyse sont la fréquence fondamentale (uniquement pour les parties voisées), un gain (G) et les coefficients d’un filtre tout pôle (filtre autorégressif AR). Ces paramètres seront quantifiés avant d’être envoyé sur un canal de transmission.

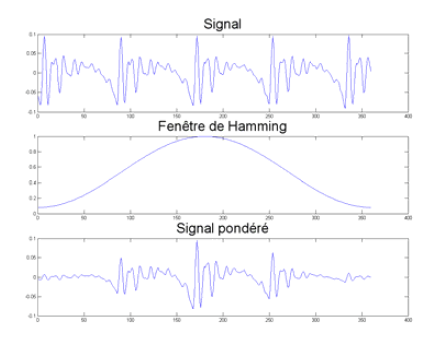
A l’autre bout du canal, on décode et on reconstitue le signal et passera ensuite par un filtre avec les paramètres précédemment déterminés.

### Analyse LPC

L’analyse à proprement parlé consiste en un découpage du signal en tranches de quelques dizaines de millisecondes. Si on prend en compte les limites de la perception humaine, on peut dire que le son ne varie pas pendant environ 20ms  et que donc l’analyse de la parole se fera sur des segments bornés du signal.

Une fois le signal découpé en tranches de 20ms, il est nécessaire de les chevauchées les unes aux autres en déterminant un décalage entre les débuts de deux tranches successives afin limiter les erreurs lors de la reconstitution du signal.

Ensuite chaque tranche de son va être pondérée par une fenêtre. Plusieurs choix de fenêtre sont possibles mais la fenêtre de Hamming va être retenue ici.



Signal, fenêtre de Hamming et signal pondéré par cette fenêtre.

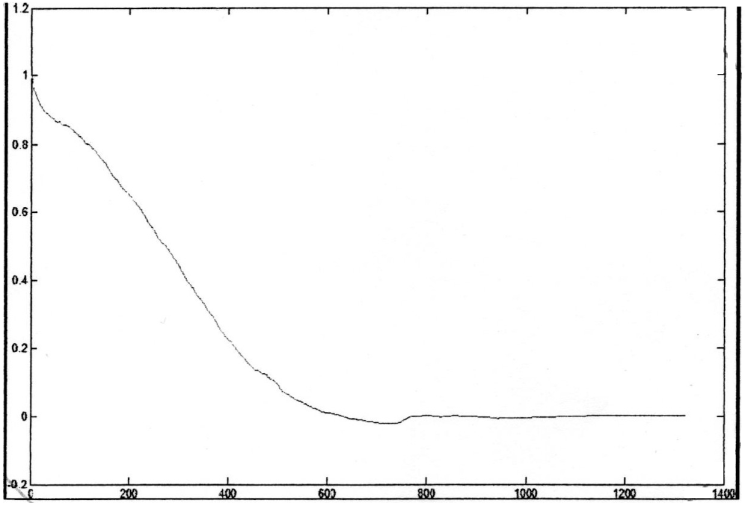
Le but d’une telle opération est de limiter les erreurs lors du calcul des coefficients. En effet, l’analyse LPC étant une analyse prédictive qui utilise de ce fait les valeurs entourant un échantillon pour le prédire et que les échantillons du début et de fin n’ont soit pas de prédécesseurs ou de successeurs, leur prédiction sera naturellement en partie erronée. D’où l’utilisation d’une fenêtre de Hamming nulle en ses deux bords pour diminuer l’influence de ces échantillons extrêmes.

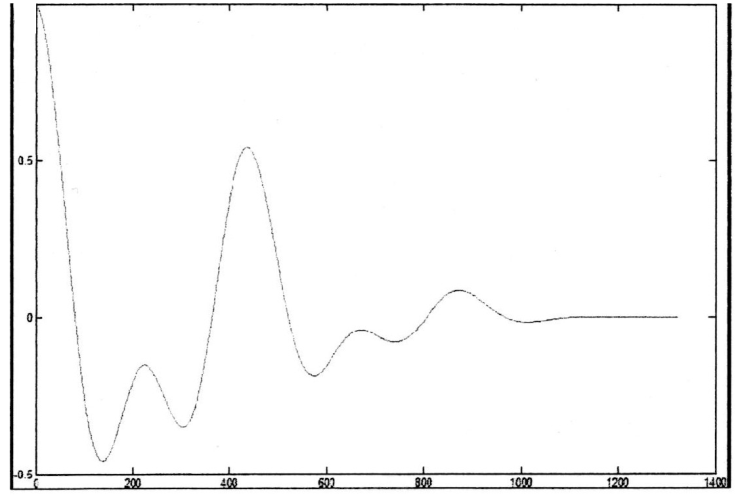
La distinction entre les sons voisés et non voisés se fera à l’aide de la méthode d’autocorrélation. La corrélation d’un signal avec un autre est la mesure de la similarité entre ces deux signaux. Pour cela on mesure l’énergie commune contenue dans ces deux signaux selon le décalage de l’un par rapport à l’autre. Plus les signaux se ressemblent et plus cette énergie sera grande et plus l’énergie du produit sera grande. Dans ce cas-ci on pratiquera une autocorrélation, c’est-à-dire une corrélation du signal avec lui-même.

On peut quantifier la ressemblance d’un signal ‘y’ par rapport à lui-même selon le décalage que l’on lui introduit  et dont voici l’équation :

En toute logique cette fonction a un maximum pour ‘p’ nul. En effet, quand le signal est en phase avec lui-même selon la fréquence prépondérante qui le constitue, l’énergie mutuelle sera naturellement maximale. Elle contient également des pics à chaque fois que le signal est décalé du nombre ‘p’ qui correspond en réalité au pitch fondamental ou à ses multiples.

Une courbe d’autocorrélation du signal définie par l’équation où « s » est le signal, permettra ensuite de définir si le son est voisé ou non :

  
  
 Si le son n’est pas voisé, comme sur le schéma ci-dessus, la courbe d’autocorrélation décroit sans pic jusque zéro.



Si le son est maintenant voisé, on peut observer des pics sur la courbe d’autocorrélation comme sur le schéma ci-dessus. Si après que la courbe ait traversé l’axe des abscisses et qu’il existe au moins un point dont la valeur est supérieure ou égale au tiers du maximum cela signifie que le son est voisé. Dans ce cas le pitch correspond à l’abscisse en fréquence du maximum de tous ces points.

## Les contraintes

### Le dictionnaire de mots

Le système de reconnaissance vocale aura besoin de posséder une librairie de mots auquel sera comparé le mot cité. Ne possédant pas de librairie de base pour mon système il est impératif de l’entrainer avant d'entamer toute forme de reconnaissance.

Dans ce principe, il faut que l'utilisateur énonce au moins une fois l'ensemble du vocabulaire à connaitre afin de créer un dictionnaire de références acoustiques. C'est ce dictionnaire qui va permettre une reconnaissance plus ou moins efficace suivant la façon dont la librairie est étoffée ou non.

### Reconnaissance mono locuteur / multi locuteur

Il faut tout d'abord savoir qu'un même locuteur ne prononce pas deux fois le même mot de la même façon. Les variations de la vitesse d'élocution ou du rythme de prononciation entraînent des transformations non-linéaires au sein du signal acoustique.

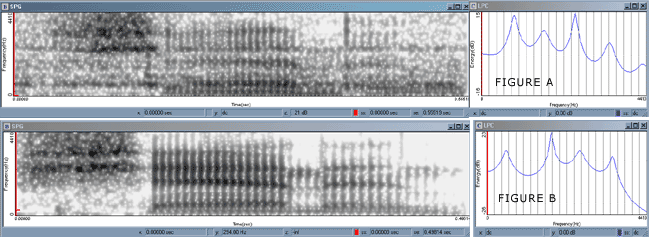
Durant l'apprentissage, le locuteur prononce les mots du dictionnaire. Il n’est donc pas inutile d'avoir plusieurs références de chaque mot, à cause de la variabilité intra-locuteur.

Pour la reconnaissance multi locuteur, la comparaison dynamique peut être appliquée. Il faut que chaque mot de vocabulaire du dictionnaire soit prononcé par un échantillon important d'individus ayant des particularités propres (timbres différents, sexe, accents...). Il semblerait que 8 à 12 représentants donnent une bonne image de l'ensemble des prononciations d'un mot.

### Influence du micro dans la reconnaissance vocale

La reconnaissance vocale se faisant par l'intermédiaire d'un micro, la qualité de celui-ci est de ce fait importante. Il est évident qu’au plus le micro restituera un son de qualité et avec le moins de bruit possible, au plus la reconnaissance vocale se fera correctement.

Pour simplement démontrer l'influence d'un micro, voici l'exemple suivant :



Les figures A et B montrent tous les deux un spectrogramme du même mot prononcé par la même personne. Le premier est enregistré avec un microphone sur pied et tandis que le second est enregistré avec un micro-casque.

Les résultats LPC correspondants montrent des fréquences de formants différentes et ainsi que des valeurs de bande passante différentes.

On aisément voir que les valeurs LPC obtenues sur la seconde figure sont plus précises. Il est donc primordial de s’assurer de la qualité de restitution du micro employé.

### Milieu sonore ambiant

Le milieu sonore ambiant est, quant à lui, directement issu des conditions d’enregistrement du signal audio. La présence d’équipements bruyants ou d’éléments sonores perturbateurs peuvent altérer le signal, et rendre son traitement plus délicat.

Dans ce cas, comme dans le cas du bruit affectant les fréquences audibles, il est difficile de traiter efficacement le signal pour éliminer ces perturbations. De nombreuses techniques de filtrage existent, en se basant notamment sur la nature de la source sonore, et permettent de limiter les perturbations, mais il est impossible de les supprimer complètement et de façon certaine.

La solution est donc, pour ces deux cas, d’effectuer l’enregistrement dans les meilleures conditions possibles afin que le signal enregistré soit le plus fidèle possible à l’original.

## Comparaison dynamique

La reconnaissance des mots se fera donc par comparaison entre les mots énoncés par l'humain et ceux contenus dans le dictionnaire de références acoustiques. Quand une personne prononcera un mot, le signal devra donc être paramétré pour ensuite être comparé aux mots du dictionnaire de références acoustiques.

Comme dit précédemment il faut garder toujours en tête qu'un même locuteur ne prononce jamais deux fois le même mot de la même manière. Il est donc impossible de comparer deux spectres directement car la durée, l’intensité, ou même le rythme ne sera jamais deux fois le même.

Il existe plusieurs techniques de comparaison, j’ai opté pour une comparaison de spectre à spectre mais de façon dynamique par l’algorithme DTW (Dynamic Time Warping).

### L’algorithme DTW

Cette méthode de comparaison a pour principal avantage de pouvoir donc comparer deux spectres malgré leurs différences inévitables par une sorte de normalisation temporelle. On peut distinguer deux sources de variation de l’échelle temporelle, c’est-à-dire, la variation du rythme de prononciation et de la variation de vitesse.

Le principe est le suivant, soient A et B, avec respectivement des longueurs I et J. La distance entre l’évènement de A et l’évènement de B se calcule avec une distance euclidienne :

avec : i = et

On considère naturellement la même plage de fréquence pour les deux signaux. On crée donc un chemin. Il est nécessaire que les fonctions et soient croissantes et doivent correspondre à certaines contraintes. Les seuls chemins valides arrivant au point sont ceux provenant des points et . De plus on prendra K tel que

La méthode consiste à choisir en réalité le chemin qui passe par les distances les plus petites de sort que la distance cumulée le long de ce chemin soit la plus petite possible. On définit la distance cumulée au point comme :

On remplit ensuite la matrice avec en ième et jème colonnes le résultat de. Enfin, on définit la distance normalisée entre les deux prononciations du mot :

On obtient bien une distance entre deux spectres. On effectue alors ce travail entre le mot à reconnaitre et tous les mots présents dans le dictionnaire. On choisira donc par la suite, le mot du dictionnaire qui aura la plus petite distance spectrale avec le mot à reconnaitre.



Chemin idéal entre deux spectres

**

Représentation de la notion de chemin entre les deux spectres. Les différences entre les deux spectres « tordent » le chemin idéal qui est la diagonale comme montré sur le schéma d’avant.

Maintenant toutes les notions théoriques nécessaires à l’application pratiques sont vues et donc je peux passer à la mise en application.

## Mise en application

Pour effectuer cette mise en pratique j’ai travaillé à la fois sur MATLAB et sur SIMULINK. Ce dernier étant une extension à MATLAB utile pour la modélisation, l’analyse et la simulation d’une variété de systèmes physiques et mathématiques. Il ne fonctionne pas par code, contrairement à MATLAB mais par blocs. Chaque bloc représentant une fonction complexe, une constante, un opérateur. On les assemble donc afin de créer facilement des algorithmes complexes.

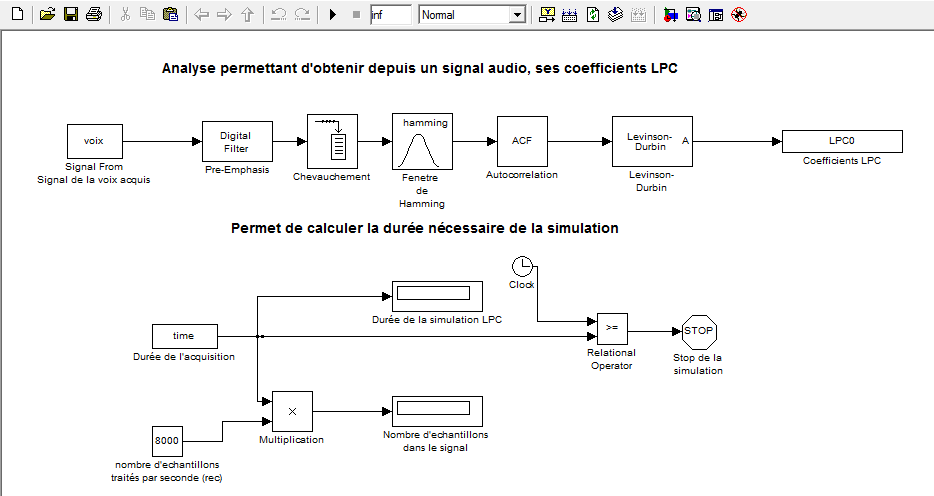
Les deux méthodes sur SIMULINK sont inspirées d’un ancien travail sur la reconnaissance vocale fait par un ancien élève, Mike DeMan. Toute la suite du projet est quant à elle totalement personnelle. J’ai notamment décidé de prendre une direction différente dans le système de comparaison des mots en utilisant par exemple la comparaison par l’algorithme DTW qui permet une comparaison sur de longs mots là où la sienne se limitait à des mots de maximum deux syllabes, ce qui était de toute évidence insuffisant pour mon application. Mais également dans une gestion différente des librairies, de la phase d’apprentissage et de nombreux autres points. Un autre ajout significatif n’est autre que la partie concernant la synthèse vocale.

### Acquisition de la parole

Comme expliqué dans la théorie, la première étape pour effectuer une reconnaissance vocale et d’effectuer une acquisition d’un signal sonore à l’aide d’un microphone. Pour cette opération j’ai donc utilisé SIMULINK. Un module ‘**acquisition\_voix.mdl’** est créé et contient des blocs assemblés de façon à acquérir le signal sonore, d’obtenir la durée de celui-ci par une simple pression sur le bouton <Play>. On stop l’enregistrement en appuyant alors sur le bouton <Stop>. Il est naturellement recommandé d’énoncer clairement et distinctement le mot à enregistrer et de stopper l’enregistrement directement à la de l’énonciation du mot.

### Acquisition des coefficients LPC

Notre son est donc désormais enregistré et sauvegardé à la fois sous la forme d’un son « .wav » mais aussi sous forme d’une matrice. La deuxième étape consiste désormais à effectuer une analyse sur ce signal afin d’en calculer les coefficients LPC comme expliqué dans la partie théorique. Cette partie s’effectue, toujours sur SIMULINK, avec le module ‘**traitement\_LPC.mdl’**.



La partie du haut est une succession de blocs couplés les uns aux autres. En entrée nous avons donc le signal acquis et en sortie nous auront les coefficients LPC de ce signal. Ce signal possède une fréquence d’échantillonnage de 8000 Hz et sera comme expliqué dans la théorie, divisé en trames de 20ms (160 samples) avec un chevauchement de 10ms (80 samples). Ces paramètres sont facilement définissables dans les propriétés de chaque bloc. Ensuite, chaque trame est ensuite fenêtrée par une fenêtre de Hamming qui est nulle en ses bords pour limiter les erreurs lors des calculs.

Une fois cela, les trames passent dans un algorithme d’autocorrélation où on définira un ordre de 12, valeur moyenne conseillée pour les voix d’hommes et de femmes. Tout cela dans le but de trouver les coefficients de réflexion en passant les coefficients AR dans l’algorithme de Levinson-Durbin qui est le bloc suivant. La sortie de ce bloc sera une matrice de taille n\*13, ‘n’ étant dépendant de la durée du signal d’entrée et 13 étant le nombre de coefficients LPC par fenêtre, soit l’ordre du modèle d’auto corrélation choisi plus 1.

La partie du bas sert à calculer le temps nécessaire à une simulation complète, soit l’équivalent de la durée du signal d’entrée. Une fois que le temps de la « clock » est donc égal à la durée du signal, la simulation s’arrête.

### Initialisation du système

Afin de faciliter l’utilisation de l’application j’ai créé une méthode ‘**initialisation\_application.m’** qui lance tous les modules et fonctions nécessaires au préalable pour le bon fonctionnement de l'application. Tel que ; lancer les deux modules SIMULINK, la création des librairies de phrases et autres « constantes » nécessaires.

### Apprentissage du système

Etape primordiale, l’apprentissage du système se fait en lançant la méthode **‘apprentissage\_systeme.m’**, celle-ci va afficher dans la console de MATLAB toute une série d’étapes à suivre par l’utilisateur afin d’entrainer le système. L’apprentissage peut se faire différemment mais j’ai voulu rendre cette phase plus accessible.

Durant cette phase, le système demandera donc pour un mot précis affiché d’effectuer les deux modules SIMULINK afin de calculer les coefficients LPC correspondants à ce mot. La liste de mots qui seront à faire apprendre est contenue dans cette méthode et peut être aisément extensible. Une fois le calcul des coefficients LPC effectué, ils seront sauvegardés individuellement sous la forme « action\_01.dat ». On fait cette suite d’opération pour chaque liste d’actions et chaque liste d’objets.

|  |
| --- |
|  |

Capture de la console MATLAB avec un exemple d’apprentissage du système

### Reconnaissance d’un mot

Pour faire une première reconnaissance vocale, il faut lancer la méthode ‘**reconnaissance.m’**. Là encore, une série d’étapes est affichée dans la console et requiert l’utilisation des deux modules SIMULINK. On capte donc l’action que l’on désire ainsi que l’objet concerné par cette action. Une fois c’est deux signaux captés et analysés, la méthode ‘**comparaison.m’** sera appelée. Elle effectue la comparaison entre toutes les matrices de coefficients LPC étalons contenus dans le système et les coefficients des sons à  reconnaitre afin de déterminer laquelle est la plus proche. Cette notion de distance est faite par la comparaison dynamique DTW et est effectuée par un appel de la fonction ‘**distance\_dtw.m’**. Une fois toutes les distances calculées, le système va considérer que le mot à reconnaitre est celui qui possède la distance la plus courte.

J’ai placé un seuil pour éviter une réponse non pertinente. En effet si je dis un mot qui n’est pas contenu dans le système, il y’aura néanmoins toujours un mot avec qui il aura une distance plus courte qu’avec les autres. Un seuil de 60 a donc été fixé, cette valeur a été établie suite à une batterie de tests qui ont montré qu’une distance entre deux même mots dépasse rarement les 50 et se situe en moyenne vers 45. Tandis qu’un mot avec un différent, la distance se situe plus dans les 70-80.

Une fois les deux mots reconnus, ils sont affichés dans la console et une réponse positive est envoyée vers la synthèse vocale. Si au contraire un mot n’est pas reconnu, des points d’interrogations seront affichés au bon endroit pour signaler quel mot n’a pas été reconnu et une réponse négative sera envoyée vers la synthèse vocale. Cette réponse déterminera quelle type de phrase sera dite vocalement.

### Estimation du signal acquis

Une dernière méthode, la méthode ‘**estimation\_signal\_acquis.m**’ permet d’estimer le signal avec les différents coefficients LPC obtenus. Pour cela les coefficients LPC sont placés comme dénominateurs d’un filtre et cette démarche est effectuée pour chaque trame de 160 samples (20ms). Ensuite un affichage du signal est effectué grâce à la méthode de Burg. Cette représentation du signal a pour intérêt d’offrir une visualisation des différents formants qui composent le mot. Ceux-ci étant représentés par les pics sur la courbe.

|  |  |
| --- | --- |
| Signal estimé du mot « Allumer » | Signal estimé du mot « Eteindre » |

# Synthèse vocale

## Historique

Il y’eu plusieurs générations de techniques de synthèse vocale qui furent proposées. La première d’entre elles fut proposée entre 1965 et 1985 et était appelée synthèse vocale par règles ou synthèse vocale par formants. Elle était, comme son nom l’indique, basée sur la modélisation du spectre sonore de la parole à partir des formants mesurés en Hz.

Des paramètres et des règles servaient à décrire l’évolution temporelle de ces formants ce qui permettait donc de générer le spectre sonore à partir duquel était généré le signal acoustique de synthèse. Cette technique qui était donc entièrement synthétique avait comme principal avantage d’être compatible avec la mémoire informatique limitée de l’époque.

La seconde génération fut semi-synthétique et fut utilisable grâce à notamment une plus grande mémoire informatique. Cette plus grande mémoire autorisait donc de stocker de façon permanente des durées suffisantes de parole naturelle. La technique ensuite consistait donc à assembler ensemble tous des petits segments élémentaires de la parole naturelle afin de constituer n’importe quel énoncé synthétique souhaité. C’est cette technique que je vais donc développer dans la suite de ce projet.

## Techniques de synthèse vocale

La première technique de cette génération semi-synthétique est la synthèse vocale par diphones. Comme dit précédemment, les sons synthétiques sont en réalité des segments d’enregistrement de la parole qui sont artificiellement attachés les uns à la suite de l’autre.

Il est montré qu’une base de 1200 diphones serait suffisante afin de reproduire la langue française, ce qui nécessite le stockage de 120 secondes de parole ce qui ne pose de nos jours plus aucun soucis pour nos machines. Avec cette technique la parole semble plus naturelle que celle produites par règles, cependant des problèmes apparaissent lorsque l’intonation et l’accent tonique mais aussi la longueur des phonèmes sont pris en compte.

Une solution possible pour remédier à ces problèmes est d’augmenter la quantité d’extrait sonore à utiliser et mis à disposition lors de la concaténation. En effet, plusieurs extraits pour un même diphone peuvent alors être utilisés dans un même contexte pour par exemple tenter de réduire la mauvaise transition entre deux diphones. Mais peuvent également être utilisés suivant le type de contexte comme l’intonation, l’accent ainsi que le type de phrase. On parle alors de la synthèse par sélection de diphones. Naturellement l’espace mémoire requis devient nettement plus imposant et des quelques mégaoctets que demandaient les 1200 diphones de base, on passe facilement à des bases de données bien plus lourdes pouvant atteindre plusieurs giga-octets.

## Synthèse à partir d’un texte écrit

Lorsque l’on souhaite convertir un texte écrit en une parole synthétisée, la première étape est de retranscrire ce texte en une séquence de phonèmes. Ceux-ci représentant exactement les sons qui doivent être prononcés.

Prenons l’exemple d’une phrase que mon système devra être à même de dire, si je veux retranscrire la phrase « La commande est comprise », celle-ci deviendra en phonétique :

***« /la/ /kɔ.mɑ̃d/ /ɛst/ /kɔ̃.pʁiz/ »***

Une fois cette conversion effectuée il ne restera donc plus qu’à associer ces phonèmes aux diphones correspondant et préalablement enregistrés et de concaténer ceux-ci l’un à la suite de l’autre.

## Contraintes

Naturellement, lors de cette retranscription il sera nécessaire d’appliquer une série de règles de réécritures dépendant du contexte aux suites de lettres composant le texte. La connaissance de la catégorie grammaticale de certains mots est dans ce cas nécessaire. Par exemple, le mot « président» ne se prononcera pas de la même manière et n’aura de ce fait pas les mêmes règles que pour « ils président ».

Mais il faut ajouter l’intonation. En effet l'intonation est une caractéristique importante en langue et notamment en français car là aussi les effets varient d’une langue à une autre. Elle représente le parcours mélodique de la voix pendant la prononciation de la phrase. Il faut donc pouvoir déterminer si la phrase est affirmative, impérative ou interrogative. De plus l'ordre des mots dans la phrase aura une influence car l'intonation dans la langue française est descendante en fin de phrase.

Dans tous les cas, la voix synthétique doit s'efforcer d'imiter l'intonation naturelle afin de donner une impression de voix naturelle. A l'inverse un énoncé synthétisé sans intonation est perçu comme anormal, et à la limite une voix synthétisée avec une fréquence fondamentale absolument constante, est perçue comme totalement artificielle avec une impression de voix de robot.

Et pour finir les notions de rythme et d’intensité sont importantes si on veut obtenir une voix la plus naturelle possible. Le rythme doit donc être naturel, ce qui signifie une durée naturelle des phonèmes. Quant à l’intensité, qui correspond au volume sonore, elle est beaucoup moins importante que l’intonation et le rythme pour obtenir un rendu naturel mais il est toujours bon de savoir qu’elle peut également y contribuer.

## Mise en application

Pour réaliser cette partie j’ai également utilisé le programme MATLAB. Comme j’ai expliqué précédemment, la synthèse vocale est en réalité des segments d’enregistrement de la parole qui sont artificiellement attachés les uns à la suite de l’autre.

### Création des segments de la parole

La première étape était donc de créer une librairie de ces segments de la parole pour qu’ensuite le programme puisse utiliser ceux qu’il aurait besoin. Pour cela j’ai un programme tiers, Audacity, dans lequel j’ai énoncé des mots, des bouts de phrases de façon la plus naturelle et compréhensible possible. La capture du son s’est faite en « mono » pour que les segments soient des matrices de type (1,n) et donc plus facilement manipulables par la suite. Ensuite, j’ai extrait ces fameux segments à une fréquence d’échantillonnage de 22050Hz et d’une durée de chacun 0.1 seconde, soit 2205 échantillons pour chacun de ces segments. On aura donc bien à faire à des matrices (1 , 2205).

Quand j’avais commencé cette première étape j’avais tout d’abord pris des échantillons d’une durée supérieure, de l’ordre de 0.2 seconde mais avec une telle durée de segment la parole une fois synthétisée manquait profondément de dynamisme et donc de naturel.

Pour une raison de temps et d’utilité, je me suis limité aux segments qui seront par la suite utiles et pertinents pour les phrases types que devront savoir être énoncées par mon application. En effet, n’étant pas dans le cas d’un programme permettant la conversion d’un texte inconnu en un fichier sonore, les segments nécessaires sont connus.

### Programmation sous MATLAB

La seconde étape fut d’exploiter tous ces segments. Premièrement il faut donc transformer chaque segment sonore en une matrice. Cette opération se fait à l’aide du fichier ***‘creation\_librairie\_diphones.m’.*** Dans ce fichier en plus de faire cette opération, je crée une matrice ‘*diphones\_liste\_nom*’ qui contient la liste en caractère de tous les diphones disponibles. Ensuite est créé une matrice ‘*diphones\_liste\_matrice* ‘ dans laquelle est contenu toutes lignes matricielles des segments.

Il est indispensable que l’ordre des segments sonores soit identiquement le même dans les deux matrices. Car c’est de cette manière que seront retrouvés les diphones nécessaires pour la reconstitution du son. Le tout est ensuite sauver dans une librairie nommée ‘*librairie\_diphones.mat*’.

Une fois cela fait, il faut lancer le fichier ***‘creation\_librairie\_phrases.m’.*** Dans ce fichier sont contenues les différentes phrases que doit être capable de dire l’application. Chacune des phrases s’y trouve ainsi que sa décomposition en phonétique. MATLAB ne supportant pas les signes inhabituels j’ai donc dû choisir une alternative à la vraie écriture phonétique, ainsi, ***« /la/ /kɔ.mɑ̃d/ »*** devient ***« ‘l’ ‘a’ ‘k’ ‘o2’ ‘m’ ‘a2’ ‘d’ ».***  La correspondance de mon écriture avec la vraie écriture se trouve en annexe.

Comme le but premier de la présence dans cette application de la synthèse vocale est le fait de répondre à la personne, deux de ces phrases sont obligatoires : « La commande est comprise » et son opposé « La commande n’est pas comprise », laquelle sera complétée par « Veuillez la répéter ». D’autres phrases peuvent également enrichir la panoplie de réponses possibles.

Ensuite, le tout est également enregistré dans la librairie ‘*librairie\_phrases.mat*’. La création de ces librairies n’est pas innocente, elle a pour but de permettre de ne pas à chaque fois devoir refaire les opérations précitées avant chaque utilisation.

Dernière étape, la reconnaissance vocale va donc renvoyer un argument « *reponse* » dont la valeur servira à déterminer quelle phrase doit être synthétisée (pour effectuer un test sans reconnaissance vocale au préalable il suffit de définir manuellement l’argument « *reponse* »).

C’est le fichier ***‘lire\_phrase.m’*** qui va s’occuper de créer la phrase demandée, il va donc tout d’abord charger les deux librairies nécessaires. Une matrice vide ‘*phrase\_a\_dire*’ et allant servir à la concaténation est créé. Ensuite, il va regarder donc quelle phrase est demandée et à partir de là il va, à l’aide de boucles, parcourir les noms des éléments de la phrase et les comparer un par un avec tous les noms des diphones disponibles, une fois qu’il y’a une concordance entre ces noms. Il va ajouter dans la matrice ‘*phrase\_a\_dire*’ la matrice du diphones et ce l’un à la suite de l’autre et ainsi les concaténer.

|  |
| --- |
| *for i=1:size(phrase\_a\_synthetiser,1)*    *for k=1:size(diphones\_liste\_nom,1)*    *if phrase\_positive(i,:) == diphones\_liste\_nom(k,:)*    *% Concatène les diphones l'un à la suite de l'autre*  *phrase\_a\_dire = [phrase\_a\_dire , diphones\_liste\_matrice(:,k)];*    *end*  *end*  *end* |

Une fois toute la phrase parcourue et donc tous les segments concaténés, une nouvelle boucle va servir à énoncer toute la matrice ‘*phrase\_a\_dire*’.

|  |
| --- |
| [longueur, largeur] = size(phrase\_a\_dire);  for j=1:largeur    sound(phrase\_a\_dire(:,j),Fs);    end |

L’emploi d'une telle méthode de création a comme avantage de créer la matrice de concaténation à la volée et donc de ne pas avoir une matrice crée dès le départ pour chaque phrase comme ça aurait pu être le cas si on avait dès le départ effectué la concaténation lors de définition des différentes phrases. Le but était donc d’économiser la mémoire, non pas que ça soit un souci pour nos machines mais dans une optique de transposition vers un système où la quantité de mémoire pourrait faire défaut. Dans un souci de toujours économiser la mémoire, à la fin de ce fichier les deux matrices crées sont ensuite supprimées, celles-ci n’étant plus nécessaires.

|  |
| --- |
|  |

Schéma globale de la synthèse vocale.

# Améliorations à apporter par la suite

## Reconnaissance vocale

La reconnaissance vocale fut très concluante malgré de nombreux tests, je n’ai malheureusement pas vraiment eu l’occasion de la tester en multi-locuteur. Je ne sais donc pas réellement confirmer son efficacité sur la durée. Il serait donc intéressant de continuer à developper cet aspect.

Ensuite l’ergonomie, bien que j’ai tenté de faciliter la tâche de l’utilisateur au maximum, la création d’une interface graphique serait probablement plus pratique.

Enfin, il serait également intéressant de crée une plus grande librairies de matrices étalons, que ce soit en nombre de mots mais également en ayant plusieurs matrices de coefficients LPC pour un même mot.

## Synthèse vocale

Des améliorations sont naturellement possibles comme pour la reconnaissance vocale. Bien que je trouve le résultat final plutôt convaincant, mettre à disposition de l’application une plus grande quantité de diphones serait pas inutile, dans le but d’améliorer le rendu mais aussi de permettre une plus grande quantité de phrases.

Je tiens également à préciser que je n’ai pas eu le temps d’enregistrer assez de diphones pour constituer les deux phrases complètes que j’avais prévu. C’est-à-dire, « La commande est comprise » et « La commande n’est pas comprise, veuillez la répéter ». Cela sera cependant fait pour la présentation orale de mon travail.

Néanmoins, il y’a bien une synthèse vocale de présente mais incomplète, si la réponse de reconnaissance est positive la phrase « La commande » sera synthétisée, si au contraire la réponse de reconnaissance est négative la phrase « Commande la » sera synthétisée, pour quand même démontrer que cela fonctionne et que la notion de positif-négatif est prise en compte.

# Conclusion

Dans ce projet, l’objectif était donc de concevoir une interface machine qui permettra à une personne d’interagir vocalement avec son environnement. Le tout dans une optique d’offrir à des personnes invalides ou à mobilité réduite la possibilité de pouvoir interagir avec leur milieu environnant sans fournir d’effort. Cette interface nécessitait donc à la fois une reconnaissance et une synthèse vocale.

La démarche dans la conception de ce projet fut premièrement une démarche au niveau de la recherche, de la découverte et la compréhension de l’analyse qui permettra d’effectuer la reconnaissance vocale ainsi que la synthèse vocale. Et de ce fait, d’aborder tous les différents aspects que celles-ci englobent, tel que, la notion de son, de parole et sa décomposition et les différentes techniques de traitement. Cette démarche m’a naturellement appris de nombreuses nouvelles notions mais également à approfondir des notions jugées connues au préalable.

La seconde démarche fut d’appliquer toute cette théorie à la pratique afin de mener à bien l’objectif de départ. J’ai assurément approfondi mes connaissances dans la programmation sur MATLAB et la découverte totale de son extension SIMULINK. J’ai également eu tout au long de cette démarche pratique, l’envie de fournir une application accessible et facile d’utilisation, notamment en offrant une explication étape par étape tout au long de l’utilisation de l’application.

Dans une vision plus globale, ce travail fut aussi un apprentissage méthodologique, une façon de travailler en essayant d’être le plus productif mais également en rendant son développement le plus clair et organisé possible. La recherche continue afin de trouver la meilleure manière de procéder fut également un point formateur, notamment pour mon système de comparaison des matrices contenant les coefficients LPC des signaux.

En ce qui concerne mon travail de façon globale, j’ai été satisfait de voir un projet, que je jugeais au départ très complexe, petit à petit prendre forme et devenir accessible et très prenant. Il y’a naturellement des points à améliorer, à fignoler, peut-être également des nouvelles notions à apporter mais je pense qu’il constitue une base solide pour un prochain travail.

# Bibliographie

<http://www.mathworks.com/>

« Site officiel du programme Matlab »

http://www.enseignement.polytechnique.fr/profs/informatique/Georges.Gonthier/pi98/

chant.html

« Site réalisé par Emmanuel Bacry, professeur de l’*école Polytechnique de Paris*. »

http://tecfa.unige.ch/perso/staf/nova/ling/Psych.html

« Site réalisé par Nicolas Nova, étudiant en sciences appliquées d’informatique. »

<http://membres.multimania.fr/guillaumerey/reconnaissance_principes.htm>  
« Chapitre 2 - Analyse du signal de parole et décodage »

<http://msagent.online.fr/recon/globale.htm>  
« La Reconnaissance Vocale : l'approche globale »

<http://post.queensu.ca/~lessardg/Cours/215/chap2.html>  
« Chapitre2 : la phonétique »

<http://www.ears.dmu.ac.uk/>  
« ElectroAcoustic Resource Site »

<http://cours.musique.umontreal.ca/mus1321/Notes_de_cours/Csound_06_Voix.html>  
« La synthèse vocale »

<http://philduweb.free.fr/contributions/parole/analyse.htm>  
« Analyse du signal »

Mais également des forums et sites communautaires tels que :

<http://www.commentcamarche.net/>

<http://www.developpez.com/>

Principalement pour des explications et forums sur MATLAB

# Annexes

## Phonèmes du français

Dans la langue française on en compte un total de 37 :

***16*** *voyelles* ***+******3*** *semi-consonnes* ***+ 18*** *consonnes* ***=******37*** *(sans la jota espagnole, etc.).*

|  |  |  |
| --- | --- | --- |
| * ***Voyelles.***    + Phonème /i/ : **i**l ;   + Phonème /e/ : bl**é** ;   + Phonème /ε/ : l**ai**t ;   + Phonème /a/ : pl**at** ;   + Phonème /ɑ/ : p**â**te ;   + Phonème /ɔ/ : m**o**rt ;   + Phonème /o/ : m**o**t ;   + Phonème /u/ : gen**ou** ;   + Phonème /y/ : r**u**e ;   + Phonème /ø/ : p**eu** ;   + Phonème /œ/ : p**eu**r ;   + Phonème /ə/ : l**e** ;   + Phonème /ɛ̃/ : pl**ein** ;   + Phonème /ɑ̃/ : s**an**s ;   + Phonème /ɔ̃/ : b**on** ;   + Phonème /œ̃/ : br**un**. | * ***Semi-consonnes.***    + Phonème /j/ : **y**eux ;   + Phonème /w/ : **ou**i ;   + Phonème /ɥ/ : l**u**i. | * ***Consonnes***.    + Phonème /p/ : **p**ère ;   + Phonème /t/ : **t**erre ;   + Phonème /k/ : **c**ou ;   + Phonème /b/ : **b**on ;   + Phonème /d/ : **d**ans ;   + Phonème /g/ : **g**are ;   + Phonème /f/ : **f**eu ;   + Phonème /s/ : **s**ale ;   + Phonème /ʃ/ : **ch**at ;   + Phonème /v/ : **v**ous ;   + Phonème /z/ : **z**éro ;   + Phonème /ʒ/ : **j**e ;   + Phonème /l/ : **l**ent ;   + Phonème /ʁ/ : **r**ue ;   + Phonème /m/ : **m**ain ;   + Phonème /n/ : **n**ous ;   + Phonème /ɲ/ : a**gn**eau ;   + Phonème /h/ : **h**op ;   + Phonème /ŋ/ : campi**ng** ;   + Phonème /x/ : (jota, esp.) |

## Correspondance entre les noms employés pour MATLAB et les phonèmes

|  |  |  |
| --- | --- | --- |
| a | /a/ | pl**a**t |
| a2 | /ɑ̃/ | s**an**s |
| d | /d/ | **d**ans |
| k | /k/ | **c**ou |
| l | /l/ | **l**ent |
| m | /m/ | **m**ain |
| o2 | /ɔ/ | m**o**rt |

## Organisation de l’application

- acquisition\_voix.mdl

- apprentissage\_systeme.m

- comparaison.m

- distance\_dtw.m

- estimation\_signal\_acquis.m

- initialisation\_application.m

- reconnaissance.m

- sysstop\_new.m

- traitement\_LPC.mdl

----> *synthese\_vocale*

- creation\_librairie\_diphones.m

- lire\_phrase.m

----> diphones

a.wav

a2.wav

d.wav

k.wav

l.wav

m.wav

o2.wav

pause.wav

1. Le spectrogramme est un diagramme associant à chaque instant t d'un signal, son spectre de fréquence. [↑](#footnote-ref-1)
2. MATLAB est à la fois un langage de programmation et un environnement de développement [↑](#footnote-ref-2)