

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

جامعة سعد دحلب البليدة

UNIVERSITE SAAD DAHLEB DE BLIDA



كلية الهندسة - دائرة الإلكترونيك

FACULTE DES SCIENCES DE L'INGENIEUR

DEPARTEMENT D'ELECTRONIQUE

MEMOIRE

PROJET DE FIN D'ETUDES

POUR L'OBTENTION DU DIPLÔME DE

MASTER EN ELECTRONIQUE

**Option :** Traitement de l'Information et Systèmes Electroniques.

# COMMANDE VOCALE D'UN ACTIONNEUR

**Proposé et dirigé par:**

**Pro Mr:** BENSLAMA Z /A

**Réalisé par:**

HOUARI RAOUF

LARABI BILLEL

*Session: 2011*

## *Dédicace*

*Je dédie ce modeste travail à mes très chers parents, dont le sacrifice, la tendresse, l'amour, la patience, le soutien, l'aide et les encouragements ont été et reste l'essence de ma réussite.*

*Sans eux, je ne serais pas ce que je suis aujourd'hui.*

*Cette dédicace n'oubliera pas ma famille en tenant de rendre hommage à mes deux chers frères et à mes trois précieuses sœurs, pour leurs soutiens, leurs aides et leurs encouragements, et ma nièce MARAM et son papa et toute sa famille.*

*A mon compagnon de route HOUARI RAOUF et toute sa famille.*

*A AMINE MEKHTICHE et BOUDJEMAI DJAMEL pour leurs aides.*

*A tous mes enseignants, ainsi que toute mes amis restés fidèles.*

BILLEL

## *Dédicace*

*Je dédie ce modeste travail à mes très chers parents, dont le sacrifice, la tendresse, l'amour, la patience, le soutien, l'aide et les encouragements ont été et reste l'essence de ma réussite.*

*Sans eux, je ne serais pas ce que je suis aujourd'hui.*

*Cette dédicace n'oubliera pas ma famille en tenant de rendre hommage à mes deux chers frères et à ma précieuse petite sœur, pour leurs soutiens, leurs aides et leurs encouragements*

*A mon compagnon de route LARABI BILLEL et toutes sa familles*

*A tous mes enseignants, ainsi que mes amis restés fidèles.*

## *Remerciements*

*Sans l'aide et la grâce de DIEU ce travail n'aurais jamais vu son terme. Le mot « remercier » n'a pas l'amplitude voulu pour lui exprimer notre gratitude.*

*Nous remercions Monsieur BENSELAMA.Z/A notre promoteur pour son encadrement, sa rigueur, son suivi, ses précieux conseils porté à notre égard sans jamais oublier l'objectif rationnel de notre travail.*

*Nous tenons à remercier le jury d'avoir accepté d'être nos membres et de juger notre travail, nous n'oublierons pas les personnes qui nous sont chères, ainsi que ceux et celles qui ont été nos familles d'exceptions pour leurs soutiens explicites ou implicites.*

*Nous tenons aussi à remercier tous ceux qui ont participés de près ou de loin à notre formation.*

## Résumé

Ce projet de fin d'étude consiste à réaliser un système autonome de commande vocale d'un actionneur dont les commandes sont : gauche, droite et stop.

Pour cela nous avons développé et implémenté l'algorithme **VAD** (Voice Activity Detection) dans le but est d'automatiser notre système, ensuite nous avons développé l'algorithme **DTW** (Dynamique Time Warping) qui est responsable de reconnaître la commande qui est véhiculer du PC vers l'actionneur en question via une carte d'interface basé sur le microcontrôleur 68HC11 de MOTOROLA.

## Abstract

This final project is to perform an autonomous system of voice command of an actuator which orders are: left, right and stop.

For this we have developed and implemented the algorithm VAD (Voice Activity Detection) which aims to automate our system then we have developed the algorithm DTW (Dynamic Time Warping) who is responsible for recognizing the word of command that is convey the PC to the actuator in question via an interface card based on the microcontroller 68HC11 of MOTOROLA

## ملخص

هذا المشروع النهائي يتمثل في تحقيق التحكم الذاتي للمشغل بصوت عن طريق: يسار، يمين وتوقف.

لهذا قمنا بتطوير وتنفيذ خوارزمية VAD (كشف نشاط الصوت) الذي يهدف إلى جعل النظام آلي، ثم قمنا بتطوير خوارزمية DTW (التشويه الحيوي للزمن) المسؤولة عن معرفة كلمة التحكم المنقول من الكمبيوتر إلى المحرك، بواسطة بطاقة أساسها 68HC11 التابع لمتورولا.

# SOMMAIRE

Remerciements

Résumé

Sommaire

Liste des figures

Liste des abréviations

Introduction générale 02

## **Chapitre I** **Généralités**

I.1. Introduction 04

I.2. Généralité sur la parole 04

    I.2.1. Production de la parole 04

    I.2.2. L'appareil phonatoire 04

    I.2.3. L'appareille auditif 06

I.3. Les paramètres descriptifs de la parole 07

    • Classification des sons 07

        ❖ Les sons voisés 07

        ❖ Les sons non voisés 08

    I.3.1 Paramètre phonétique 08

        • La phonétique 08

        • La phonologie 08

        • Phonème 08

        • Classification des phonèmes 08

        • Mode d'articulation 09

        • Point d'articulation 09

    I.3.2. Paramètre acoustique 10

|   |    |
|---|----|
| • Numérisation  | 10 |
| • Fréquence fondamentale  | 11 |
| • Le spectre du signal  | 11 |
| • Amplitude   | 12 |
| I.3.3. Stationnarité du signal vocale                                 | 12 |
| • Le fenêtrage  | 13 |
| I.4. Les branches du traitement de parole                             | 14 |
| - Les analyseurs  | 14 |
| - Les synthétiseurs   | 14 |
| - Les reconnaisseurs  | 15 |
| I.5. Application de la reconnaissance des formes pour le signal vocal | 16 |
| I.6. Complexité du signal de parole                                   | 17 |
| I.7. Conclusion   | 17 |

## Chapitre II

## Les algorithmes VAD et DTW

|  |    |
|--|----|
| II.1. Introduction   | 19 |
| II.2. La détection d'activité vocale   | 19 |
| II.2.1. La problématique de la détection d'activité vocale                           | 19 |
| II.2.2. Détection d'activité vocale basée sur l'adaptation du spectre de bruit       | 20 |
| ❖ Déroulement de la détection d'hypothèse  | 21 |
| II.2.3. Mécanisme d'estimation Probabiliste  | 22 |
| II.2.3.1. Le théorème Bayes  | 22 |
| II.2.3.2. Application du théorème de Bayes aux échantillons du signal                | 23 |
| II.2.4. Espace des observations  | 25 |
| II.2.4.1. Aperçu de la loi Normale   | 25 |
| II.2.4.2. Application de la loi Normale au signal $X_k$                              | 26 |
| ❖ Explication sur le paramètre inconnu conjoint à la densité de probabilité $\Theta$ | 26 |
| II.2.4.3. Ajustement des paramètres du model (MLE)                                   | 27 |
| II.2.5. Règle de décision  | 28 |
| II.2.6. La prédiction du bruit de fond   | 29 |
| II.2.7. Organigramme de la VAD   | 31 |
| II.3. La Déformation Temporelle Dynamique DTW  | 31 |

|   |    |
|---|----|
| II.3.1. Déformation temporelle linéaire | 32 |
| II.3.2. Le principe de la DTW           | 33 |
| II.4. Organigramme de la DTW            | 37 |
| II.5. Difficulté et implémentation      | 38 |
| II.6. Conclusion                        | 38 |

## **Chapitre III**                      **Architecture de transmission d'un protocole série**

|  |    |
|--|----|
| III.1. Introduction                                  | 40 |
| III.2. Description général du microcontrôleur 68HC11 | 40 |
| • Les ports  | 42 |
| • Port D   | 42 |
| III.3. Transmission des données                      | 42 |
| III.3.1. Le protocole série                          | 42 |
| • La transmission sériée synchrone                   | 42 |
| • La transmission sériée asynchrone                  | 43 |
| III.4. L'interface SCI                               | 44 |
| III.4.1. Fonctionnement en émission                  | 45 |
| III.4.2. Fonctionnement en réception                 | 45 |
| ➤ Réveil par la ligne d'attente (IDLE LINE)          | 45 |
| III.4.4. Registre du port série                      | 46 |
| III.5. La liaison RS 232                             | 47 |
| III.5.1. Caractéristiques et performance             | 47 |
| • Caractéristiques électriques                       | 47 |
| • Caractéristiques physiques                         | 48 |
| • Fonctionnalité des broches (DB9)                   | 49 |
| III.5.2. Le câblage des trois fils DTE/DTE           | 50 |
| III.6. L'UART  | 51 |
| • Adresse de base des quatre premier RS 232          | 53 |
| III.6.1. Le Principe de fonctionnement               | 53 |
| III.7. La carte a microcontrôleur                    | 53 |
| III.7.1. Interface série                             | 54 |
| III.7.2. Reset                                       | 54 |

|   |    |
|---|----|
| III.7.3. Horloge                                    | 55 |
| III.7.4. Alimentation                               | 55 |
| II.8. Carte d'isolation optique                     | 55 |
| III.8.1. Principe de fonctionnement des opto-triacs | 55 |
| III.8.2. Choix d'opto-triac                         | 55 |
| III.8.3. Circuit de la carte d'isolation optique    | 56 |
| III.9. Carte de puissance                           | 56 |
| III.10. Conclusion                                  | 57 |

## **Chapitre IV** **Application et résultats**

|  |    |
|--|----|
| IV.1. Introduction                                   | 59 |
| IV.2. Description fonctionnelle de notre application | 59 |
| IV.3. Programmation et mise en marche                | 60 |
| IV.3.1. Le logiciel BASIC11                          | 61 |
| • Conditions d'application du BASIC11                | 61 |
| • Le programme de BASIC11                            | 62 |
| • Le compilateur                                     | 63 |
| • Chargement de programme                            | 63 |
| IV.3.2. Exécution de programme MATLAB                | 64 |
| • Construction de la base de données                 | 64 |
| • La détection de l'activité vocale                  | 64 |
| • Reconnaissance du mot                              | 66 |
| IV.4. Conclusion                                     | 70 |

|                     |    |
|---------------------|----|
| Conclusion générale | 72 |
|---------------------|----|

Bibliographie

Annexes

## **Introduction générale**

La reconnaissance automatique de la parole est un domaine de la science ayant toujours eu un grand attrait auprès des chercheurs comme auprès du grand public. En effet, qui n'a jamais rêvé de pouvoir parler avec une machine ou, du moins, piloter un appareil ou un ordinateur par la voix, ne plus avoir à se lever pour allumer ou éteindre tel ou tel appareil électrique, ne plus avoir à taper pendant des heures sur un clavier pour rédiger un rapport (par exemple), l'homme étant par nature indolent, une telle technologie a toujours suscité chez lui une part d'envie et d'intérêt, ce que peu d'autres technologies ont réussi à faire.

On peut citer quelque produit disponible sur le marché comme les ordinateurs comportant l'option de la dictée vocale, les récepteur TV à commande vocale ou encore l'utilisation des coffres fort dont le système de sécurité est basée sur une signature associée à la voix de son propriétaire.

Notre projet de fin d'étude consiste à réaliser un système actionné par commande vocale afin d'être activé grâce à une reconnaissance automatique d'un mot isolé. Pour ce faire, nous utilisons un PC pour faire la reconnaissance à partir d'une interface qui comporte une carte de commande basée sur l'architecture du microcontrôleur 68HC11 de MOTOROLA et associée à une carte de puissance pour la dite commande.

Notre mémoire à été partagé en quatre chapitres. Le premier est dédié aux généralités sur les notions de base du traitement de parole, suivi du deuxième chapitre dans lequel on a introduit les algorithmes respectifs de détection d'activités vocales **VAD (Vocal Activity Detection)** et de la reconnaissance du mot isolé **DTW (DynamicTime Warping)** qui sont responsables de l'automatisation de notre système et de la reconnaissance du mot isolé, arrivé au troisième chapitre nous avons souhaiter présenter le processus de la transmission des données entre l'ordinateur et la carte de commande par l'intermédiaire du MAX 232 appuyés par leurs schémas respectifs. Le dernier chapitre a été élaboré de manière à présenter les résultats obtenus grâce à la réalisation de notre application ainsi que les logiciels utilisés pour la programmation. Une conclusion scellera notre chapitre par la présentation sommaire de la conception de notre panel d'interface.

## I.1. Introduction

Le traitement de parole est aujourd'hui une composante fondamentale des sciences de l'ingénieur, située au croisement du traitement du signal numérique et du traitement du langage (c'est-à-dire du traitement de données symboliques), cette discipline scientifique a connu depuis les années 60 une expansion fulgurante, liée au développement des moyens et des techniques de télécommunication, c'est pour cela nous avons l'idée de réaliser un système de commande vocale et pour cela nous avons besoin des outils suivants :

- Un outil de traitement (PC).
- Une interface qui se compose d'une carte de commande et d'une carte de puissance.

## I.2. Généralité sur la parole

Pour mieux comprendre le principe de fonctionnement de notre système il faut avoir une certaine culture sur la parole et pour cela nous allons introduire les points essentiels de la parole.

### I.2.1. Production de la parole

La parole peut être décrite comme résultat de l'action volontaire et coordonné d'un certain nombre de muscles, cette action se déroule sous le contrôle du système nerveux central, qui reçoit en permanence des informations par rétroaction auditive.

### I.2.2. L'appareil phonatoire

L'appareil respiratoire fournit l'énergie nécessaire à la production de sons, elle se décompose en trois sous-ensembles fonctionnels :

Les poumons et la trachée-artère, le larynx et le conduit vocal, les muscles abdominaux et thoraciques sont les moteurs de la respiration les poumons compriment par cette musculature agissent comme un sifflet et fournissent de l'air sous pression dans la trachée-artère (pression subglottique), le débit d'air exprimé est ensuite modulé par les mouvements du larynx et du conduit vocale. [1]

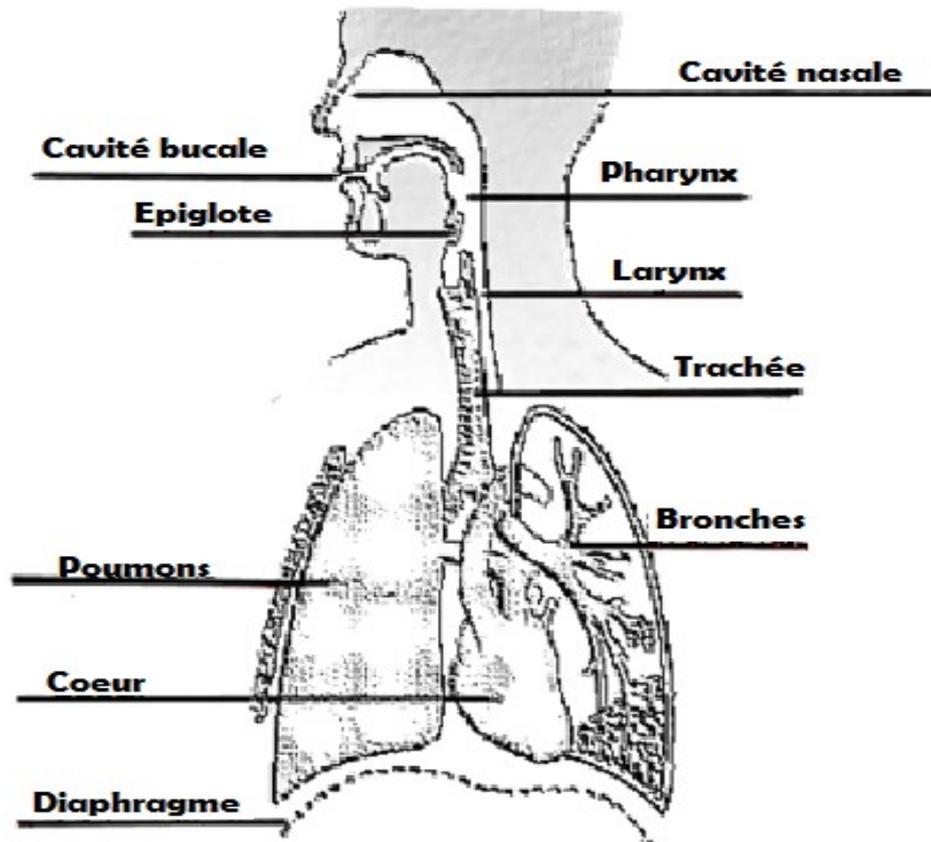


Figure I.1 : L'appareil phonatoire.

Le **larynx** est un ensemble de muscle et de cartilages mobiles qui entourent une cavité située à la partie supérieur de trachée-artère, où la pression de l'air est modulée avant d'être appliquée au conduit vocal.

Les **cordes vocales** sont en fait deux lèvres symétrique placées en travers le larynx, ces lèvres peuvent fermer complètement le larynx et, en s'écartant progressivement, détermine une ouverture triangulaire appelée **glotte**, l'air y passe librement pendant la respiration et la voix chuchote, et pendant la phonation des sons non-voisé (sourds) ; les sons voisés (sonore) résultent d'une vibration des cordes vocale, le larynx est d'abord fermé, se qui croit la pression on amont des cordes vocales, et les forces à s'ouvrir, ce qui fait tomber la pression, et permet aux cordes vocales de se fermer ; des impulsions périodiques de pressions sont ainsi appliquées au conduit vocale composé des cavités pharyngienne et buccale pour la plupart des sons.

La **luette** est en position basse, la cavité nasale vient s'y ajouter en dérivation, la langue jouant un rôle prépondérant dans le processus phonatoire.

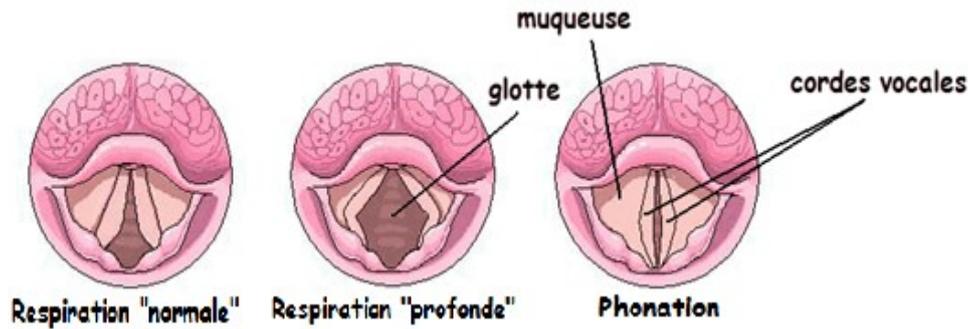
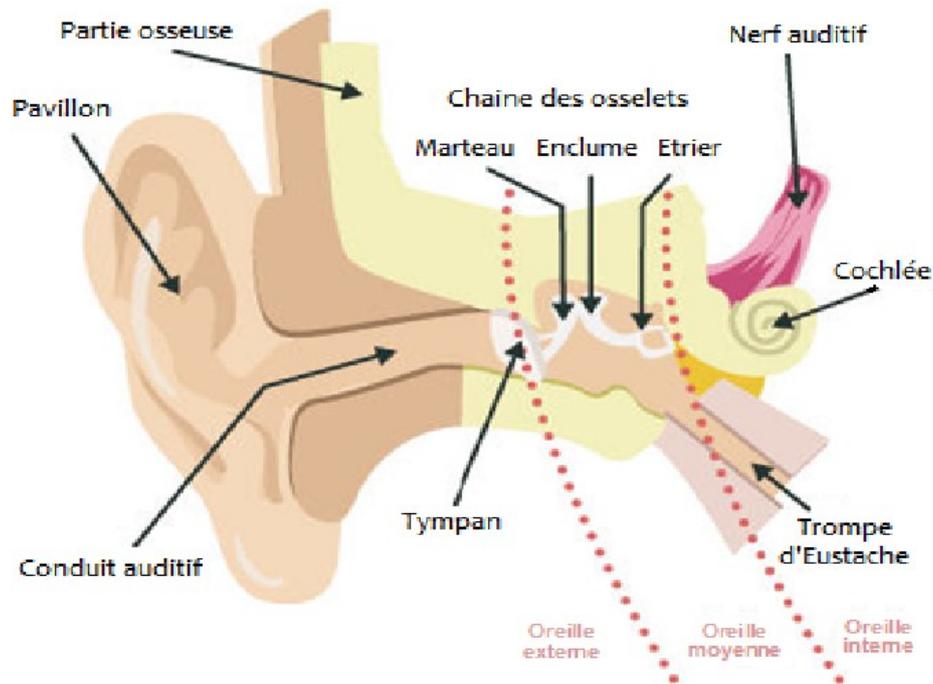


Figure I.2 : Vue de dessus de la section du larynx.

### I.2.3. L'appareil auditif

L'appareil auditif comprend l'oreille externe, l'oreille moyenne, et l'oreille interne. **Figure I.3**, le conduit auditif relie le pavillon au tympan : c'est un tube acoustique de section uniforme fermé à une extrémité, son premier mode de résonance est situé vers 3000 Hz, ce qui accroît la sensibilité du système auditif dans cette gamme de fréquences, le mécanisme de l'oreille interne (marteau, étrier, enclume) permet une adaptation d'impédance entre l'air et le milieu liquide de l'oreille interne, les vibrations de l'étrier sont transmises au liquide de la cochlée, celle-ci contient la membrane basilaire qui transforme les vibrations mécaniques en impulsions nerveuses ; la membrane s'élargit et s'épaissit au fur et à mesure que l'on se rapproche de l'apex de la cochlée, elle est le support de l'organe de Corti qui est constitué par environ 25000 cellules ciliées raccordées au nerf auditif.

Les fibres nerveuses aboutissent à une région de l'écorce cérébrale appelée aire de projection auditive et située dans le lobe temporal, en cas de lésion de cette aire, on peut observer des troubles auditifs, les fibres nerveuses auditives afférentes (de l'oreille au cerveau) et efférentes (du cerveau vers l'oreille) sont partiellement croisées chaque moitié du cerveau est mise en relation avec les deux oreilles internes. [2]



**Figure I.3** : Le système auditif.

### I.3. Les paramètres descriptifs de la parole

Le signal de la parole peut être décrit de plusieurs façons, on distingue généralement plusieurs niveaux de description non exclusifs : Acoustique, phonétique, morphologique, syntaxique, sémantique, pragmatique et phonologique, dans ce qui va suivre nous allez-vous parler de quelque uns.

Nous n'allons pas développer les autres paramètres puisque nous n'auront pas besoin d'eux dans la suite.

- **Classification des sons**

Comme on la dit auparavant il y a plusieurs organes qui interviennent dans la production de la parole, c'est pour cela qu'il en résulte différents types de son, on peut citer :

- ❖ **Les sons voisés**

Ce sont les sons qui résultent de l'excitation du conduit vocale par des impulsions périodiques dues aux vibrations des cordes vocales avec une fréquence appelée fréquence fondamentale.

**❖ Les sons non voisés**

Ils résultent du passage d'air dans le conduit vocale, il peut être assimilé à un bruit blanc filtré par la transmittance du conduit, son spectre ne présente aucune périodicité et son amplitude est plus importante vers les hautes fréquences.

**I.3.1. Paramètre phonétique :****• La phonétique**

Branche de la linguistique qui étudie les phonèmes qui s'intéresse à la manière dont les sons du langage sont produits, transmis et perçus par les sujets parlants.

**• La phonologie**

Science qui étudie les phonèmes non en eux-mêmes, mais quant à leur fonction dans la langue.

**• Phonème**

Un phonème est la plus petite unité dans la parole, le nombre de phonème est très limité, on l'estime à 36 phonèmes, les phonèmes sont un ensemble d'images acoustiques emmagasinées dans le cerveau de l'individu à condition qu'il maîtrise la langue. Il y a quatre sortes de phonème ou quatre classes.

**• Classification des phonèmes**

## 1) Les voyelles :

- Voyelles orales : Qui sont des sons sonores, correspondant à une configuration du conduit vocales sans l'intervention de la cavité nasale.
- Voyelles nasales : Ce sont des voyelles qui sortent par les narines et par la bouche.

## 2) Les semi-consonnes.

## 3) Les consonnes : Qui correspondent à des sons essentiellement dynamiques, elles sont décomposées en : liquide, nasale, fricatives, et occlusive selon le mode d'articulation.

- **Mode d'articulation**

Le mode d'articulation est défini par un certain nombre de facteurs qui modifient la nature du courant d'air expiré :

- Intervention des cordes vocales ou mise en vibration: **articulation sonore.**
- Fermeture momentanée du passage de l'air suivie d'une ouverture brusque (explosion) : **articulation occlusive.**
- Rétrécissement du passage de l'air qui produit un bruit de friction ou de frôlement : **articulation fricative.**
- Position abaissée du voile du palais: **articulation nasale.**
- Contact de la langue au milieu du canal buccal; l'air sort des deux côtés: **articulation latérale.**
- Une série d'occlusions brèves et séparées de la luvette: **articulation vibrante.**

- **Point d'articulation**

Le point d'articulation est l'endroit où se trouve, dans la cavité buccale, un obstacle au passage de l'air. De manière générale, on peut dire que le point d'articulation est l'endroit où vient se placer la langue pour obstruer le passage du canal d'air.

Le point d'articulation peut se situer aux endroits suivants :

- Les lèvres (**articulations labiales ou bilabiales**).
- Les dents (**articulations dentales**).
- Les lèvres et les dents (**articulations labiodentales**).
- Les alvéoles (**articulations alvéolaires**).
- Le palais (**articulations palatales**).
- Le voile du palais (**articulations vélares**).
- La luvette (**articulations uvulaires**).

| Mode d'articulation |        | Lieu d'articulation |               |               |                |               |                |
|---------------------|--------|---------------------|---------------|---------------|----------------|---------------|----------------|
| Occlusive           |        | Bi-labiale          | Labio-dentale | Apico-dentale | Dorso-palatale | Dorso-vélaire | Dorso-uvulaire |
| Sourde              | Orale  | p                   |               | t             |                | k             |                |
|                     | Nasale |                     |               |               |                |               |                |
| Sonore              | Orale  | b                   |               | d             |                | g             |                |
|                     | Nasale | m                   |               | n             |                | ɲ             |                |
| Constrictive        |        |                     |               |               |                |               |                |
| Sourde              | Orale  |                     | f             | s             |                | ʃ             |                |
| Sonore              | Orale  |                     | v             | z             |                | ʒ             |                |
| Liquide             |        |                     |               | l             |                |               | R              |

Tableau I.1 : Exemples phonétique.

### I.3.2. Paramètre acoustique

Pour étudier les paramètres acoustiques du signal de la parole, on doit en premier lieu convertir se dernier en un signal électrique grâce à un transducteur (un microphone) souvent numérisé, on peut alors extraire les traits acoustiques : sa fréquence fondamentale, son énergie, et son spectre, chaque trait acoustique est lui-même lié à une grandeur perceptuelle : pitch, intensité, et timbre.

- **Numérisation**

Cette opération est schématisée à la figure ci-dessous, requiert successivement : un filtrage, un échantillonnage et une quantification **Figure 1.4**, l'échantillonnage transforme le signal à temps continu  $x(t)$  en signal à temps discret  $x(n)$  défini aux instants d'échantillonnage, multiples entiers de la période d'échantillonnage; celle-ci est elle-même l'inverse de la fréquence d'échantillonnage, pour ce qui concerne le signal vocal, le choix de cette fréquence d'échantillonnage résulte d'un compromis, son spectre peut s'étendre jusque 12 KHz ; il faut donc en principe choisir une fréquence égale à 24 KHz au moins pour satisfaire raisonnablement au théorème de Shannon. Cependant, le coût d'un traitement numérique, filtrage, transmission, ou simplement enregistrement peut être réduit d'une façon notable si l'on accepte une limitation du spectre par un filtrage préalable. Pour la téléphonie, on estime que le signal garde une qualité suffisante lorsque son spectre est limité à 3400 Hz et l'on

choisit une fréquence d'échantillonnage égale à 8000 Hz, pour les techniques d'analyse, de synthèse ou de reconnaissance de la parole, la fréquence peut varier de 6000 à 16000 Hz, par contre pour le signal audio (parole et musique), on exige une bonne représentation du signal jusque 20 KHz et l'on utilise des fréquences d'échantillonnage de 44.1 ou 48 KHz.

Parmi le continuum des valeurs possibles pour les échantillons  $x(n)$ , la quantification ne retient qu'un nombre fini  $2b$  de valeurs ( $b$  étant le nombre de bits de la quantification), espacées du pas de quantification  $q$ , le signal numérique résultant est noté  $x(n)$ , une quantification de bonne qualité requiert en général 16 bits.

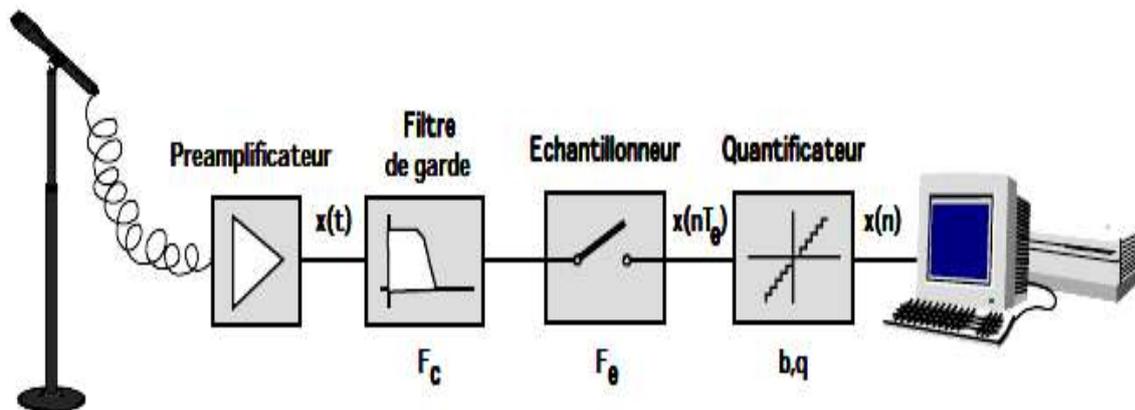


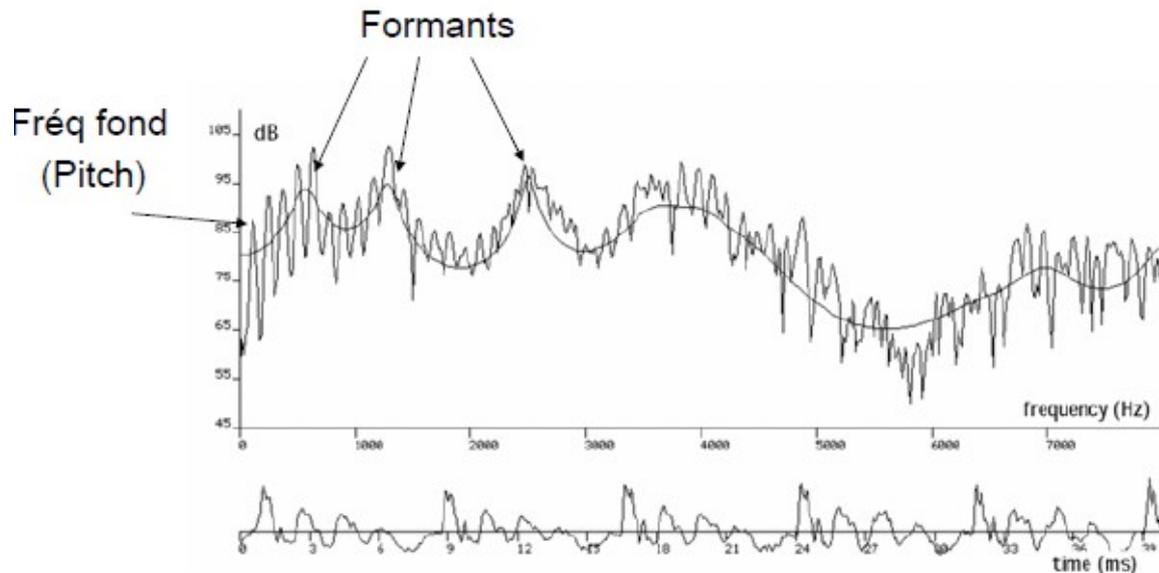
Figure I.4 : Enregistrement numérique d'un signal acoustique.

- **Fréquence fondamentale**

La fréquence correspond au nombre de cycles par seconde (ou nombre de Hertz), la fréquence fondamentale pour le sexe masculin varie de 70 Hz à 250 Hz et pour le sexe féminin sa varie de 150 Hz à 400 Hz, et de 200 Hz à 600 Hz pour les enfants ; on déduit que plus la fréquence est élevée plus la voix est aigue, l'oreille humaine a pour bande de perception de [20 Hz, 2 KHz].

- **Le spectre du signal**

Le spectre d'un signal c'est sa représentation en fonction des fréquences au lieu du temps, il est obtenu au moyen de la transformée de Fourier où plutôt la transformée de Fourier rapide c'est-à-dire la FFT (**F**ast **F**ourier **T**ransform).



**Figure I.5:** Evolution temporelle d'un son voisé et son spectre obtenu par transformée rapide de Fourier (FFT)

Dans un signal de parole on trouve des parties voisées qui apparaissent sous la forme de successions de pics spectraux marqués, dont les fréquences centrales sont multiples de la fréquence fondamentale, par contre le spectre du signal non voisé ne présente aucune structure particulière, la forme générale de ces spectres, appelée **enveloppe spectrale**, présente elle-même des pics et des creux qui correspondent aux résonances et aux antirésonances du conduit vocale et sont appelés **formants** et **antiformants**, l'évolution temporelle de leur fréquence centrale et de leur bande détermine le timbre du son, il apparaît en pratique que l'enveloppe spectrale des sons voisés est de type passe-bas, avec environ un formant par KHz de bande passante, et dont seuls les trois ou quatre premiers contribuent de façon importante au timbre. Par contre les sons non voisés présentent souvent une accentuation vers les hautes fréquences. [3]

- **Amplitude**

C'est la distance entre le point de repos et le point le plus élevée du signal, on la mesure en décibel (dB), le seuil de perception du son pour l'oreille humaine est le 0 dB et pour celui de la douleur c'est à 120 dB.

### I.3.3. Stationnarité du signal vocale

Etant donnée que le signal vocale a la propriété d'avoir une structure complexe et variable dans le temps, tantôt périodique (plus exactement pseudopériodique) pour les sons voisés, tantôt aléatoire pour les sons fricatifs et parfois impulsionnelles dans les phases explosives des son occlusifs c'est pour cela qu'il est nécessaire de faire un fenêtrage du signal vocale.

- **Le fenêtrage**

Le signal de parole est globalement non stationnaire mais il l'est par tranche, c'est pour cela qu'on doit le segmenter avec une fenêtre en plusieurs tranches périodique de l'ordre de 20 ms à 30 ms et pour ne pas avoir une perte du signal on fait glisser cette fenêtre de façon qu'elles se recouvrent avec un taux de 40 % à 50%, pour ce qui est de la valeurs de la fenêtre et du taux de recouvrement ,ils sont prédéfinie par les expérience des gents du domaine.

- Pondération par une fenêtre de Hamming :

Le fenêtrage du signal de parole par une fenêtre de Hamming permet de diminuer les effets du bord c'est à dire le changement brusque du signal.

La fenêtre de Hamming est définie dans ce cas :

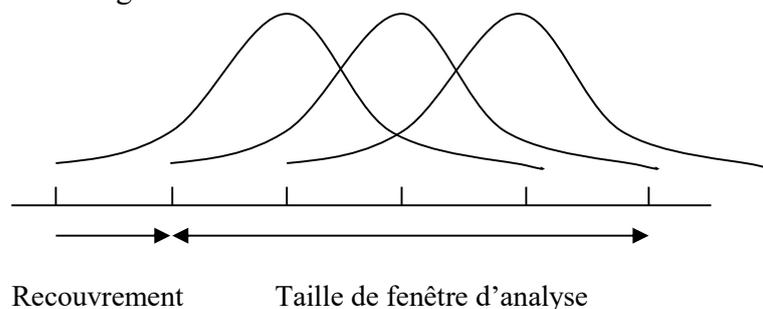
$$W(n) = 0.54 - 0.46 \cdot \cos\left(\frac{2\pi \cdot n}{N-1}\right) \quad (\text{I.1})$$

Pour la pondération, il suffit de multiplier notre signal par la fenêtre de Hamming :

$$X(n) = S(n) \cdot W(n) \quad (\text{I.2})$$

$X(n)$  : le signal après fenêtrage.

$S(n)$  : le signal avant fenêtrage



**Figure I.6 :** Fenêtre de Hamming.

## I.4. Les différents traitements du signal de parole

Que peut-on faire comme traitement pour un signal de parole ? La réponse à cette question est la suivante : dans cette discipline on trouve plusieurs branches parallèles dont on va vous parler, on les classe comme suit :

### - Les analyseurs

Les analyseurs de parole cherchent à mettre en évidence les caractéristiques du signal vocal tel qu'il est produit, ou parfois tel qu'il est perçu (on parle alors d'*analyseur perceptuel*), mais jamais tel qu'il est compris, ce rôle étant réservé aux reconnaisseurs.

Les caractéristiques en question sont :

- LPC **L**inear **P**redictive **C**oding (codage par prédiction linéaire).
- LPCC **L**inear **P**rédition **C**epstral **C**oefficients.
- MFCC **M**EL-**F**requency **C**epstral **C**oefficients.

L'analyse est utilisé soit comme composante de base des systèmes de codage, de reconnaissance ou de synthèse (voir ci-dessous), soit en tant que tels pour des applications spécialisées, comme l'aide au diagnostic médical (pour les pathologies du larynx, par analyse du signal vocal) ou l'étude des langues.

### - Les synthétiseurs

Les synthétiseurs ont quant à eux la fonction inverse de celle des analyseurs et des reconnaisseurs de parole : ils produisent de la parole artificielle, on distingue fondamentalement deux types de synthétiseurs : les ***synthétiseurs de parole à partir d'une représentation numérique***, inverses des analyseurs, dont la mission est de produire de la parole à partir des caractéristiques numériques d'un signal vocal telles qu'obtenues par analyse, et les ***synthétiseurs de parole à partir d'une représentation symbolique***, inverse des reconnaisseurs de parole et capables en principe de prononcer n'importe quelle phrase sans qu'il soit nécessaire de la faire prononcer par un locuteur humain au préalable. Dans cette seconde catégorie, on classe également les synthétiseurs en fonction de leur mode opératoire :

- Les ***synthétiseurs à partir du texte*** reçoivent en entrée un texte orthographique et doivent en donner lecture.

- Les *synthétiseurs à partir de concepts*, appelés à être insérés dans des systèmes de dialogue homme-machine, reçoivent le texte à prononcer et sa structure linguistique, telle que produite par le système de dialogue.

### - Les reconnaisseurs

Les reconnaisseurs ont pour mission de décoder l'information portée par le signal vocal à partir des données fournies par l'analyse, on distingue fondamentalement deux types de reconnaissance, en fonction de l'information que l'on cherche à extraire du signal vocal : la *reconnaissance du locuteur*, dont l'objectif est de reconnaître la personne qui parle, et la *reconnaissance de la parole*, où l'on s'attache plutôt à reconnaître ce qui est dit, on classe également les reconnaisseurs en fonction des hypothèses simplificatrices sous les quelles ils sont appelés à fonctionner. Ainsi :

- En reconnaissance du locuteur, on fait la différence entre l'*identification* et la *vérification* du locuteur, selon que le problème est de vérifier que la voix analysée correspond bien à la personne qui est sensée la produire, ou qu'il s'agit de déterminer qui, parmi un nombre fini et préétabli de locuteurs, a produit le signal analysé.

- On sépare reconnaissance du locuteur *dépendante du texte*, reconnaissance *avec texte dicté*, et reconnaissance *indépendante du texte*. Dans le premier cas, la phrase à prononcer pour être reconnu est fixée dès la conception du système ; elle est fixée lors du test dans le deuxième cas, et n'est pas précisée dans le troisième.

- On parle de reconnaisseur de parole *monolocuteur*, *multilocuteur*, ou *indépendant du locuteur*, selon qu'il a été entraîné à reconnaître la voix d'une personne, d'un groupe fini de personnes, ou qu'il est en principe capable de reconnaître n'importe qui.

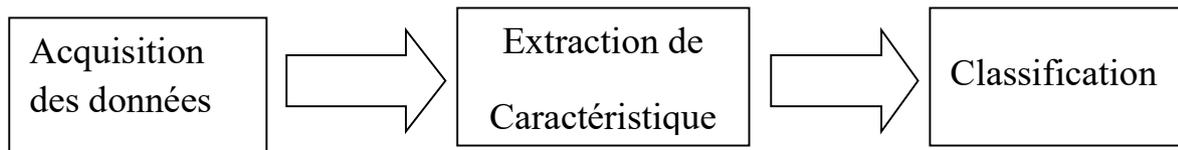
- On distingue enfin reconnaisseur de *mots isolés*, reconnaisseur de *mots connectés*, et reconnaisseur de *parole continue*, selon que le locuteur sépare chaque mot par un silence, qu'il prononce de façon continue une suite de mots prédéfinis, ou qu'il prononce n'importe quelle suite de mots de façon continue. [2]

Afin d'atteindre notre objectif nous devons passer par le chemin des reconnaisseurs et plus précisément celui de la reconnaissance de mots isolé, son principe est simple, on a un

nombre de mots du vocabulaire qui constitue notre base de référence avec laquelle on doit comparer le mot qu'on veut reconnaître.

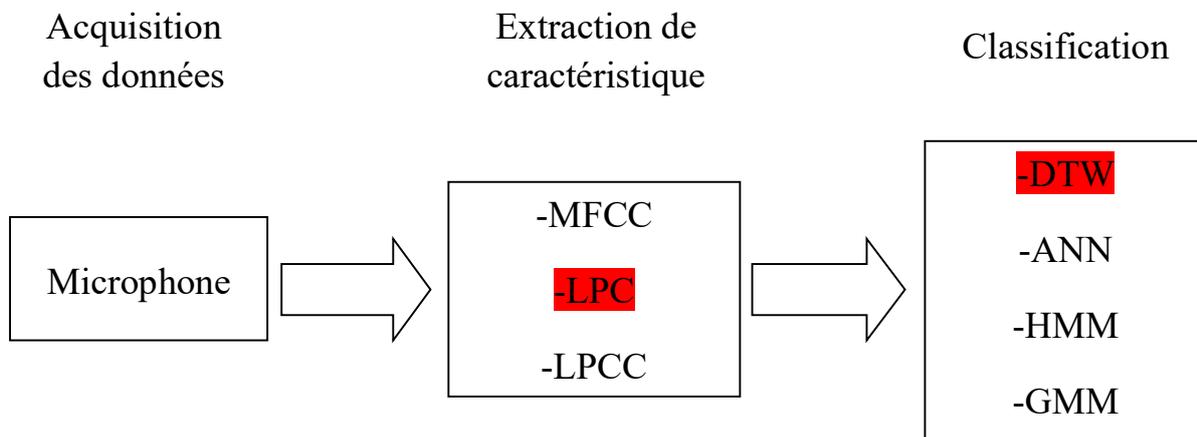
### I.5. Application de la reconnaissance des formes pour le signal vocal

Pour la reconnaissance de parole en générale on trouve plusieurs techniques, parmi ces techniques il y a les techniques de reconnaissance de forme qui suivent un parcours décrit comme suit :



**Figure I.7 :** Schéma générale de la reconnaissance de forme.

On voit bien qu'il ya trois étapes essentiels dans ce processus, la première étape c'est l'acquisition de données, ensuite on a l'extraction de caractéristique, et enfin arrive la classification, pour la parole on trouve le même schéma seulement la troisième étape nous avons une reconnaissance au lieu d'une classification, et comme nous faisons de la reconnaissance du mot isolé on aura qu'à suivre le schéma de la figure ci-dessous :



**Figure I.8 :** Schéma de la reconnaissance du mot isolé.

- HMM : Les chaînes de Markov caché.
- ANN : Les réseaux de neurones artificiels (RN).
- DTW : Déformation temporelle dynamique.
- GMM : Mixture de multi gaussienne.

Comme on voit dans la **Figure I.8**, il y a plusieurs caractéristiques qu'on peut utiliser pour notre reconnaissance mais on a choisie les LPC [1], car ils sont simples à extraire et donnent des bons résultats, et pour l'étape de la reconnaissance on a opté pour l'utilisation de l'algorithme de la DTW puisque il est l'algorithme par référence pour la reconnaissance du mot isolé.

## **I.6.Complexité du signal de parole**

Le signal de parole est des plus complexes signaux connue et l'une des raisons qu'il le soit c'est sa grande variabilité, en effet une même personne ne prononce jamais un mot deux fois d'une manière identique, par exemple pour la vitesse d'élocution c'est elle qui détermine la durée du signal, l'intensité qui peut fléchira résultat d'un coup de fatigue enfin beaucoup de variables qui rentrent en considération (l'âge le sexe, la santé, l'origine, l'émotion, l'intention).

## **I.7. Conclusion**

Dans ce chapitre nous avons introduit des notions de base tel que : la parole, l'appareil phonatoire, les paramètres du signal vocale, la reconnaissance vocale, dans la perspective d'aboutir à notre système de commande vocale basée sur la reconnaissance des mots isolés.

Le chapitre suivant est dédié aux algorithmes utilisés dans ce projet, nous parlons au pluriel car en plus de l'algorithme DTW nous avons besoin d'un autre algorithme afin d'automatisé notre système qui est nulle autre que celui de la détection d'activité vocale (VAD).

## II.1. Introduction

Dans ce chapitre nous présentons les algorithmes utilisés permettant d'atteindre notre objectif à savoir l'algorithme VAD responsable de l'automatisation de notre système de commande vocale pour ensuite greffer le principe de la DTW et son fonctionnement car considéré comme algorithme adéquat à la reconnaissance du mot isolé.

## II.2. La détection d'activité vocale

La détection d'activité vocale est un élément important pour la reconnaissance vocale en temps réel la robustesse du système ou bien l'échec de la reconnaissance dépend essentiellement de l'implémentation d'un élément appelé VAD (*Voice Activity Detection*) celle-ci opère en isolant le signal utile (Le signal de la parole) du bruit de fond en temps réel, sachant que ce dernier peut changer constamment, le VAD utilise une grande variété de méthodes qui sont : la comparaison de l'énergie à un seuil, la détection de voisement et l'estimation du pitch, l'analyse spectrale, le rythme de passage par zéro ou de changement de signe, la mesure de périodicité, l'étude statistique du résidu de la prédiction linéaire. Incluent souvent une combinaison de plusieurs de ces méthodes.

### II.2.1. Problématique de la détection d'activité vocale

Lorsqu'on parle on a une variation de l'énergie du signal dans le domaine temporelle comme il est montré sur ces deux figures ci-dessous.

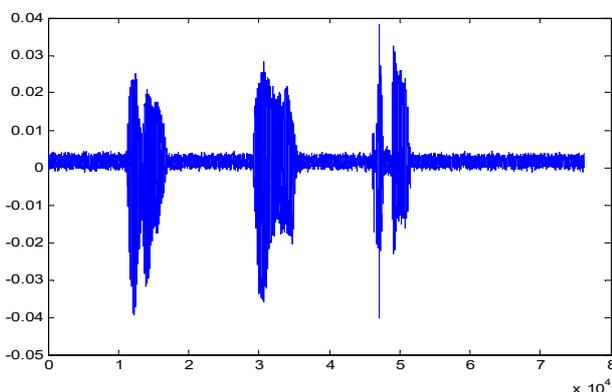


Figure II.1 : Bruit de fond + signal vocal.

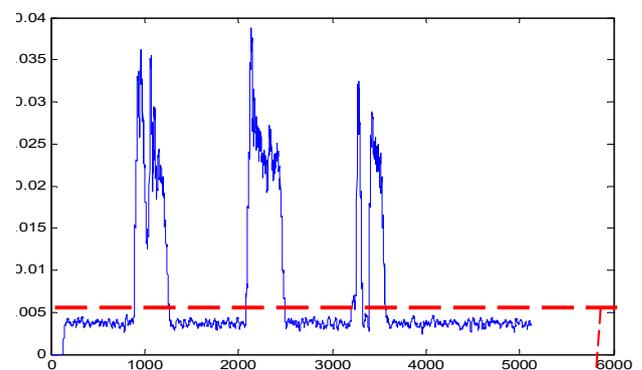
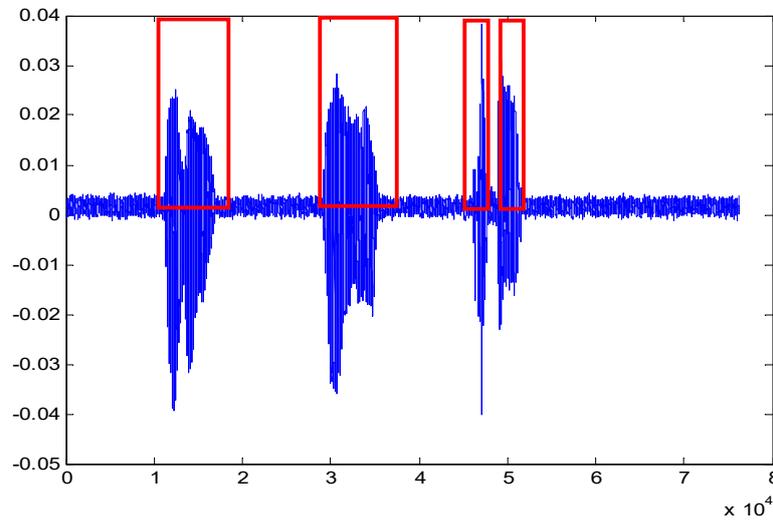


Figure II.2 : Energie du signal.

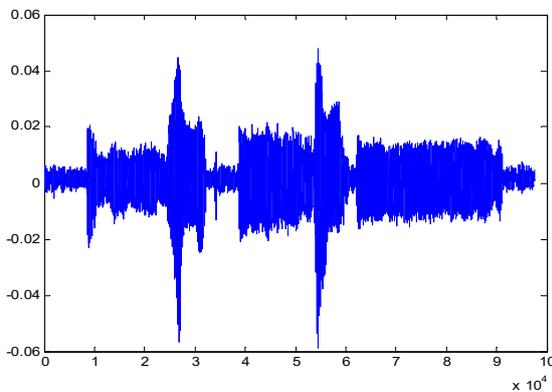
Seuil

Pour isoler la bande d'énergie utile représentant le mot, la première chose qui vient à l'esprit c'est de fixer un seuil pour l'énergie, dès que l'énergie du signal dépasse ce seuil, on a une détection d'Activité vocale en temps réel comme c'est présenté sur La **Figure II.3**.

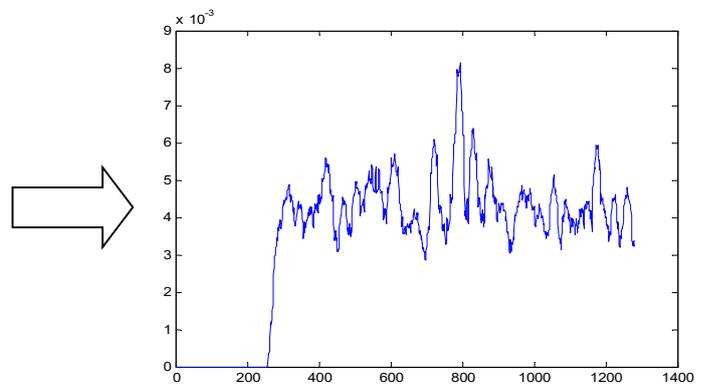


**Figure II.3 :** Fenêtrage de la parole par un seuil énergétique.

Ce qui n'est pas le cas comme pour la **Figure II.5** on ne peut pas trouver de repère sur ce spectre énergétique.



**Figure II.4 :** Bruit de fond + Parole.



**Figure II.5 :** Energie du signal.

## II.2.2. Détection d'activité vocale basée sur l'adaptation du spectre de bruit

Pour simplifier cette méthode se décompose en deux parties, une partie pour l'estimation statistique est une autre pour la prise de la décision **Figure. II.6**.

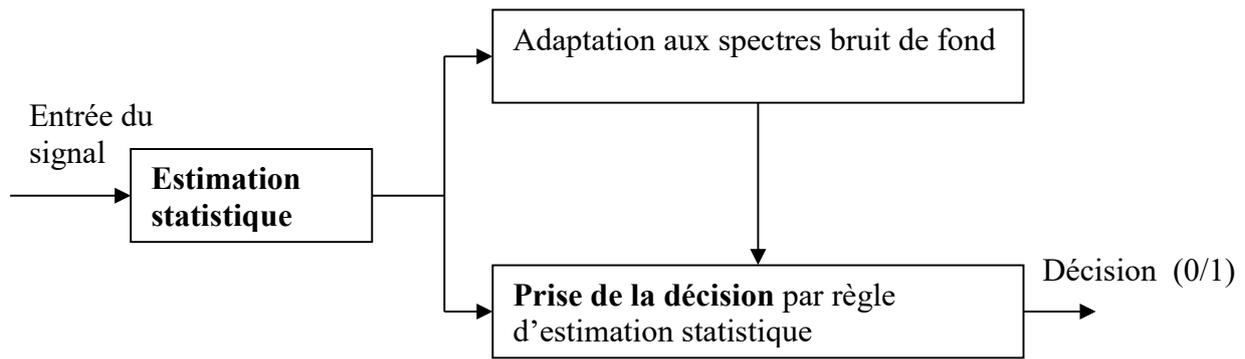


Figure II.6 : Déroulement global de l’algorithme de détection.

L’estimation statistique elle a pour but de prédire bruit de fond en calculant sa moyenne par l’espérance mathématique.

Lorsqu’on prend des échantillons du signal, la prise de décision va estimer si ça correspond ou pas à notre espérance dans le cas inverse c’est de la parole et tous cela doit ce faire dans le domaine fréquentielle pour avoir une meilleure sensibilité.

❖ Déroulement de la détection d’hypothèse

On va poser deux Hypothèses  $H_0$  le bruit de fond et  $H_1$  le bruit de fond + parole :

Hypothèse 0 ( $H_0$ ) :  $X(t) = N(t)$ .

Hypothèse 1 ( $H_1$ ) :  $X(t) = N(t) + S(t)$ .

Où N : Bruit, S : Signal vocal.

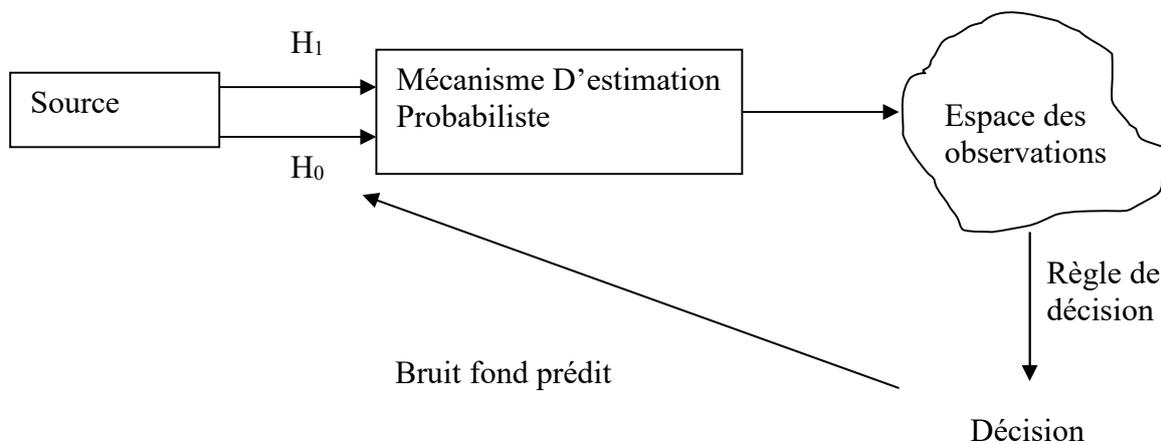


Figure II.7 : Déroulement de la détection d’hypothèse.

Le **Mécanisme d'estimation Probabiliste** va être estimé à l'aide du théorème de Bayes qui utilise la probabilité a priori pour estimer une hypothèse par rapport à une autre en connaissant une des deux hypothèses.

L'**Espace des observations** va permettre de mieux calibrer le système précédant on calculant l'estimation maximum de probabilité en d'autre terme la meilleur approche de la variance du signal.

La **Règle de décision** va être calculée par la vraisemblance.

La **Prédiction du bruit de fond** que l'on pourra estimer par l'espérance mathématique.

### II.2.3. Mécanisme d'estimation Probabiliste

Lors du démarrage du programme de la VAD dans notre application les premiers échantillons vont représenter le bruit de fond ce qui sera utilisé comme donnée initial pour l'application du théorème de Bayes.

#### II.2.3.1. Le théorème Bayes

En théorie des probabilités, le théorème de Bayes énonce des **probabilités conditionnelles** étant donné deux évènements  $A$  et  $B$ , le théorème de Bayes permet de déterminer la probabilité de  $A$  sachant  $B$ , si l'on connaît les probabilités : De  $A$ , De  $B$ , De  $B$  sachant  $A$ .

Le théorème de Bayes permet d'écrire [4] [5] [6].

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (\text{II.1})$$

$P(A|B)$  : Loi a posteriori.

$P(B|A)$  : Vraisemblance.

$P(A)$  : Loi a priori.

$P(B)$  : Evidence (constante de normalisation).

Plus généralement, si  $\{A_i\}$  est une partition de l'ensemble des possibles.

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_j P(B|A_j)P(A_j)}$$

### II.2.3.2. Application du théorème de Bayes aux échantillons du signal

Pour notre cas le signal utilisé est un signal de parole superposé à un bruit (bruit de fond) de manière épisodique. Ce qui induit deux hypothèses dans le domaine temporel.

$$\text{Hypothèse 0 } (H_0) : X(t) = N(t).$$

$$\text{Hypothèse 1 } (H_1) : X(t) = N(t) + S(t).$$

Comme k Fenêtre (une Fenêtre contient plusieurs échantillons à la fois) sont utilisées, la formule devient.

$$\text{Hypothèse 0 } (H_0) : X_k = N_k.$$

$$\text{Hypothèse 1 } (H_1) : X_k = N_k + S_k.$$

Pour L échantillons, dans une Fenêtre k, la transformé de fourrier discrète est utilisé et permet d'obtenir [7] :

$$\text{Hypothèse 0 } (H_0) : X_k(j\omega) = N_k(j\omega).$$

$$\text{Hypothèse 1 } (H_1) : X_k(j\omega) = N_k(j\omega) + S_k(j\omega).$$

Les deux hypothèses appliquées au théorème de Bayes induisent permet d'estimer la probabilité du bruit connaissant le signal.

$$P(H_0 | X_k) = \frac{P(X_k | H_0).P(H_0)}{P(X_k | H_0).P(H_0) + P(X_k | H_1).P(H_1)} \quad (\text{II.2})$$

La formule (II.2) peut fournir les formules (II.3), (II.4), (II.5).

$$P(H_0 | X_k) = \frac{P(X_k | H_0)}{P(X_k | H_0) + P(X_k | H_1)} \cdot \frac{P(H_0)}{P(H_0)}$$

$$P(H_0 | X_k) = \frac{P(X_k | H_0)}{P(X_k | H_0) + P(X_k | H_1)} \cdot \frac{P(H_0)}{P(H_0)}$$

$$P(H_0 | X_k) = \frac{1}{1 + \frac{P(X_k | H_1)}{P(X_k | H_0)} \cdot \frac{P(H_1)}{P(H_0)}} \cdot \frac{P(X_k | H_0)}{P(X_k | H_0)}$$

$$P(H_0 | X_k) = \frac{1}{1 + \frac{P(X_k | H_1) \cdot P(H_1)}{P(X_k | H_0) \cdot P(H_0)}} = \frac{1}{1 + \varepsilon \cdot \Lambda(k)} \quad (\text{II.3})$$

$$\varepsilon = \frac{P(H_1)}{P(H_0)} \quad (\text{II.4})$$

$$\Lambda(k) = \frac{P(X_k | H_1)}{P(X_k | H_0)} \quad (\text{II.5})$$

On peut remarquer que  $\varepsilon, \Lambda(k)$  représente **SNR** (Signal to Noise Ratio) le rapport du signal sur le bruit. [7]

$$SNR_k^{a \text{ priori}}(\omega) = \frac{|X_k(j\omega)|^2}{|N_k(j\omega)|^2} = \varepsilon$$

$$SNR_k^{a \text{ posteriori}}(\omega) = \frac{|S_k(j\omega)|^2}{|N_k(j\omega)|^2} = \Lambda(k)$$

$\varepsilon$  va symboliser SNR à priori et  $\Lambda(k)$  va symboliser SNR à posteriori ce qui résulte de l'équation (II.4) et (II.5) appliqué au théorème de Bayes, la présence de la parole dans le signal  $X$ .

$$P(H_1 | X_k) = \frac{\varepsilon \cdot \Lambda(k)}{1 + \varepsilon \cdot \Lambda(k)} \quad (\text{II.6})$$

A noter que  $\Lambda(k) = \frac{P(X_k | H_1)}{P(X_k | H_0)}$  c'est aussi ce que l'on appelle **Likelihood Ratio Test**

(LRT) par traduction c'est test des ratios probabilistes ou bien le terme en français le rapport de vraisemblance. [8]

$$P(H_0 | X_k) = \frac{1}{1 + \varepsilon \cdot \Lambda(k)} \quad (\text{II.7})$$

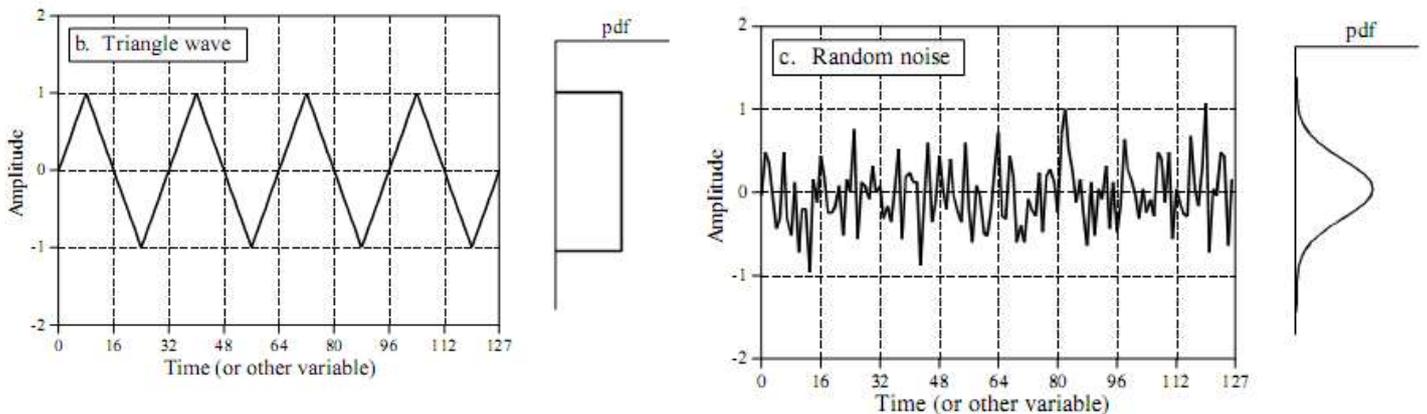
$$P(H_1 | X_k) = \frac{\varepsilon \cdot \Lambda(k)}{1 + \varepsilon \cdot \Lambda(k)} \quad (\text{II.8})$$

Le théorème de Bayes a deux cas, un cas discret est un cas continu [6] :

- \* Pour le cas discret en travail avec les classes  $C_k$ .
- \* Pour le cas continu en travail avec la densité de probabilité.

Alors pour notre cas  $P(H_0 | X_k), P(H_1 | X_k)$  ce sont des densités de probabilité [9] [7], mais avec quelle densité de probabilité ?, il y a plusieurs lois, loi binomiale, loi normale, loi log normale, loi gamma, loi bêta,....

Notre signal suit une loi normale (**gaussienne**) [9] [7], les **tests de normalité** permettent de vérifier si des données réelles suivent une loi normale ou non. Les tests de normalité sont des cas particuliers des **tests d'adéquation** (ou **tests d'ajustement**, tests permettant de comparer des distributions), appliqués à une loi normale. [10]



**Figure II.8** : Signal et fonction de densité de probabilité (pdf) associée

#### II.2.4. Espace des observations

Lorsqu'on prend plusieurs échantillons en sachant que c'est du bruit de fond on va estimer leur variance qui représente le degré de dispersion autour de la moyenne, l'estimation maximal de la variance va donner les limites du bruit de fond qui va être appliqué à notre modèle statistique qui est la loi normale.

##### II.2.4.1. Aperçu de la loi Normale

En probabilité, on dit qu'une variable aléatoire réelle  $X$  suit une **loi normale** (ou **loi normale gaussienne, loi de Laplace Gauss**) de moyenne  $\mu$  et d'écart type  $\sigma$  strictement positif (donc de variance  $\sigma^2$ ) si cette variable aléatoire réelle  $X$  admet pour densité de probabilité la fonction  $p(x)$  Définie, pour tout nombre réel  $x$ , par :

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Une telle variable aléatoire est alors dite variable gaussienne. [11] [12]

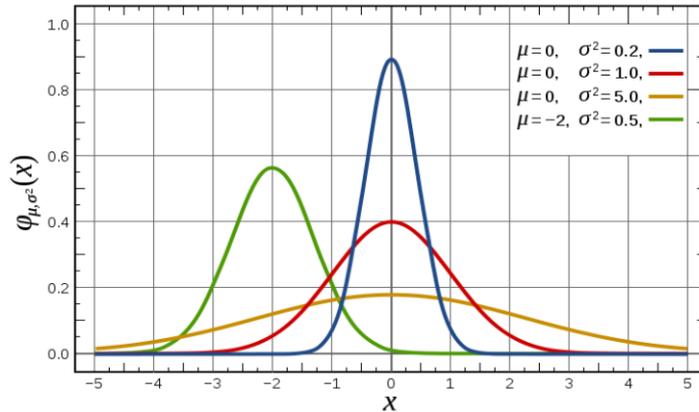


Figure II.9 : Distribution gaussienne.

### II.2.4.2. Application de la loi Normale au signale $X_k$

En appliquant la loi normale Aux deux hypothèses ( $H_0$  et  $H_1$ ) dans le domaine discret nous obtenant les équations suivant :

$$P(X | H_0) = \prod_{k=0}^{L-1} \frac{1}{\pi \cdot \lambda_N(k)} e^{-\frac{|X_k|^2}{\lambda_N(k)}} \tag{II.9}$$

$$P(X | \Theta, H_1) = \prod_{k=0}^{L-1} \frac{1}{\pi \cdot [\lambda_N(k) + \lambda_S(k)]} e^{-\frac{|X_k|^2}{\lambda_N(k) + \lambda_S(k)}} \tag{II.10}$$

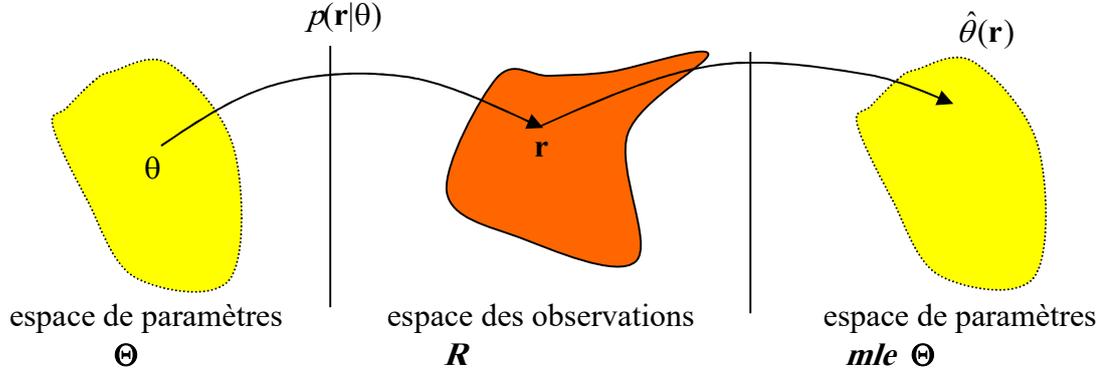
Où  $\Theta = \{\lambda_S(k) : k = 0, \dots, L-1\}$  est ensembles de paramètre inconnus conjoint à la densité de probabilité contenu dans l’hypothèse  $H_1$ .

$$P(X_1, X_2, \dots, X_n | \Theta) = P(X_1 | \Theta) \cdot P(X_2 | \Theta) \dots P(X_n | \Theta) = \prod P(X_i | \Theta)$$

#### ❖ Explication sur le paramètre inconnu conjoint à la densité de probabilité $\Theta$

Ce paramètre est introduit pour permettre de calculer l’estimation maximum de probabilité  $\hat{\Theta}$  (Maximum Likelihood Estimation - MLE) C’est une méthode courant dans les statistiques, elle est l’utilisées pour ajuster un modèle statistique des données, et de fournir des estimations pour les paramètres du modèle, comme pour notre model gaussienne il est

paramétré sur la variance est la moyenne c'est deux valeurs doive être fixe est la seul variable étant  $X_k$ , l'estimation maximum de probabilité va permettre de mieux calibré ces deux paramètre la moyenne et La variance. [13]



### II.2.4.3. Ajustement des paramètres du model (MLE)

L'estimation maximale représente la dérivé de la densité de probabilité par rapport à cette variable (variance) on va chercher l'estimation maximale de du bruit de fond dans les échantillons, le Logarithme népérien est utiliser pour simplifier les calcule (Log Likelihood).

$$P(X | \Theta, H_1) = \prod_{k=0}^{n-1} \frac{1}{\pi \cdot [\lambda_N(k) + \lambda_S(k)]} e^{-\frac{|X_k|^2}{\lambda_N(k) + \lambda_S(k)}} \quad (\text{II.10})$$

$$P(X | \Theta, H_1) = \frac{1}{(\pi \cdot (\lambda_N(k) + \lambda_S(k)))^n} e^{-\sum \frac{|X_k|^2}{\lambda_N(k) + \lambda_S(k)}}$$

$$\text{Ln}(P(X | \Theta, H_1)) = \ln\left(\frac{1}{(\pi \cdot (\lambda_N(k) + \lambda_S(k)))^n} e^{-\sum \frac{|X_k|^2}{\lambda_N(k) + \lambda_S(k)}}\right)$$

$$\text{Ln}(P(X | \Theta, H_1)) = l(\theta)$$

$$l(\theta) = -\ln(\pi \cdot (\lambda_N(k) + \lambda_S(k)))^n + \ln\left(e^{-\sum \frac{|X_k|^2}{\lambda_N(k) + \lambda_S(k)}}\right)$$

$$l(\theta) = -n \cdot \ln \pi \cdot (\lambda_N(k) + \lambda_S(k)) - \sum \frac{|X_k|^2}{\lambda_N(k) + \lambda_S(k)}$$

$$l(\theta) = -n \cdot \ln \pi - n \cdot \ln(\lambda_N(k) + \lambda_S(k)) - \sum \frac{|X_k|^2}{\lambda_N(k) + \lambda_S(k)}$$

$$\begin{aligned}
\frac{\partial l(\theta)}{\partial \sigma_N^2} &= 0 - n \left( \frac{1}{\lambda_N(k) + \widehat{\lambda}_S(k)} \right) - \sum \frac{|X_k|^2}{(\lambda_N(k) + \widehat{\lambda}_S(k))^2} = 0 \\
-n \left( \frac{1}{\lambda_S(k) + \widehat{\lambda}_N(k)} \right) &= \sum \frac{|X_k|^2}{(\lambda_S(k) + \widehat{\lambda}_N(k))^2} \Rightarrow -n(\lambda_S(k) + \widehat{\lambda}_N(k)) = \sum |X_k|^2 \\
\lambda_S(k) + \widehat{\lambda}_N(k) &= \frac{\sum |X_k|^2}{n} = |X_k|^2 \\
\Rightarrow \widehat{\lambda}_N(k) &= |X_k|^2 - \lambda_S(k) \tag{II.11}
\end{aligned}$$

Et  $\widehat{\lambda}_N(k)$  est une estimation maximale de la variance (MLE) du bruit de fond ce résultat va être utilisé pour le calculé du Test d'hypothèse généralisé (vraisemblance généralisé) par conséquent réduit, car en utilise l'estimation maximale de la variance. [13]

### II.2.5. Règle de décision

Le rapport de vraisemblance généralisé est une approche à la résolution du problème de décision quand on ne peut pas construire un test uniformément le plus puissant.

Cette procédure consiste à associer à chaque hypothèse  $H_i$  une densité unique, parmi toutes les densités possibles, en prenant celle qui maximise la vraisemblance des données, et à l'utiliser pour construire un "rapport de vraisemblance réduit":

$$\Lambda_g(\mathbf{r}) = \frac{\max_{\theta} P(X | \widehat{\Theta}, H_1)}{\max_{\theta} P(X | H_0)} \begin{matrix} > \\ < \end{matrix} \gamma \begin{matrix} H_1 \\ H_0 \end{matrix} \tag{II.12}$$

La vraisemblance réduit est obtenu On substituant le résultat obtenu précédemment  $\widehat{\lambda}_N(k) = |X_k|^2 - \widehat{\lambda}_S(k) \Rightarrow \widehat{\lambda}_S(k) = |X_k|^2 - \widehat{\lambda}_N(k)$  dans l'équation (II.10)  $P(X | \Theta, H_1)$  nous obtiendrons  $P(X | \widehat{\Theta}, H_1)$  par conséquence.

$$\Lambda_g = \frac{1}{L} \ln \frac{P(X | \widehat{\Theta}, H_1)}{P(X | H_0)} \quad . [9]$$

$$\begin{array}{c}
 H_1 \\
 \ln \Lambda(k) \begin{array}{c} > \\ < \end{array} \ln \gamma = \eta \\
 H_0
 \end{array}$$

$$\Lambda_g = \frac{1}{L} \sum_{k=0}^{L-1} \left\{ \frac{|X_k|^2}{\lambda_N(k)} - \ln \frac{|X_k|^2}{\lambda_N(k)} - 1 \right\} \begin{array}{c} > \\ < \end{array} \eta \quad (II.13)$$

$$\eta = \ln(\gamma)$$

Où  $\gamma$  définit un seuil pour basculé d'une hypothèse à l'autre ( $0 \leq \gamma \leq 1$ ), ce qui a par conséquence aussi de mieux estimé les densités de probabilité.

$$P(H_0 | X_k) = \frac{1}{1 + \varepsilon \cdot \Lambda_g(k)} \quad (II.14)$$

$$P(H_1 | X_k) = \frac{\varepsilon \cdot \Lambda_g(k)}{1 + \varepsilon \cdot \Lambda_g(k)} \quad (II.15)$$

### II.2.6 La prédiction du bruit de fond

A une faible variation de la variance, l'espérance mathématique est égale à cette variance d'après le théorème de l'erreur quadratique moyenne (a **minimum mean square error (MMSE)** estimator).

$$E[X] = \sum_j E[X | B_j] P(B_j) = \widehat{\text{var}}(x) \quad \text{Estimation maximum de la variance de } x \text{ elle}$$

représente la meilleur approche pour l'estimation de bayes, on choisit comme variable X la variance du bruit de fond  $\lambda_N$  que nous allons prédire par la formule suivant :

$\widehat{\lambda}_N(k) = E(\lambda_N(k) | X_k)$  Où  $E(\lambda_N(k) | X_k)$  est l'espérance mathématique et  $\widehat{\lambda}_N(k)$  est l'estimation maximum de la variance du bruit de fond, ce qui donne pour la variance du bruit de fond :

$$\widehat{\lambda}_N(k) = E(\lambda_N(k) | X_k)$$

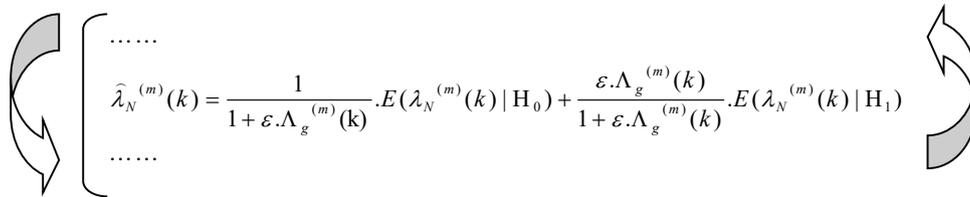
$$\widehat{\lambda}_N(k) = E(\lambda_N(k) | H_0) \cdot P(H_0 | X_k) + E(\lambda_N(k) | H_1) \cdot P(H_1 | X_k)$$

En utilisant les résultats obtenus précédemment (II.14) et (II.15) on obtient :

$$E(\lambda_N(k) | X_k) = \frac{1}{1 + \varepsilon \cdot \Lambda_g(k)} \cdot E(\lambda_N(k) | H_0) + \frac{\varepsilon \cdot \Lambda_g(k)}{1 + \varepsilon \cdot \Lambda_g(k)} \cdot E(\lambda_N(k) | H_1)$$

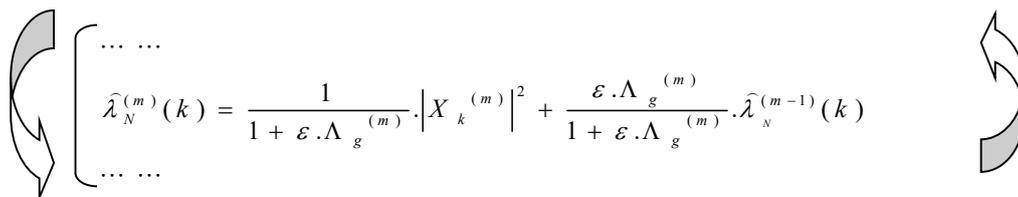
$$\widehat{\lambda}_N(k) = \frac{1}{1 + \varepsilon \cdot \Lambda_g(k)} \cdot E(\lambda_N(k) | H_0) + \frac{\varepsilon \cdot \Lambda_g(k)}{1 + \varepsilon \cdot \Lambda_g(k)} \cdot E(\lambda_N(k) | H_1)$$

Cette formule va estimer au mieux le bruit de fond est de le mettre à jour, cette formule va être exécuté à chaque fenêtre ‘m’ (frame) [9]. Par conséquent on va remplacer  $\widehat{\lambda}_N(k)$  par  $\widehat{\lambda}_N^{(m)}(k)$  et  $E(\lambda_N(k) | H_1)$  par  $E(\lambda_N^{(m)}(k) | H_1)$  et  $E(\lambda_N(k) | H_0)$  par  $E(\lambda_N^{(m)}(k) | H_0)$ .



$$\widehat{\lambda}_N^{(m)}(k) = \frac{1}{1 + \varepsilon \cdot \Lambda_g^{(m)}(k)} \cdot E(\lambda_N^{(m)}(k) | H_0) + \frac{\varepsilon \cdot \Lambda_g^{(m)}(k)}{1 + \varepsilon \cdot \Lambda_g^{(m)}(k)} \cdot E(\lambda_N^{(m)}(k) | H_1)$$

Et lorsque on n’a pas de parole on peut remplacer  $E(\lambda_N^{(m)}(k) | H_0)$  par  $|X_k^{(m)}|^2$ , aussi  $E(\lambda_N^{(m)}(k) | H_1)$  par  $\widehat{\lambda}_N^{(m-1)}(k)$ .



$$\widehat{\lambda}_N^{(m)}(k) = \frac{1}{1 + \varepsilon \cdot \Lambda_g^{(m)}(k)} \cdot |X_k^{(m)}|^2 + \frac{\varepsilon \cdot \Lambda_g^{(m)}(k)}{1 + \varepsilon \cdot \Lambda_g^{(m)}(k)} \cdot \widehat{\lambda}_N^{(m-1)}(k)$$

Est cette formule va permettre de mettre à jour la prédiction du bruit de fond dès que y a pas de parole. [9]

## II.2.7. Organigramme de la VAD

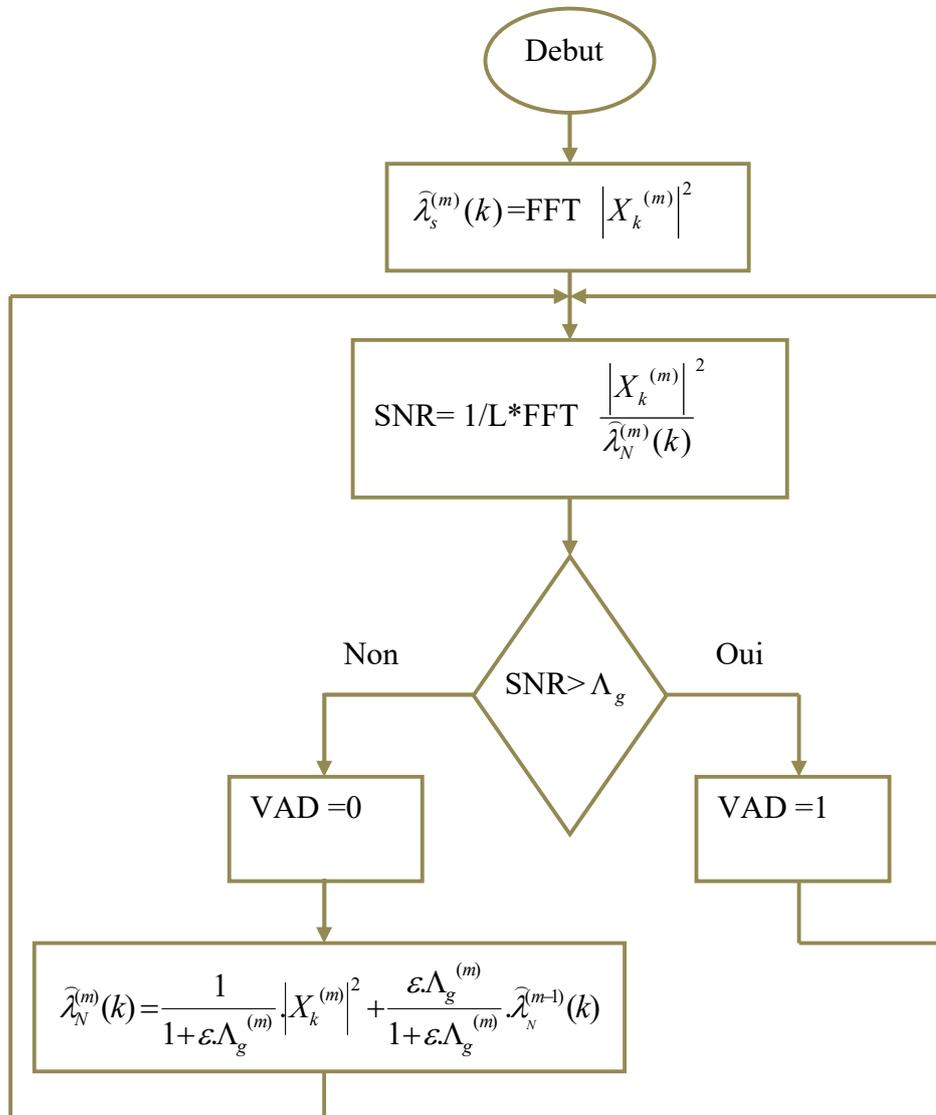


Figure II.11 : Organigramme de la VAD.

## II.3. La Déformation Temporelle Dynamique DTW

Il s'agit d'une méthode de comparaison permettant de tenir compte de la distorsion qui a pu être engendrée par une prononciation déferente du mot et qui cherche à calculer le chemin le plus court entre les informations courantes et les informations de référence en cherchant la déformation temporelle minimale. la DTW est un algorithme qui vise à régler les inconvénients de l'algorithme du LTD « déformation temporelle linéaire ».

### II.3.1. Déformation temporelle linéaire

Soit  $X$  et  $Y^k$  deux séquences, le premier représentant le signal vocal à reconnaître et le second représentant le signal vocal de référence, deux cas possible peuvent se présenter la première, est que les deux séquences sont de même dimension ( $N$ ). Et dans ce cas il sera facile de définir la distance globale comme des distances locales :

$$D(X, Y^k) = \sum_{n=1}^N d(x_n, y_n^k) \quad (\text{II.16})$$

$$d(x_n, y_n^k) = \sqrt{\sum_{i=1}^l (x_{ni} - y_{ni}^k)^2} \quad (\text{II.17})$$

D: distance global.

d : distance locale.

Et la deuxième, est que si les deux séquences aient des longueurs différentes et la comparaison tranche par tranche n'est pas possible. La solution la plus simple (comme souvent utilisée dans les années 1960 1970) consiste alors à faire une normalisation linéaire de l'axe du temps. Comme illustre à la **figure II.11**. Cette solution consiste simplement à associer plusieurs vecteur de référence à un seul vecteur d'entrée (ou inversement, selon la séquence la plus longue).

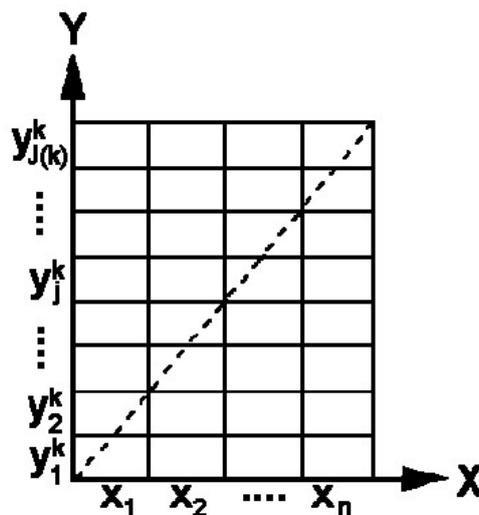
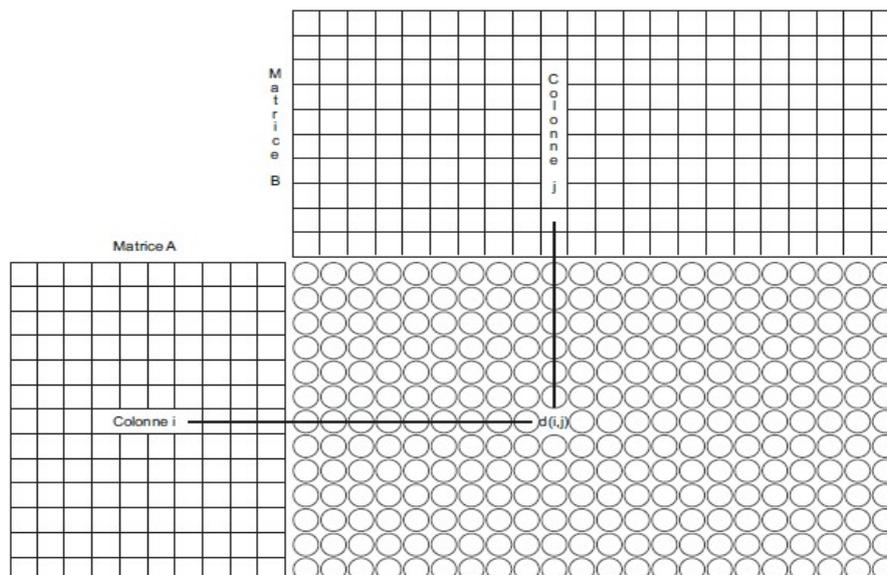


Figure II.12 : Déformation temporelle linéaire.

Malheureusement, ceci n'est certainement pas la solution optimale. Même si les deux prononciations ont la même longueur, il est probable que la meilleure solution ne consiste pas simplement à additionner les distances entre tranches. En effet, il est possible que certaines parties (phonèmes) du mot test soient prononcées plus rapidement que pour le mot de référence, alors que d'autres sections sont prononcées plus lentement. Ce problème nécessite une comparaison dynamique. [2]

### II.3.2. Le principe de la DTW

La DTW (déformation temporelle dynamique) sert à obtenir une mesure de dissemblance entre deux matrices, c'est-à-dire permet d'estimer si deux matrices sont plus ou moins différentes. Cette mesure pourrait se rapprocher d'une distance, mais la propriété de symétrie n'est pas satisfaite, et l'inégalité triangulaire est rarement utile en reconnaissance de la parole. On parle donc bien d'une mesure de dissemblance, cet algorithme a donc la capacité de comparer deux matrices de taille respectives  $(l,m)$  et  $(l,n)$ , on note  $d(i,j)$  la distance euclidienne entre la colonne  $i$  de  $A$  et la colonne  $j$  de  $B$  ainsi on pratique pour comparer deux matrices  $A$  et  $B$  de taille respectives  $(l,m)$  et  $(l,n)$ , on remplit tout d'abord une matrice  $C$  de taille  $(m,n)$ , où un élément à la ligne  $i$  et à la colonne  $j$  représente la distance euclidienne entre la colonne  $i$  de  $A$  et la colonne  $j$  de  $B$  ( $d(i,j)$ ) (**figure II.13**).

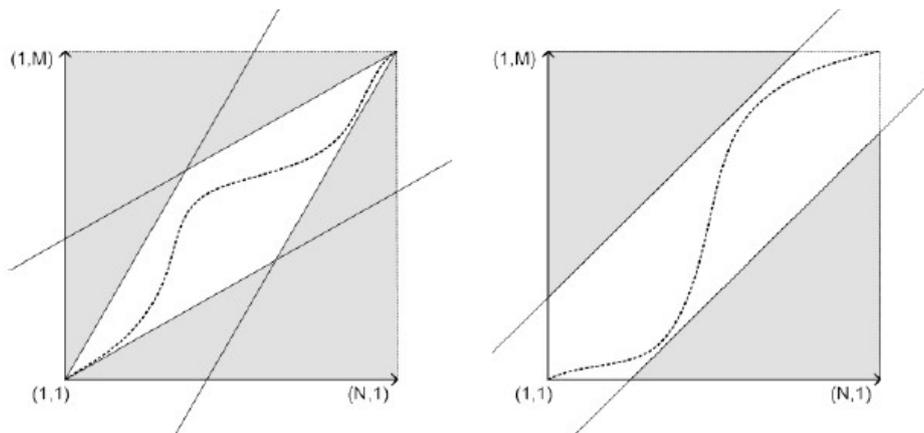


**Figure II.13** : Matrice des distances.

Où  $A_i(l)$  est l' $i$ -ième composante de la colonne  $i$  de  $A$ . En suite, on construit une quatrième matrice  $D$  dont chaque élément est obtenu par récurrence en fonction d'une contrainte choisie auparavant et des distances contenues dans  $C$ .

Cet algorithme est généralement applicable à des cas généraux (à condition que le point de départ soit distinct du point d'arrivée), dans le cas de la reconnaissance de la parole et étant donné les caractéristiques physiques associées, ce dernier doit être optimisé et pour le faire il est nécessaire de solliciter la présence de l'une des contraintes suivantes :

- ❖ Certaine contrainte de mono-cité du chemin, typiquement partant des coordonnées  $(1,1)$  et se terminant en  $(m,n)$  correspondant à la fin de la matrice  $C$  dans notre cas.
- ❖ Contraintes globales : parmi elles, des contraintes qui permettent de réduire l'espace de recherche (en imposant que le chemin optimal reste dans une zone déterminée, proche de la diagonale) comme illustré dans la figure suivante.



**Figure II.14 :** Établissement d'une zone de recherche.

- ❖ Contraintes locales : les prédécesseurs  $p(i,j)$  sont limités à quelle que soit l'élément proche et garantissant un chemin strictement gauche-droit comme illustré dans la figure ci-dessous. [17]

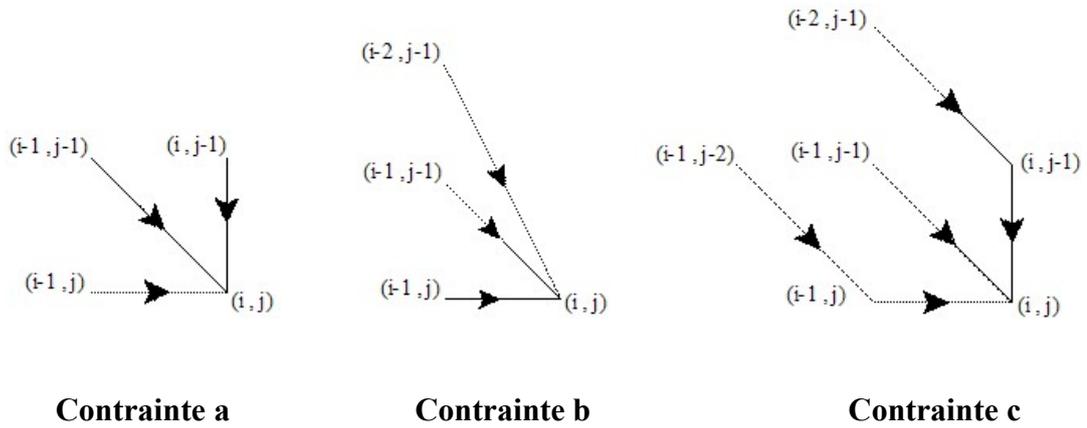


Figure II.15 : Les contraintes locales.

Nous allons utiliser la contrainte locale définie comme suit :

$$D(1,1) = d(1,1)$$

$$D(i,j) = \min \begin{cases} d(i,j) + D(i-1,j) \\ 2*d(i,j) + D(i-1,j-1) \\ d(i,j) + D(i,j-1) \end{cases} \quad \text{(II.18)}$$

Représenter dans la figure ci-dessous :

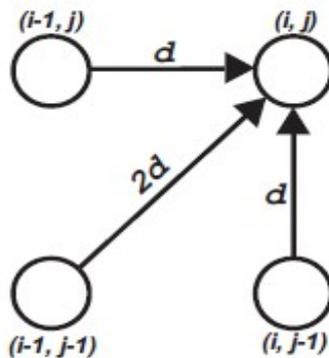


Figure II.16 : Contrainte de parcours.

Sauf pour la première ligne et la première colonne nous n'allons pas utiliser la contrainte définie dans l'équation (II.18), sinon les indices  $i-1$  et  $j-1$  sortiraient de la matrice. Ces contraintes deviennent :

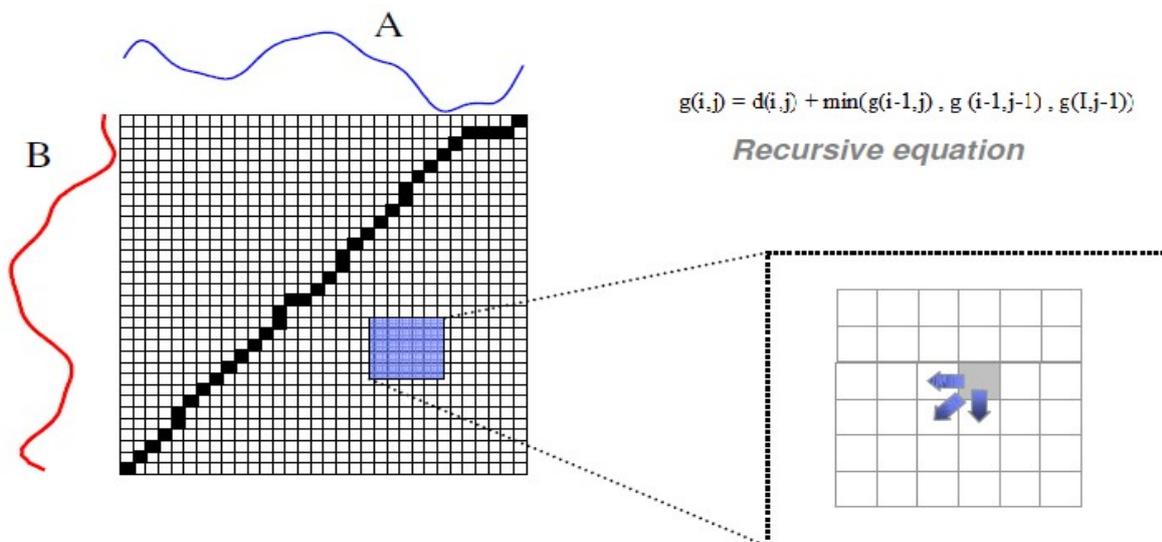
$$\begin{cases} D(i,1) = D(i-1,1) + d(i,1) & \forall i > 1 \\ D(1,j) = D(1,j-1) + d(1,j) & \forall j > 1 \\ D(1,1) = d(1,1) \end{cases} \quad (\text{II.19})$$

Nous aurions pu garder la même contrainte pour tous les calculs en insérant une première ligne et une première colonne d'infinis, le nombre que l'on cherche à obtenir pour évaluer la dissemblance entre A et B est obtenu par l'équation suivante :

$$DTW = \frac{D(m,n)}{m+n} \quad (\text{II.20})$$

**Exemple**

Voici un petit exemple récapitulatif de ce qu'est dit précédemment afin de comprendre mieux comprendre la DTW.



**Figure II.17 :** Comparaison entre deux signaux A et B par DTW.



## II.5. Difficulté et implémentation

La difficulté réside surtout dans la compréhension de l'algorithme et dans la mise en place des contraintes choisis.

De la se profilent 3 tapes :

- obtention des deux matrices ainsi que leurs dimensions respectives.
- puis remplissage des premières lignes et première colonne.
- enfin récurrence sur tous les éléments de D compris dans la bande. [15]

## II.6. Conclusion

Dans ce chapitre nous avons développés les algorithmes qu'on va utiliser, en premier lieu nous avons introduit la détection d'activité vocale, et on a vue que notre algorithme est basée sur l'estimation du bruit du fond, et on a constaté que l'utilisation d'une estimation statistique basé sur le théorème de bayes a permis de mieux estimé le bruit de fond en se basant sur un pourcentage de vraisemblance qui a permis de donné plus de souplesse au programme en s'adaptant a plusieurs type de bruit de fond l'étude approfondi du problème a permis d'avoir plus de précision sur des échantillons variable, en second lieu on a développai la DTW qui est l'algorithme responsable de la reconnaissance du mot isolé.

Après la reconnaissance du mot on activera une action correspondante a ce mot qui sera détaillée dans le chapitre suivant qui traite de la transmission des données du PC vers la carte de commande basé sur le microcontrôleur 68HC11.

### III.1. Introduction

Pour pouvoir réaliser la détection de l'activité vocale avec l'algorithme VAD afin de faire de la reconnaissance du mot isolé, nous avons utilisé l'algorithme DTW associé à une carte d'interface basée sur le microcontrôleur 68HC11 permettant d'activer les actionneurs selon un protocole de communication spécifique aux deux éléments que nous présentons.

### III.2. Description général du microcontrôleur 68HC11

Le microcontrôleur Motorola 68HC11 peut fonctionner avec des horloges allant jusqu'à 12 MHz. Tous les registres étant statiques, une coupure d'horloge n'entraîne pas de perte de donnée.

La plupart des fonctions périphériques sont intégrées sur la puce : un convertisseur analogique/digital, multiplexé sur 8 canaux, permet le traitement des grandeurs analogiques sur 8 bits, le circuit d'interface asynchrone (SCI), du type RS 232, ainsi qu'un port synchrone (SPI) permettent les communications avec le mode extérieur disposant d'un capteur sur 16 bits, HC11 apportent la souplesse nécessaire à la générateur d'impulsion ainsi que la mesure de temps avec la précision donnée par le quartz de l'horloge de base. Les différents sous fonction du HC11 sont montés par le schéma bloc du MC68HC11A1 donnée dans la **Figure III.1**. Les bloc sont validés ou inhibé selon le type de produit choisi, le système pouvant être étendu par le type de périphériques RAM/EPROM utilisé.

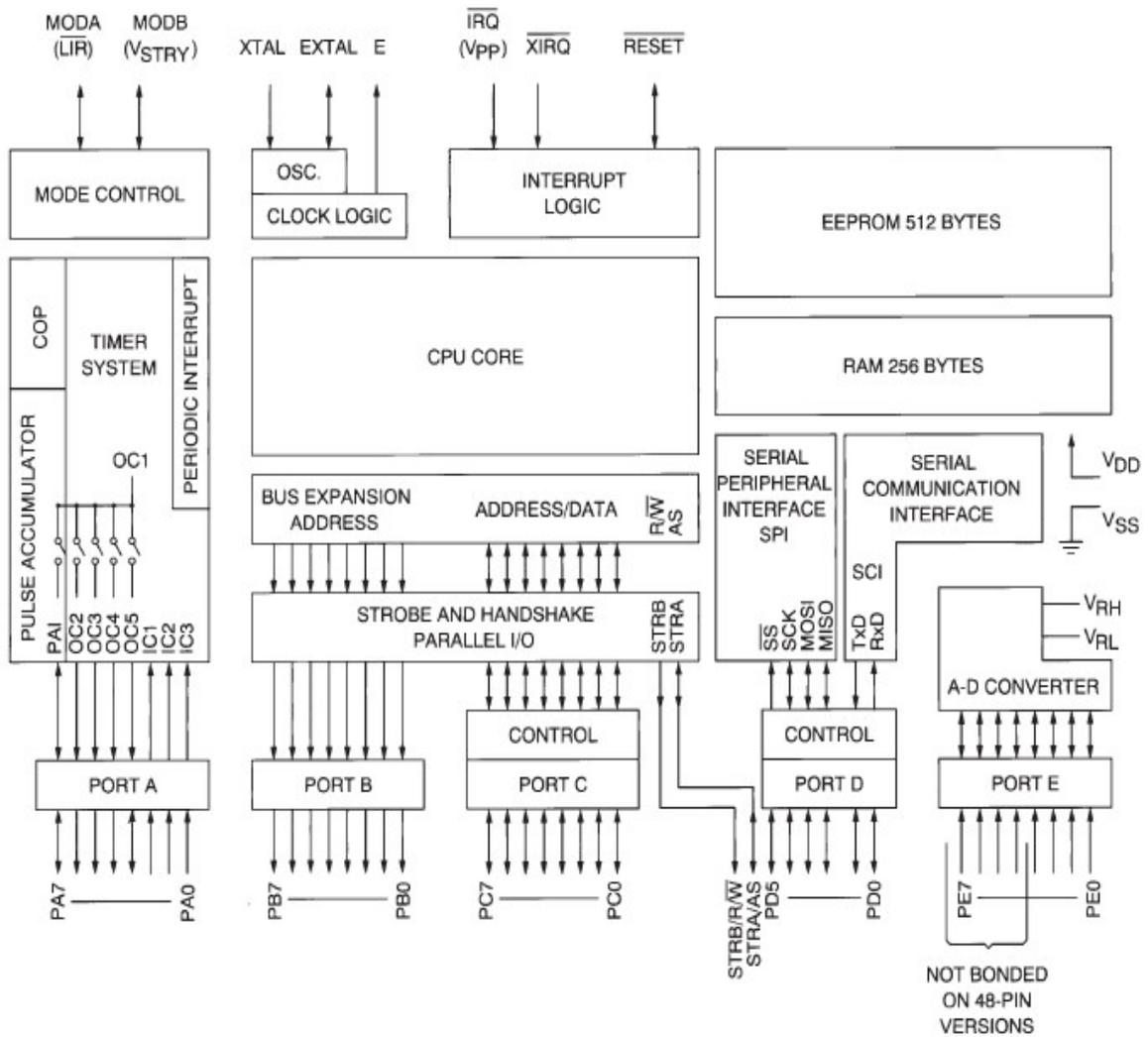


Figure III.1 : Schéma interne du 68HC11A1.

Les circuits de la famille 68HC11 disposent de moyen de protection internes qui permettent de garder le contrôle du microprocesseur dans l'éventualité d'une défaillance système ou logicielle. La fonction **COP** assure la surveillance de l'exécution correcte du programme, l'exécution de code illégal étant détectée par une interruption  $\overline{XIRQ}$ . Le MCU effectue un reset automatique dans le cas où l'horloge du système tombe à un seuil défini par le programmeur.

- **Les ports**

Il existe plusieurs ports dans le  $\mu\text{C}$  68HC11 dont le port D qui est responsable de la transmission série et le port B pour l'activation des actionneurs.

- **Port D**

Le port D est un port d'entrées sorties à usage général partagé avec les interfaces séries asynchrones (SCI) et synchrones (SPI), tant que ces interfaces ne sont pas utilisées, le sens de travail des lignes du port est programmé par les bits correspondants du registre DDRD.

### **III.3. Transmission des données**

Après qu'on a reconnue le mot isolé il nous faut envoyer la commande vers notre carte de commande, pour ce la on va utiliser le port de transmission série RS 232 et définir le mode de transmission série qu'il convient ainsi le branchement de se port avec le PC.

#### **III.3.1. Le protocole série**

Au contraire de la transmission parallèle qui se fait par paquet de bits, la transmission série se fait bit par bit avec une même fréquence de transmission c'est pourquoi cette dernière est plus lente que la transmission parallèle mais en contre partie elle nous permet de transférer des données a des distances plus importante que celle de la transmission parallèle allons jusqu'à plusieurs dizaines de mètre et elle est moins couteuse.

Étant donné que dans la transmission série, il y a qu'un seul fil qui transporte l'information, il existe un problème de synchronisation entre émetteur et récepteur, c'est-à-dire que le récepteur ne peut pas a priori distinguer les séquences de bits car ils sont envoyés successivement, il existe donc deux types de transmission série permettant de remédier à ce problème :

- **La transmission sériée synchrone**

Dans une transmission synchrone, les bits sont émis d'une façon régulière, sans séparation entre les caractères, pour cela un signal d'horloge périodique de période T, fonctionne pendant toute la durée de l'émission, de plus des informations supplémentaires sont insérées afin de garantir l'absence d'erreurs lors de la transmission.

Le principal inconvénient de la transmission synchrone est la reconnaissance des informations au niveau du récepteur, car il peut exister des différences entre les horloges de l'émetteur et du récepteur, c'est pourquoi chaque envoi de données doit se faire sur une période assez longue pour que le récepteur la distingue. Ainsi, la vitesse de transmission ne peut pas être très élevée dans une liaison synchrone. [16]

- **La transmission sériée asynchrone**

En transmission de données asynchrone, un bit de **Start** est toujours envoyé en tête par marquer le début d'un caractère. Les bits **de donné du caractère** suivent le bit Start dans l'ordre, en commençant par le bit le moins significatif et terminant par le plus significatif et le récepteur est prévenu de l'arrivée du caractère par le bit Start, il se synchronise alors sur le signal qui arrive.

Sur de nombreuses liaisons séries asynchrones, afin de disposer d'un moyen simple de contrôle de l'exactitude de la transmission, on introduit la notion de parité grâce à l'envoi d'un bit supplémentaire réservé à cet effet, le principe de ce bit de **parité** est le suivant : on compte les bits utiles du caractère transmis (c'est à dire les bits du caractère transmis seulement et non le « Start » et le ou les « stops ») et selon que leur nombre est pair ou impair, on met à 1 ou à 0 le bit de parité. Ce travail est fait à l'émission et le bit de parité est donc émis comme faisant partie du caractère transmis ainsi que la montre la figure. A la réception, on refait le même compte et on compare la parité ainsi obtenue avec celle reçue, si elles sont différentes, il y a certainement eu erreur dans la transmission (due à un parasite qui a fait changer un bit d'état par exemple), s'il y a identité, cela ne veut pas dire que la liaison s'est bien passée à 100 % car deux bits peuvent avoir changé d'état en sens contraire simultanément, mais cela donne tout de même une indication relative quant à sa qualité cette parité peut être paire ou impaire. En **parité paire**, le bit de parité est positionné pour que le nombre total de bits à 1 (y compris celui de parité) soit pair, et en **parité impaire** pour qu'il soit impaire.

Les dernier bits transmit à être transmit sont les bits de **stop**, ce bit ne contient aucune information mais elles donnent au récepteur le temps d'assemblé les bits sériés qui viennent d'être reçu et de se préparer pour le caractère suivant, les bits de stop dur de 1 à 2 duré de bits **Figure III.2**.

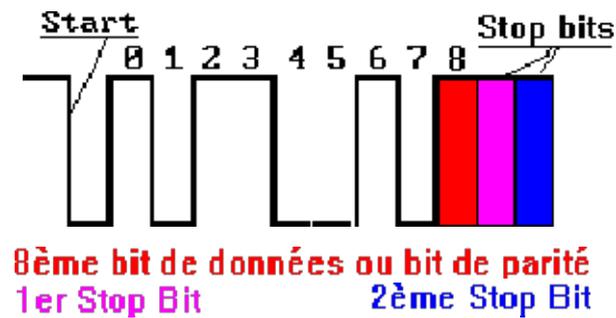


Figure III.2 : Format de données.

Les fréquences de bits normalisés pour les liaisons asynchrones sont: 50, 75, 110, 134, 5, 150, 300, 600, 1200, 2400, 3600, 4800 et 9600 bits par second. [17]

Dans notre application le récepteur sera le microcontrôleur 68HC11 qui communiquera avec le PC grâce à son interface série asynchrone le SCI qu'on va définir dans ce qui va suivre.

### III.4. L'interface SCI

L'interface de communication SCI est une interface de type asynchrone, elle est de type UART classique, tout en disposant de certains perfectionnements, elle possède les caractéristiques suivantes :

- Standard NRZ (Non Retour à Zéro) full duplex ; un bit de départ, 8 ou 9 bit de données, un bit de stop, avec ou sans parité.
- Les fréquences standard sont fabriquées par le 68HC11 à partir de l'horloge.
- Positionnement d'indicateurs dans le cas d'une mauvaise réception.
- Erreur de format, erreur due au bruit.
- Echanges aux niveaux TTL, il conviendra, dans le cas d'une liaison du type RS 232, de prévoir des translaters inverseurs (ex : MAX 232) externes.
- Mode réveil automatique (utile dans le cas où plusieurs récepteurs sont connectés sur la même liaison).
- Lorsque l'interface de communication est utilisée, les broches PD0 et PD1 sont dédiées à cette fonction.
- PD0 : est l'entrée de réception de données RxD.
- PD1 : est la sortie de transmission de données TxD. [18]

### III.4.1. Fonctionnement en Emission

Le transmetteur du SCI inclut un registre de données parallèle (SCDR) et un registre à décalage pour la transmission série, le contenu du registre à décalage pour la transmission série ne peut être écrit que par le biais du registre SCDR.

Cette opération permet le transfert d'un caractère sur le port série pendant que le caractère suivant est en attente dans le registre SCDR, la sortie du registre à décalage est appliquée à la broche TxD tant que la transmission est en cours, ou que le bit TE du registre de contrôle SCCR2 est mis à 1.

### III.4.2. Fonctionnement en réception

Lors des opérations de réception, la séquence de transmission est inversée, le registre à décalage du port série reçoit les données et les transfère dans le registre parallèle dédié au port série (SCDR) sous la forme d'un mot complet, cette opération permet de lire un caractère sur le registre à décalage tandis que le caractère précédant est disponible dans le registre SCDR. Pour recevoir la donnée on utilise la ligne IDLE :

#### ➤ Réveil par la ligne d'attente (IDLE LINE)

Lors d'un réveil par la ligne d'attente, le récepteur qui es en veille est activé dès que la ligne RxD passe en mode attente (début d'un message), tous les récepteurs sont activés (RWU=0) dès que chaque message commence, dès qu'un récepteur a déterminé qu'un message ne lui est pas adressé, il remet le bit RWU à 1 en d'autre mots (SCCR2.2=1), ce qui inhibe la mise à un des indicateur SCI jusqu'au retour au niveau IDLE (SCSR.5=1) de la ligne RxD en fin du message en cours, ainsi les caractères suivants du message ne se sont pas pris en compte par le récepteur.

Dont voici la routine correspondante que nous avons appliquée dans notre application

```

#include "startfl.bas"
byte c
  DDRA=255      'initialiser le port a en sortie'
  PORTA=0
  BAUD.5=1      'selectionner la vitesse de baud
  BAUD.4=1      'à 9600'
  BAUD.2=0
  BAUD.1=0
  BAUD.0=0
do
  do
  loop until SCSR.5=1  La ligne RxD passe en mode attente (début d'un message)
  'SCCR2.2=1          Inhibition de tous les récepteur

  c=SCDR
  PORTB=c  'l'envoi de ces signaux sur le port B
loop

```

#### III.4.4. Registre du port série

La liaison série asynchrone (SCI) dispose de 5 registres adressables :

❖ 3 registres de contrôle :

- SCCR1: registre de contrôle N°1 du SCI, les bits du registre SCCDR1 permettent de définir la longueur du mot transmis sur le port série ainsi que la méthode devant être utilisée pour effectuer l'activation du port.

- SCCR2 : registre de contrôle N°2 du SCI, les bits de ce registre permettent de définir les bits de contrôle pour valider ou inhiber les fonctions SCI individuelles.

- BAUD : registre de définition BAUD du SCI, ce registre contrôle la vitesse de transfert du SCI :

❖ 1 registre contenant les données du SCI :

- SCDR : registre de données du SCI : contient la donnée transmise ou émise

❖ 1 registre d'état du SCI :

- SCSR : registre de statut du SCI : condition pour la génération des demandes d'interruption du système SCI.

Comme on va communiquer avec l'interface série asynchrone le SCI alors il faut choisir un support de transmission qui convient à ce type de transmission. [19]

### III.5. La liaison RS 232

La carte à microcontrôleur est couplée avec l'ordinateur, par une liaison série asynchrone à travers un connecteur indiqué dans la **Figure III.3** permettant le transfert de l'information et des ordres de contrôle, RS est l'abréviation de « **R**ecommanded **S**tandard », actuellement nommé (Electronic Industries Association) d'où le EIA 232 des années 1990, a développé un standard d'interface de transmission de donnée en série entre équipement, entre les ordinateur ou entre pc et équipement.



**Figure III.3** : Connecteur RS 232.

#### III.5.1. Caractéristiques et performance

- Mot transmis de 5,6 7 ou 8 bits.
- 1 bits de Start (0) et 1, 1.5 ou 2 bits de stop (1) indiquent le début et la fin du mot transmis.
- 1 bit de parité (paire ou impaire) est éventuellement ajouté pour détecter les erreurs de transmissions.
- Transmission à 75, 150, 300, 600, 1200, 2400, 4800, 9600, 19200,....115.2 Kbit/s (appelé aussi Baud)
- Half duplex (émission/réception alternées) ou Full-duplex (émission/réception simultanées).
- Pour transmettre 8 bits utiles : de 10 à 12 bits sont envoyés.
- **Caractéristiques électriques**

- '0' niveau émis +V.

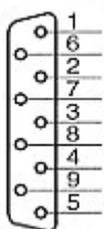
- '1' niveau émis -V.

Avec V compris entre 5 et 25 Volts (12 Volts pour les ports séries COM1, COM2 sur un PC).

Transmission limitée à une dizaine de mètres maximum, théoriquement jusqu'à 15 mètres. [20]

- **Caractéristiques physiques**

Selon la norme, elle comporte 22 fils - 23 dans sa déclinaison RS-232C, ces fils sont placés de manière standard sur une broche DB25 (  ). Or, lorsqu'IBM créa son PC en 1981, Intel lui fournissait un circuit (le 8250) capable de ne gérer directement que 9 de ces fils, qu'à cela ne tienne : les PC n'implémentent en standard qu'une sous-version de la norme complète, limitée à ces 9 fils, pour des soucis de simplicité, d'économie et d'encombrement, ces 9 fils furent positionnés non plus sur une DB25, mais sur une DB9 (  ). Qui est représenté sur la **Figure III.4** avec toutes les significations des broches. [21]



| N° de broche DB 9 | Nom US | Nom EIA | N° CCITT | Fonction                 |
|-------------------|--------|---------|----------|--------------------------|
| 1(entrée)         | DCD    | CF      | 109      | Détection de porteuse    |
| 2(entrée)         | RD     | BB      | 104      | Réception de données     |
| 3(sortie)         | TD     | BA      | 103      | Emission de données      |
| 4                 | DTR    | CD      | 108      | Terminal de données prêt |
| 5(masse)          | SG     | AB      | 102      | Masse électrique         |
| 6(entrée)         | DSR    | CC      | 107      | Poste de données prêt    |
| 7(sortie)         | RTS    | -       | 105      | Demande d'émission       |
| 8(entrée)         | CTS    | CB      | 106      | Prêt à émettre           |
| 9                 | RI     | CE      | 125      | Indicateur de sonnerie   |

**Figure III.4** : Schéma d'un connecteur DB9 et signification des broches.

- **Fonctionnalité des broches (DB9)**

**Broche 1 :** DCD (**Data Carrier Detect**) cette ligne est une entrée active à l'état haut, elle signale à l'ordinateur qu'une liaison a été établie avec un correspondant.

**Broche 2 :** RxD (**Receive Data**) cette ligne est une entrée, c'est ici que transitent les informations du correspondant vers l'ordinateur.

**Broche 3 :** TxD (**Transmit Data**) cette ligne est une sortie, les données de l'ordinateur sont véhiculées par son intermédiaire vers son correspondant.

**Broche 4 :** DTR (**Data Terminal Ready**) cette ligne est une sortie active à l'état haut. Elle permet à l'ordinateur de signaler au correspondant que le port série a été libéré et qu'il peut être utilisé s'il le souhaite.

**Broche 5 :** GND (**Ground**) c'est la masse.

**Broche 6 :** DSR (**Data Set Read**) cette ligne est une entrée active à l'état haut, elle permet au correspondant de signaler qu'une donnée est prête.

**Broche 7 :** RTS (**Request To Send**) cette ligne est une sortie, qui quand elle est active est à l'état haut, elle indique au correspondant que l'ordinateur veut lui transmettre des données.

**Broche 8 :** CTS (**Clear To Send**) cette ligne est une entrée active à l'état haut, elle indique à l'ordinateur que le correspondant est prêt à recevoir des données.

**Broche 9 :** RI (**Ring Indicator**) cette ligne est une entrée active à l'état haut, elle permet à l'ordinateur de savoir qu'un correspondant veut initier une communication avec lui.

En résumé, les signaux les plus utilisés en micro informatique (plus de 99 % des cas) sont FG et SG qui sont très souvent reliés entre eux, RD et TD puisque ce sont eux qui assurent la transmission proprement dite et ensuite, RTS, CTS, DCD, DTR et DSR, par ailleurs, la fonction de ces signaux est souvent simplifiée ; DTR par exemple est fréquemment couplée aux lignes d'alimentation de l'appareil et passe au niveau haut lorsque celui-ci est mis sous tension.

Sur de nombreux matériels, la simplification va encore plus loin puisque parfois ne restent que SG (la masse), RD et TD, cela suffit pour établir une liaison série asynchrone mais peut conduire à des problèmes si l'équipement auquel vous reliez un tel appareil exploite certains signaux de contrôle, le cas le plus classique est un blocage de l'ensemble du au fait, par exemple, que l'appareil connecté attend DTR ou CTS pour travailler.

### ■ Remarque

La communication entre systèmes informatiques s'effectue grâce à des liaisons dont les principaux éléments sont définis par des normes dans le cadre du **CCITT** (Comité Consultatif International des Téléphones et Télégraphes).

**ETTD** : Equipement Terminal de Traitement de Données (ou **DTE** : Data Terminal Equipment).

**ETCD** : Equipement de Terminaison de Circuit de Données (ou **DCE** : Data Communication Equipment).

L'ETTD est un élément qui ne se connecte pas directement à la ligne de transmission. Par exemple : Un ordinateur, un terminal, une imprimante...

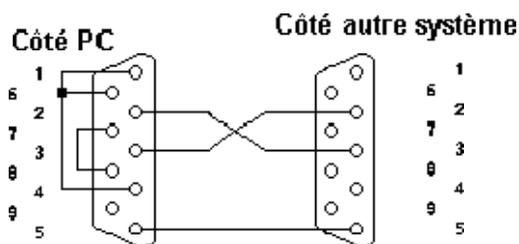
L'ETCD assure la transmission des données. Par exemple : Un modem, un multiplexeur... il a deux fonctions principales :

- L'adaptation du signal de l'ETTD à la ligne (codage et modulation en émission, démodulation et décodage en réception). [22]
- La gestion de la liaison (établissement, maintien et libération de la ligne), dans notre montage on envoie que des bits on utilise le RS 232 et on ne va pas faire une modulation/démodulation alors la liaison que nous choisis est défini ci-dessous :

### III.5.2. Le câblage des trois fils DTE/DTE

DTE : désignait, à l'origine, uniquement les ordinateurs et les terminaux. À présent, on y retrouve quasiment la totalité des équipements pourvus d'une liaison RS 232.

Ce type de câblage est sans doute le plus utilisé, car il répond aux milieux à un besoin de facilité d'utilisation ainsi que de mise en œuvre, c'est également ce type de liaison qui est le plus utilisé avec les microcontrôleurs.



**Figure III.5 :** Liaison à trois fils.

La liaison série bidirectionnelle la plus simple utilise 3 fils : Tx, Rx, Masse, c'est une connexion entre 2 équipements qui se conforme au schéma suivant :

Ce type de connexion 3 fils est peu fiable pour la raison suivante : à 9600 Baud, la transmission de 8 bits de données nécessite l'envoi de 10 bits minimum (8 bits + 1 bit de Start + 1 bit de stop), si l'un des équipements envoie des données en permanence, il le fait au rythme de  $9600 \text{ Baud} / 10 \text{ bits} = 960$  caractères par seconde, ainsi l'équipement de réception doit être capable de retirer 1 caractère toute les millisecondes et de le traiter.

Dans ces systèmes, les INTERRUPTIONS peuvent être masquées pendant plusieurs centaines de millisecondes interdisant la prise en compte de l'interruption UART signalant la présence d'un caractère dans le buffer de réception, les caractères reçus sont écrasés au fur et à mesure par les nouveaux arrivants, les spécifications du DOS imposent un temps de masquage des interruptions toujours inférieur à 1ms.

La liaison ne nécessite que 3 fils, et convient particulièrement aux microcontrôleurs qui implémentent rarement le contrôle de flux matériel par RTS/CTS (liaison 5 fils).

Dans notre système on n'a pas besoin de contrôler le flux car il s'agit d'envoyer une commande sous la forme des bites. [23]

### III.6. L'UART

UART signifie «**U**niversal **A**synchronous **R**eceiver **T**ransmitter », cela se traduit, en français, par émetteur récepteur asynchrone universel. Sous ce terme se cache un composant qui décharge le microprocesseur de la gestion du port série, ce composant dispose d'une interface parallèle pour dialoguer avec le microprocesseur et plus généralement de deux interfaces série **Figure III.6.** [23]

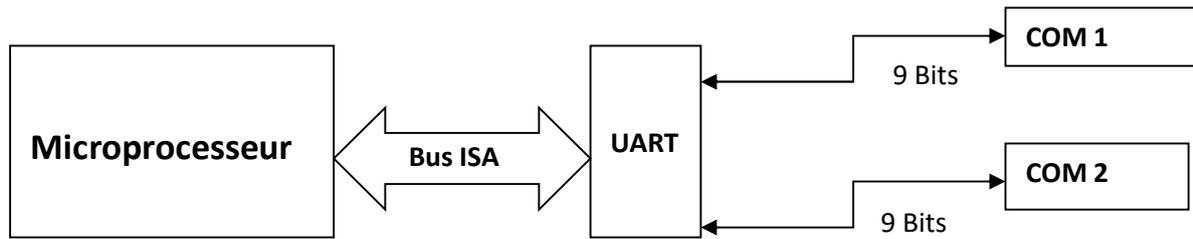


Figure III.6 : Synoptique d'une liaison série.

L'UART a pour fonction de sérialiser les données et de réguler le débit d'information sur la liaison RS 232, elle contient les registres permettant de configurer la liaison et de mettre à disposition les données en vue d'une transmission **Figure III.7.**

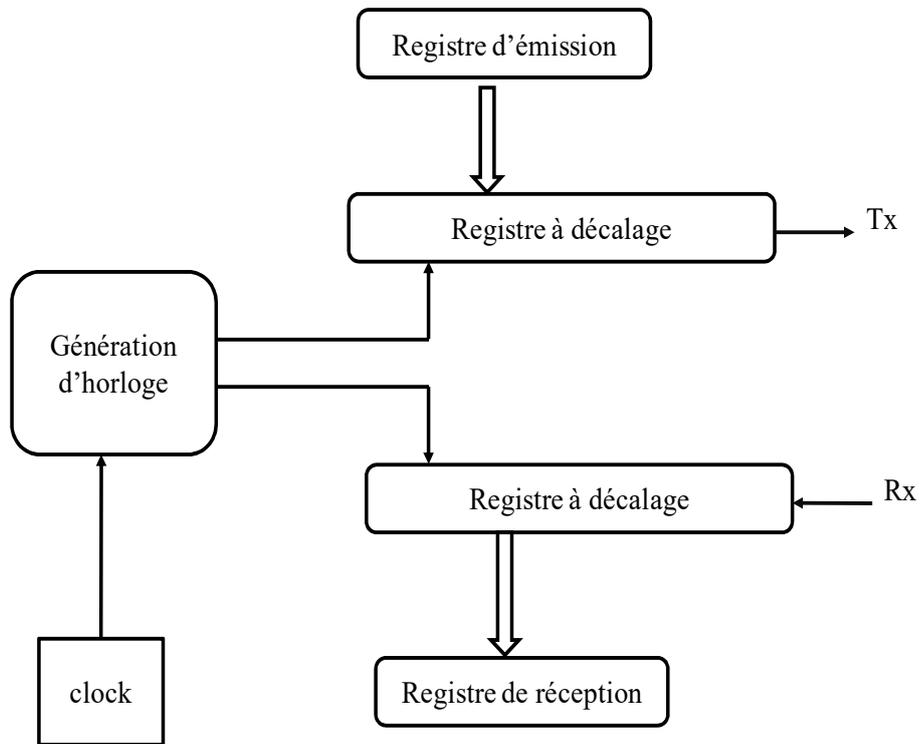


Figure III.7 : Schéma de fonctionnement de l'UART.

La référence en terme L'UART, c'est le 8250 qui étaient présent sur les premières cartes RS 232 pour PC. Depuis, les UART ont beaucoup évolué, le débit devient 115Kbauds pour le 16C550 contre 9600 bauds à l'origine, maintenant on ne trouve que très rarement des UART en tant que composant seul elles sont généralement intégré dans un ASIC remplissant également d'autre fonction comme la gestion...etc.

- **Adresse de base des quatre premier RS 232**

Les adresses affectés au port série sont donc au nombre de quatre d'ordre (COM1, COM2, COM3, COM4), on peut aller au de la avec une carte spécifique mais rien n'est normalisé dans le monde PC.

COM1: 3F8h (1016d)

COM2: 2F8h (760d)

COM3: 3E8h (1000d)

COM4: 2E8h (744d)

### **III.6.1. Le Principe de fonctionnement**

Le microprocesseur transmet à L'UART un message sous forme de 7 à 8 bits on parallèles, L'UART retransmet ce message en série au rythme d'horloge d'émission en l'encadrant de repaires de début (bit Start), de fin (1 ou 2 bits stop) et éventuellement d'un bit de contrôle de parité.

Si L'UART reçoit un message série, elle transforme le format série on format parallèle pour le présenter au microprocesseur, il effectue de plus différents contrôle (parité, format) de façon à signaler au microprocesseur de possibles erreurs de transmission. [18]

### **III.7. La carte a microcontrôleur**

Il s'agit d'une carte à base d'un microcontrôleur 68HC11A1, autour de ce dernier des connecteurs donnent accès aux ports d'entrées/sorties.

Elle est composée de plusieurs étages que nous définirons après, étage de l'alimentation, l'horloge, l'interface sérié, le reset et le brochage entre ses étages et tout ça est décrit dans le schéma développé donné par la **Figure III.8**.

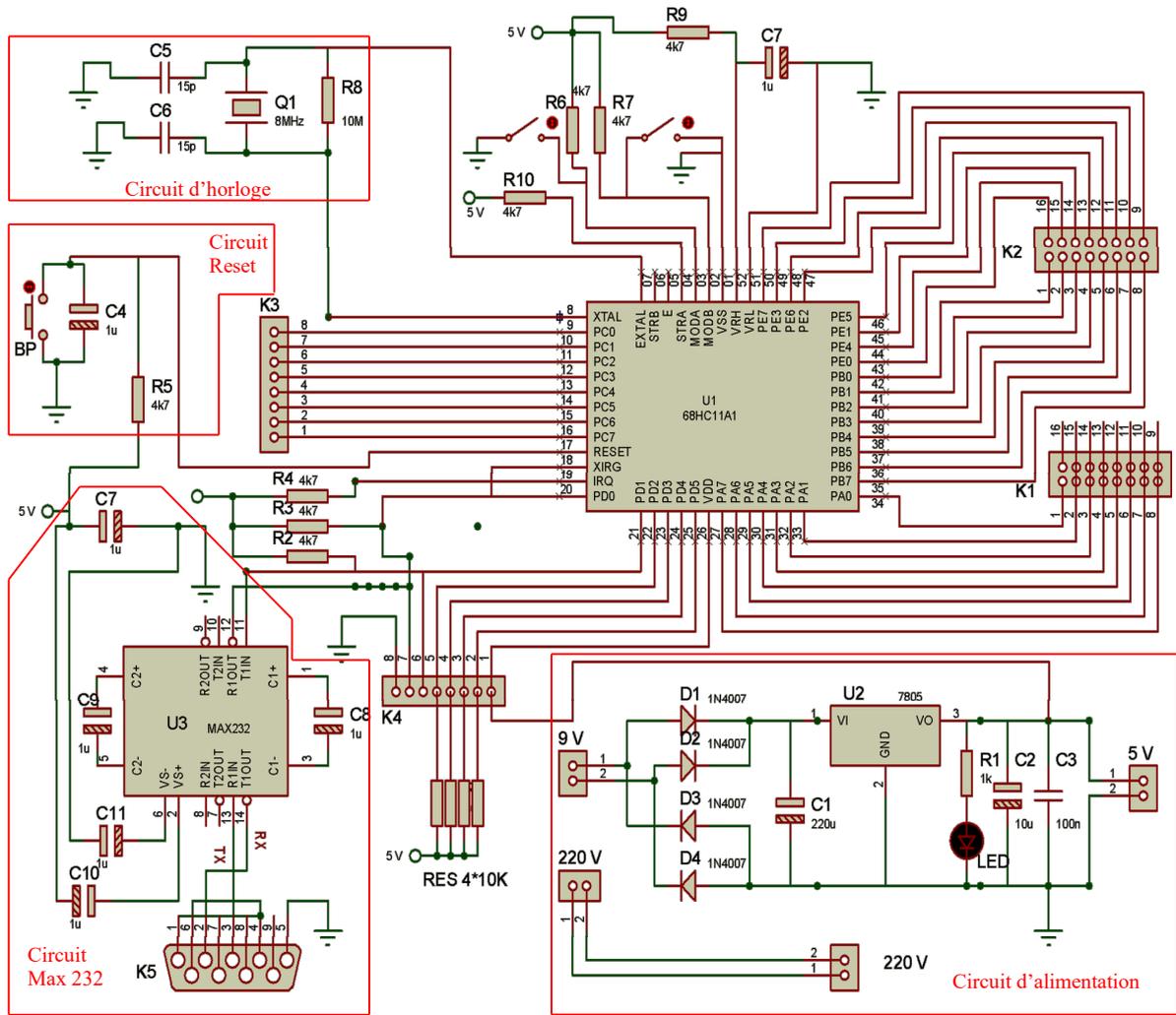


Figure III.8 : Schéma électrique de la carte de commande à base de 68HC11A1.

### III.7.1. Interface série

La communication entre le PC et la carte de commande est assurée par une liaison série RS 232, les deux premières lignes du port D, la broche PD0 qui est liée à RX (réception) et la broche PD1 qui est liée à TX (transmission), assurent le transfert des données via le MAX 232, ce dernier est un convertisseur entre une liaison série TTL (0-5 V) et une liaison série RS 232 (+12 -12 V) et ce avec une simple alimentation 5 V.

### III.7.2. Reset

A la mise sous tension, la fonction RESET est réalisée par un bouton poussoir qui permet la remise à zéro manuel, ce circuit d'initialisation est composé d'une résistance et un condensateur.

### III.7.3. Horloge

Pour produire un signal d'horloge, la fonction d'oscillation est assurée par un circuit a quartz 8 MHz.

La **Figure III.8** montre la connexion du quartz aux broches XTAL et EXTAL, la fréquence du quartz doit être égale a quatre fois la fréquence réelle de fonctionnement ce qui donne dans ce cas 2 MHz.

### III.7.4. Alimentation

Dans notre projet on a utilisé une alimentation stabilisée qui délivre une tension 5 V obtenu de la source 220 V 50 Hz ; autour d'un transformateur, pont redresseur, régulateur, les capacités de filtrage et une LED rouge permet de vérifier la présence de la tension sur notre circuit.

## II.8. Carte d'isolation optique

L'isolation optique a pour rôle d'isolé la partie basse tension (+5 V) et la tension de secteur 220 V.

### III.8.1. Principe de fonctionnement des opto-triacs

Un opto-triac est composé d'un triac et une LED intégrés, la mise en œuvre de ce dispositif est plus simple, puisqu'il suffit d'appliquer un niveau positif bas tension sur la LED de l'opto-triac, qui commande à son tour le triac, l'opto-triac fait donc figure d'interface, en quelque sorte, entre le circuit de commande et le circuit commandé.

### III.8.2. Choix d'opto-triac

Il existe différent type d'opto-triac dans le marché et parmi les plus connus :

- MOC3041.
- MOC3031.
- MOC3021.

Dans notre cas utilisé le MOC3041 qui correspond a nos besoins (courant et tension).

### III.8.3. Circuit de la carte d'isolation optique

Le montage qui suit est basé sur des opto-triacs de type MOC3041, une seule résistance est ajoutée entre l'opto-triac et le triac, qui permet de limiter le courant de gâchette du triac, la valeur de cette résistance doit être adaptée en fonction de la sensibilité du triac.

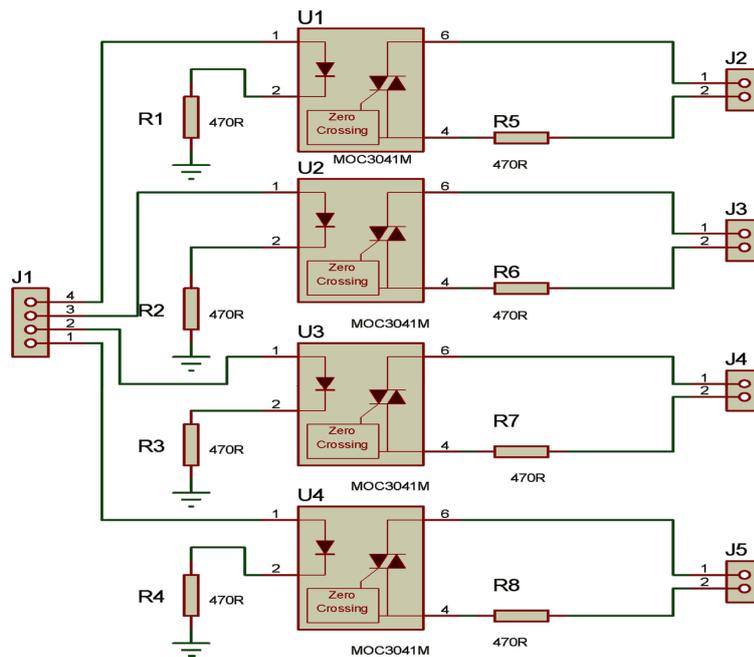


Figure III.9 : Le schéma développé de la carte d'isolation optique.

### III.9. Carte de puissance

Les 4 sorties de la carte de commande (PORTB) sont reliées à les 4 entrées de la carte de puissance, ces entrées seront transformer en 4 sorties de la carte de puissance en 220 V avec une protection totale de la carte de commande, une séparation est due grâce à une isolation galvanique, ce qui permet le déclenchement du triac qui a le rôle de soutenir des courants très élevés, la liaison entre la partie commande et la partie puissance se réalise sans connexion électrique, ce qui protège l'utilisateur mais aussi toute l'infrastructure de commande **Figure III.10**.

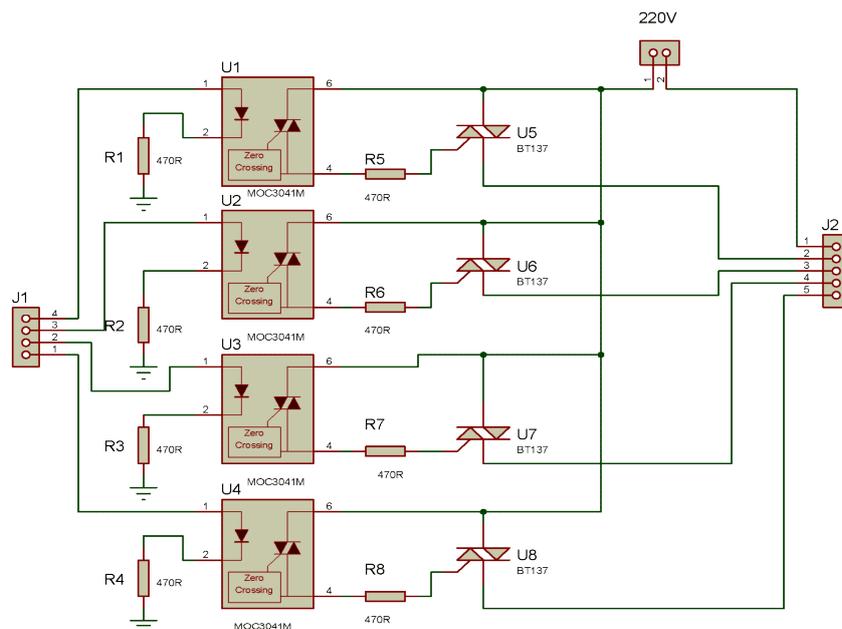


Figure III.10 : La carte de puissance.

### III.10. Conclusion

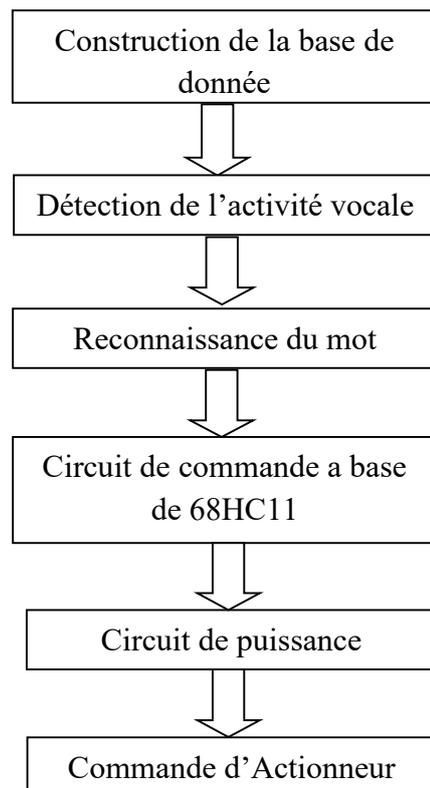
Ce chapitre a été consacré aux détails du protocole de transmission série des données utilisé, le suivant est dédié à l'exploitation des résultats expérimentaux obtenus suite aux tests.

## IV.1. Introduction

Dans ce chapitre, nous allons surtout détailler les moyens utilisés pour la réalisation de notre application, sachant que notre objectif étant de réaliser une commande automatique d'actionnaires par un signal vocal on parlera en premier lieu de l'implémentation de l'algorithme VAD pour la détection d'activité vocale ensuite celle de l'algorithme DTW pour la reconnaissance du mot isolé, ceci sera suivi par le programme du protocole série en MATLAB à la fin en parlera se la conception de notre interface.

## IV.2. Description fonctionnelle de notre application

Notre application suivra les étapes qui seront définies dans la synoptique qui suit, on commencera par le bloc la construction de notre base de données et on continuera jusqu'à ce qu'on abouti au bloc de commande d'actionneur.



**Figure IV.1 :** Schéma illustrant les étages fonctionnels du projet.

### IV.3. Programmation et mise en marche

Pour la mise en œuvre de la description citée au dessus, on suivra les étapes suivantes en utilisant le BASIC11 pour la programmation du microcontrôleur et le MATLAB pour la construction de la base de données et l'implémentation des deux algorithmes cité auparavant.

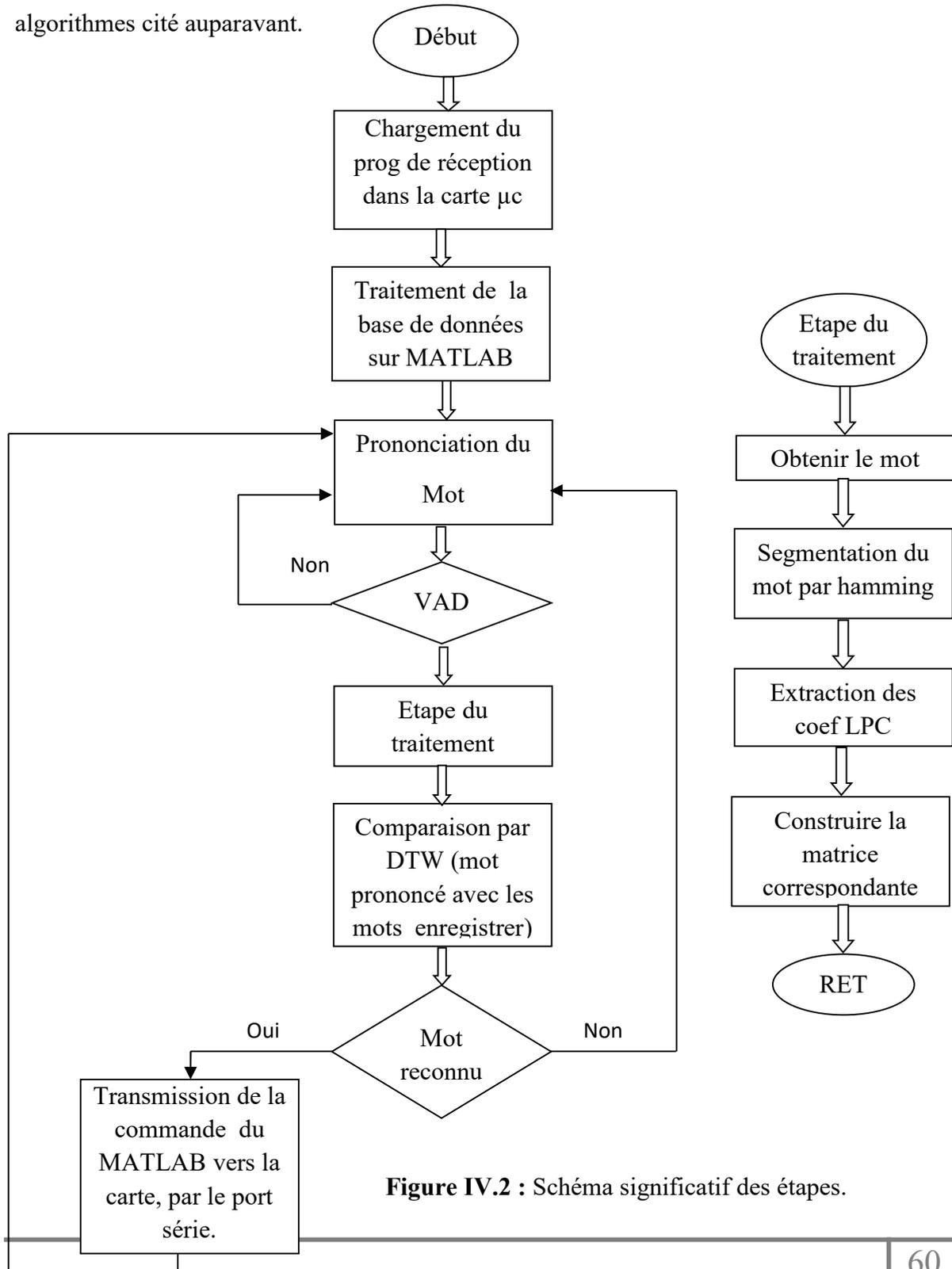


Figure IV.2 : Schéma significatif des étapes.

### IV.3.1. Le logiciel BASIC11

La surface de programmation permet d'écrire, de compiler, de charger et de déboguer notre programme en BASIC11, le programme s'affichera après le lancement deux fenêtres, la fenêtre principale nous permet de lancer un éditeur pour éditer le programme source et compiler le programme, la fenêtre de débogueur nous permet de communiquer avec la carte commande à base de 68HC11. Dans le menu FICHIER, nous trouverons les outils pour ouvrir et traiter le programme source principal. COMPILER lance le compilateur, le curseur d'attente s'affiche pendant la compilation, après on trouve la sortie des programmes de compilation. et en fin, le chargement du programme s'effectuera des qu'on sélectionne le mode de fonctionnement de la carte de commande et juste après avoir appuyer sur le bouton du Reset de cette dernière.

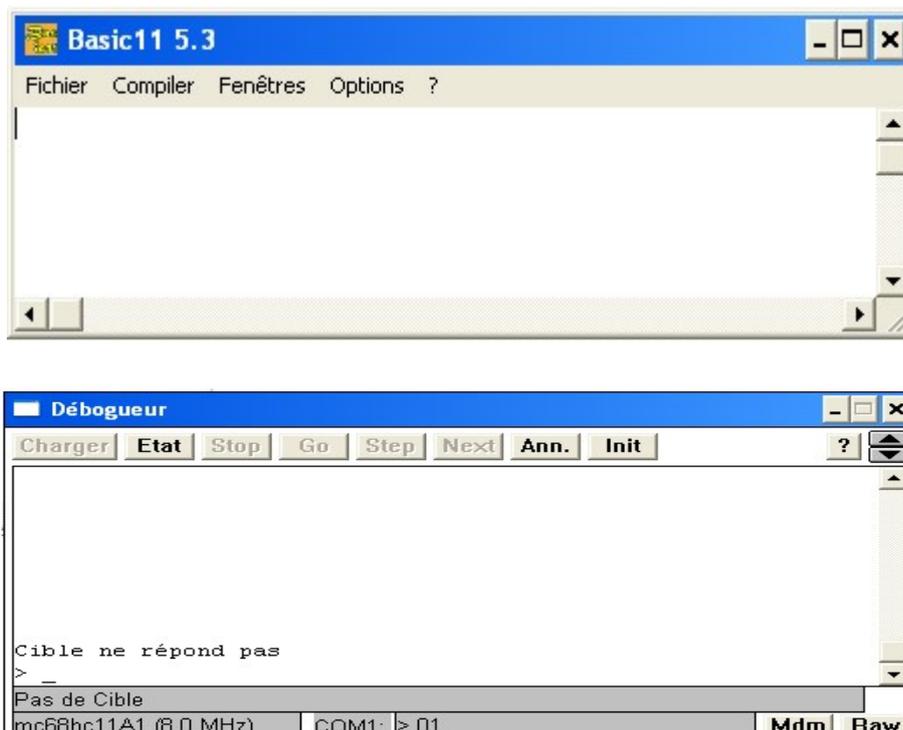


Figure IV.3 : Les fenêtres de logiciel BASIC11.

- **Conditions d'application du BASIC11**

Afin de vérifier les conditions d'application du BASIC11 sur notre carte de commande, on s'est proposé d'exécuter les étapes suivantes :

- ✓ Vérification de l'alimentation (une LED indique l'état).
  - ✓ Vérification du système d'exploitation exécuté sur le PC Windows 98 ou XP.
  - ✓ Vérification le fil de liaison entre la carte cible et l'ordinateur.
  - ✓ Vérification de la configuration (FICHIER/CONFIGURATION), il suffit de choisir le port du PC et le matériel dans la deuxième ligne.
- **Le programme de BASIC11**
    - L'organigramme :

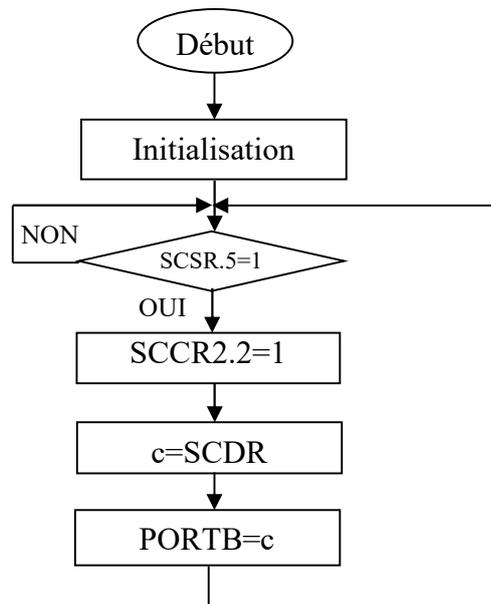


Figure IV.4.a : L'organigramme de réception de données.

```
trans.bas
Fichier  Edition  Chercher
do
    do
    loop until SCSR.5=1
B618  SCCR2.2=1
B61E   c=SCDR
B622  PORTB=c      'l'envoi de ces signaux sur le
B624 loop
```

**Figure IV.4.b** : Le programme de réception de données sur BASIC11.

- **Le compilateur**



**Figure IV.5** : Le compilateur de BASIC11.

- **Chargement de programme**

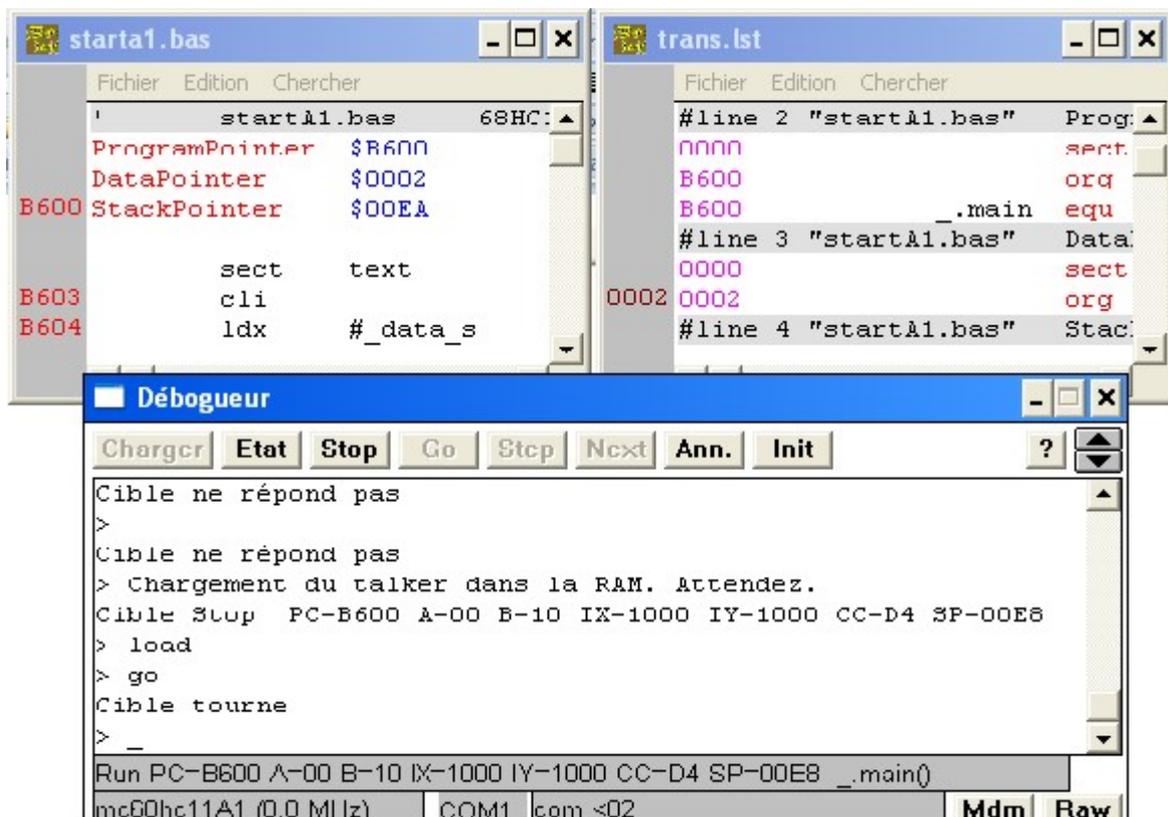


Figure IV.6 : Chargement de programme de BASIC11.

### IV.3.2. Exécution de programme MATLAB

Dans ce qui va suivre on va détailler chaque programme des trois blocs de la Figure IV.1, et le programme de la transmission de notre commande c'est-à-dire le dernier bloc de la Figure IV.2.

- **Construction de la base de données**

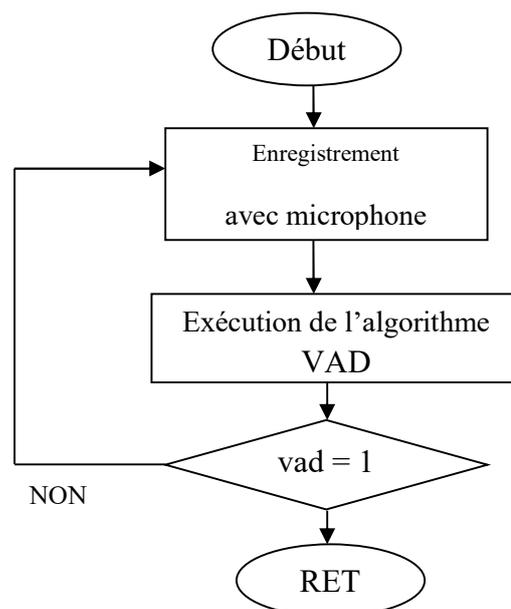
Tout d'abord on a choisis les mots (gauche, droite, stop), pour notre application et on a enregistré ces mots dans un dossier 'référence' (la bibliothèque) à l'aide d'un microphone, chaque mot enregistré 10 fois pour augmenter la probabilité de reconnaissance de mot. il paraît évident que si on augmente ce nombre on gagnera en précision mais comme dire je c'est comme une épée à double tranchant c'est-à-dire si d'un côté on gagne de la précision de l'autre côté on augmente le temps de traitement et ce tout ingénieur essaie de minimiser.

Pour remplir chaque matrice d'un mot il faut obtenir le signal afin d'appliquer la fenêtre de Hamming et extraire les coefficients de LPC.

- **La détection de l'activité vocale**

Comme il est montré dans la figure ci-dessous ce bloc commence par l'enregistrement qui se fait de façon continue via un microphone en utilisant un programme de MATLAB qui fait appel à un magnétophone virtuelle juste après l'algorithme VAD s'exécute et nous donnera en sortie une valeur ou un 0 ou bien un 1, si le résultat est un 1 cela voudrait dire qu'on a enregistré de la parole et dans ce cas on passera à la suite du programme principale dans le cas contraire ça voudrait dire qu'il y a eu un silence et qu'on doit faire un autre enregistrement.

RET : détection d'une activité vocale



**Figure IV.6 :** Organigramme du bloc de la détection d'activité vocale.

Voici un exemple de détection

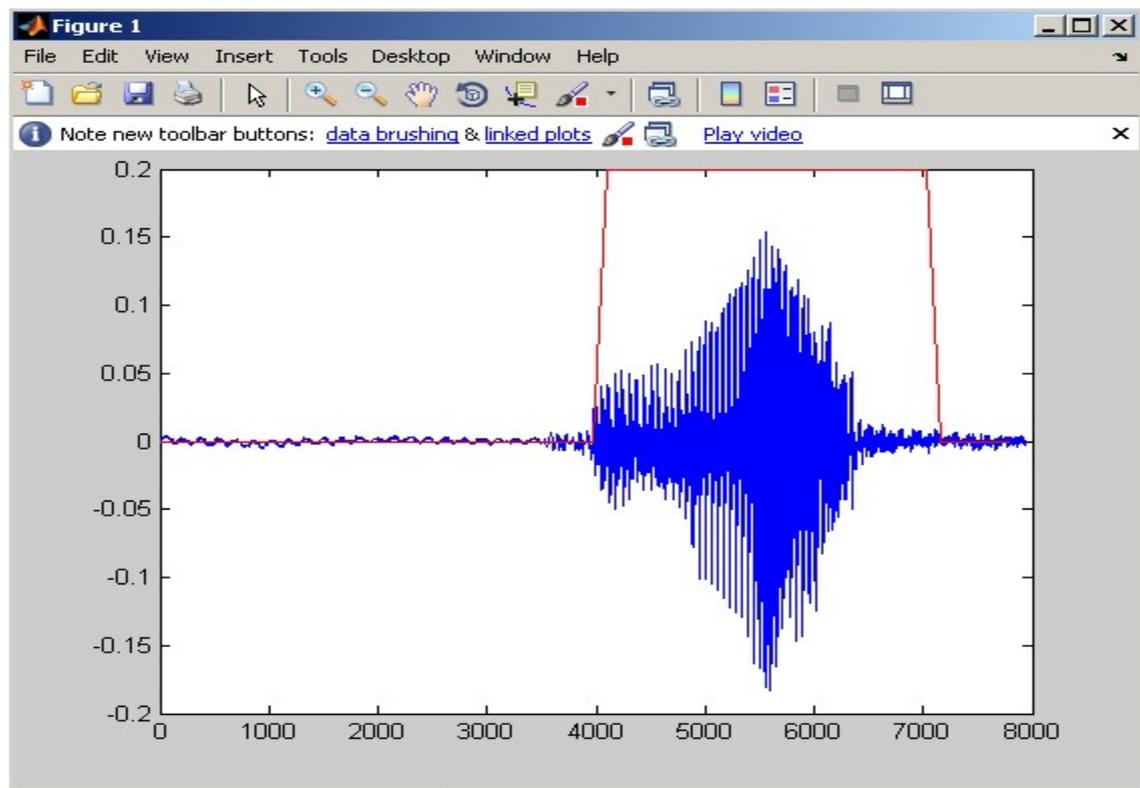
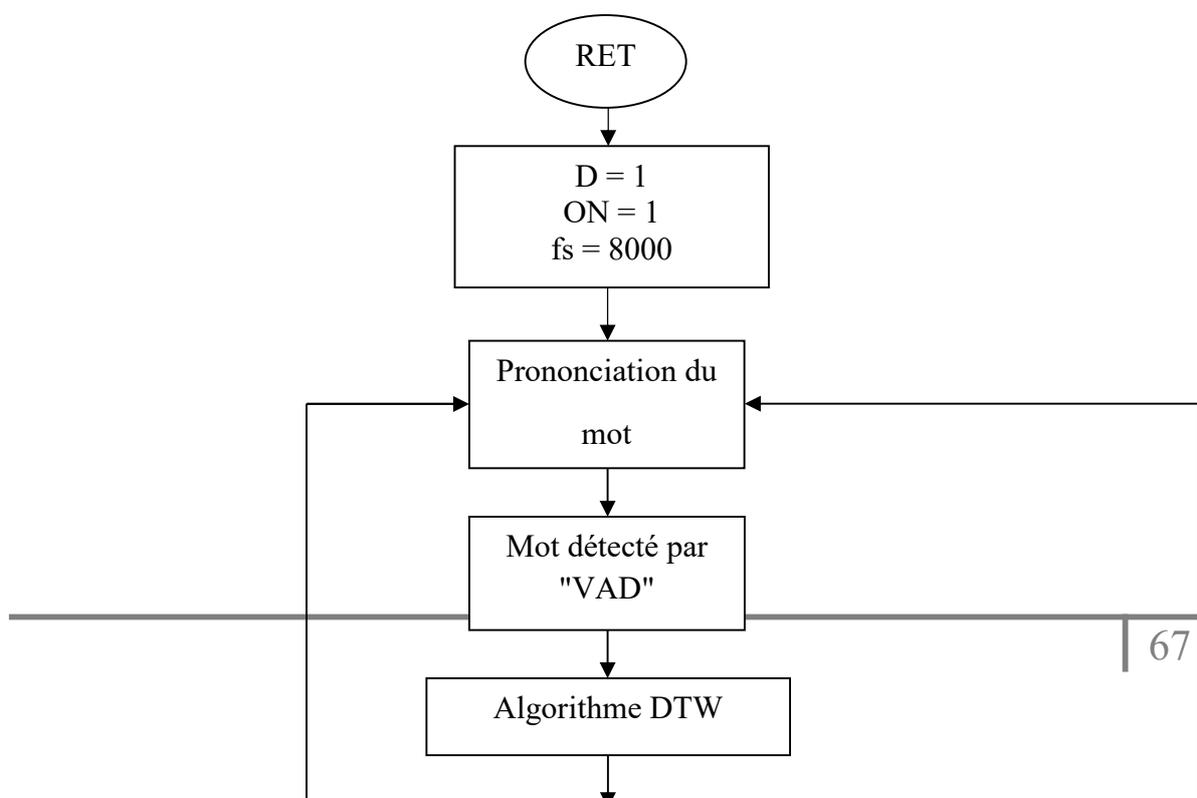


Figure IV.7 : Détection d'une activité vocale.

- **Reconnaissance du mot**

Lorsqu'on exécute notre programme, on aura une indication du moment où l'on doit prononcer le mot testé pour qu'il soit pris en considération, après la prononciation du mot on obtiendra sa matrice correspondante en utilisant la même étape du traitement, l'algorithme de la DTW nous permet de faire la comparaison de la matrice du mot testé avec les matrices des mots de références en nous donnant une matrice de dissemblance du mot testé avec tout les mots de notre base de données cette dissemblance est représentée par une distance, alors la distance la plus petite sera celle du mot reconnue, le problème c'est que quoique en prononce même si ce n'est qu'un sifflement l'algorithme DTW calcule cette dissemblance et nous donne une reconnaissance du mot qui a une distance de dissemblance la plus petit alors on doit fixer une distance minimale acceptée par notre système de reconnaissance cette distance est prise expérimentalement égale à 2,6.

Voici un organigramme qui résume ce qu'on vient de dire



RET : détection d'une activité vocale

D : durée d'enregistrement.

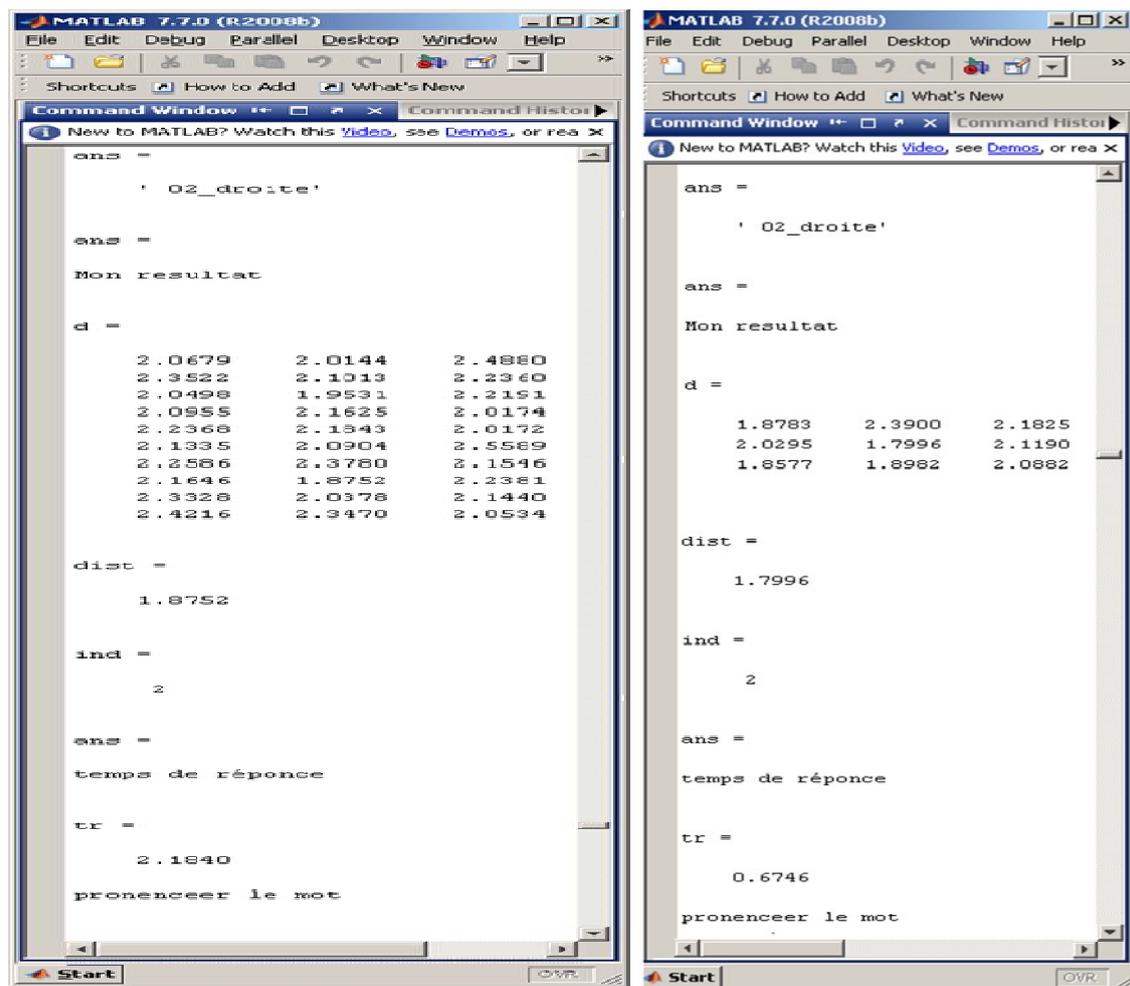
ON : système en marche.

fs : fréquence d'échantillonnage.

**Figure VI.8** : Organigramme de la reconnaissance et d'envoi de la commande.

On a procédé à la reconnaissance du mot « droite » avec deux bases de données différentes de taille.

Le MATLAB affiche le résultat du mot reconnue, la matrice des distances, la distance minimale, ainsi que le temps de réponse de notre système.



**Figure IV.9 :** Résultats du MATLAB pour la reconnaissance du mot « droite ».

Comme on le voit dans la figure ci-dessus le temps de réponse d'un système qui a une base de donnée constitué des mots : gauche, droite et stop répété trois fois chacun est vraiment minime par rapport à celui constitué de dix répétitions au lieu de trois et ça c'est tout à fait logique car en diminuant la taille de la base de donnée on diminue le temps de calcul mais coté précision de la reconnaissance qui est représenté par la distance minimale trouvée (plus la valeur de « dist » est petite plus la reconnaissance est bonne) ce n'est qu'un hasard qu'on a trouvé une meilleure précision avec le système constitué de la base de donnée de trois répétitions car si on augmente la taille de notre base de donnée on augmentera notre chance de coïncider le mot prononcé avec un mot de notre base de donnée et pour appuyer ce qu'on vient de dire, on a refait l'expérience avec le mot stop

pour une base de donnée constitué de trois répétitions alors on a trouvé beaucoup moins bonne qu'avec la base de donnée de dix répétition.



```
! 03_stop*

ans =

Mon resultat

d =

    3.1545    2.9850    2.7350
    3.2299    3.0115    2.8300
    3.0134    2.8300    2.8099

dist =

    2.7350

ind =

    3

ans =

temps de réponce

tr =

    0.6612
```

Figure IV.10 : Résultats du MATLAB pour la reconnaissance du mot « stop ».

#### IV.4. Conclusion

Dans ce chapitre on vient de voir les résultats de notre travail concernant la reconnaissance du mot isolé ainsi que le temps de réponse de notre système après quelques essais on peut dire que notre système donne de bons résultats de reconnaissance si il a une très grande base de référence mais en contre partie sa réponse sera lente.

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

وزارة التعليم العالي والبحث العلمي  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE  
جامعة سعد دحلب البليدة  
UNIVERSITE SAAD DAHLEB DE BLIDA



كلية الهندسة - دائرة الإلكترونيك  
FACULTE DES SCIENCES DE L'INGENIEUR  
DEPARTEMENT D'ELECTRONIQUE

# Commande vocale d'un actionneur

Proposé par:

Pro Mr: BENSELAMA.Z/A

Présenté par :

HOUARI RAOUF

LARABI BILLEL

## *Plan de travail*

- **Introduction**
  - **Généralité sur la parole**
  - **Les algorithmes VAD et DTW**
  - **Carte d'interface et mode de transmission**
  - **Application et résultats**
  - **Conclusion**
- 

A vintage silver microphone with a ribbed grille and a metal body, positioned on the right side of the frame. The background is a soft, light grey gradient. The word "Introduction" is written in a blue, cursive font across the center, with a faint reflection below it.

# *Introduction*

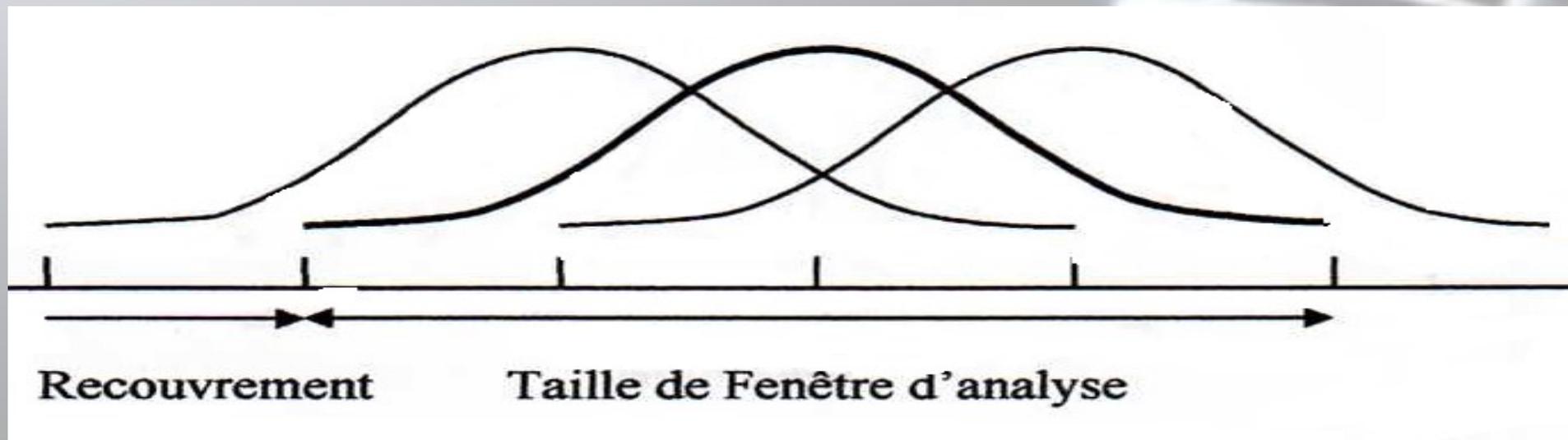


Qu'es ce que  
c'est que la  
parole ?

# Généralités sur la parole

## Paramètre descriptif de la parole

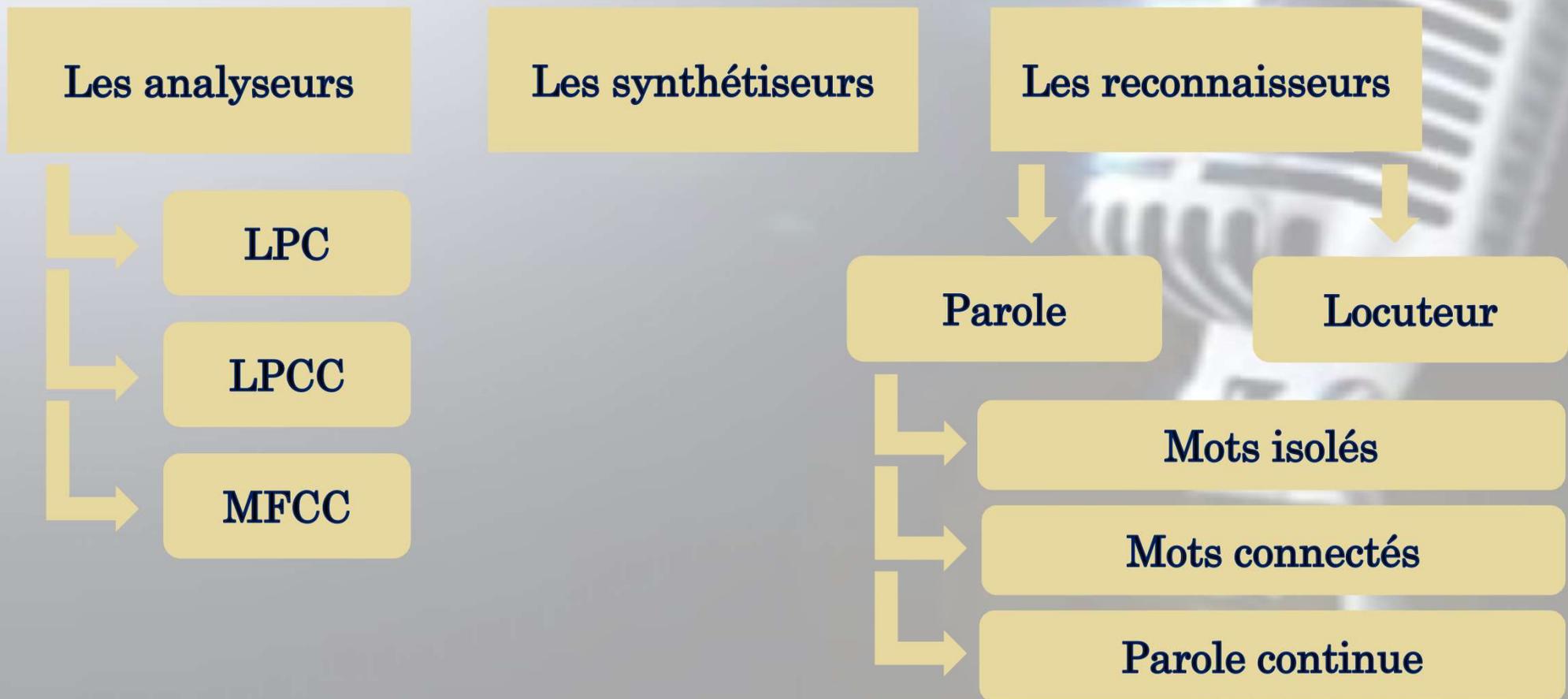
- ▣ Stationnarité du signal vocal
- ▣ Fenêtrage



$$W(n) = 0,54 - 0,46 \cdot \cos\left(\frac{2\pi n}{N-1}\right)$$

# Généralités sur la parole

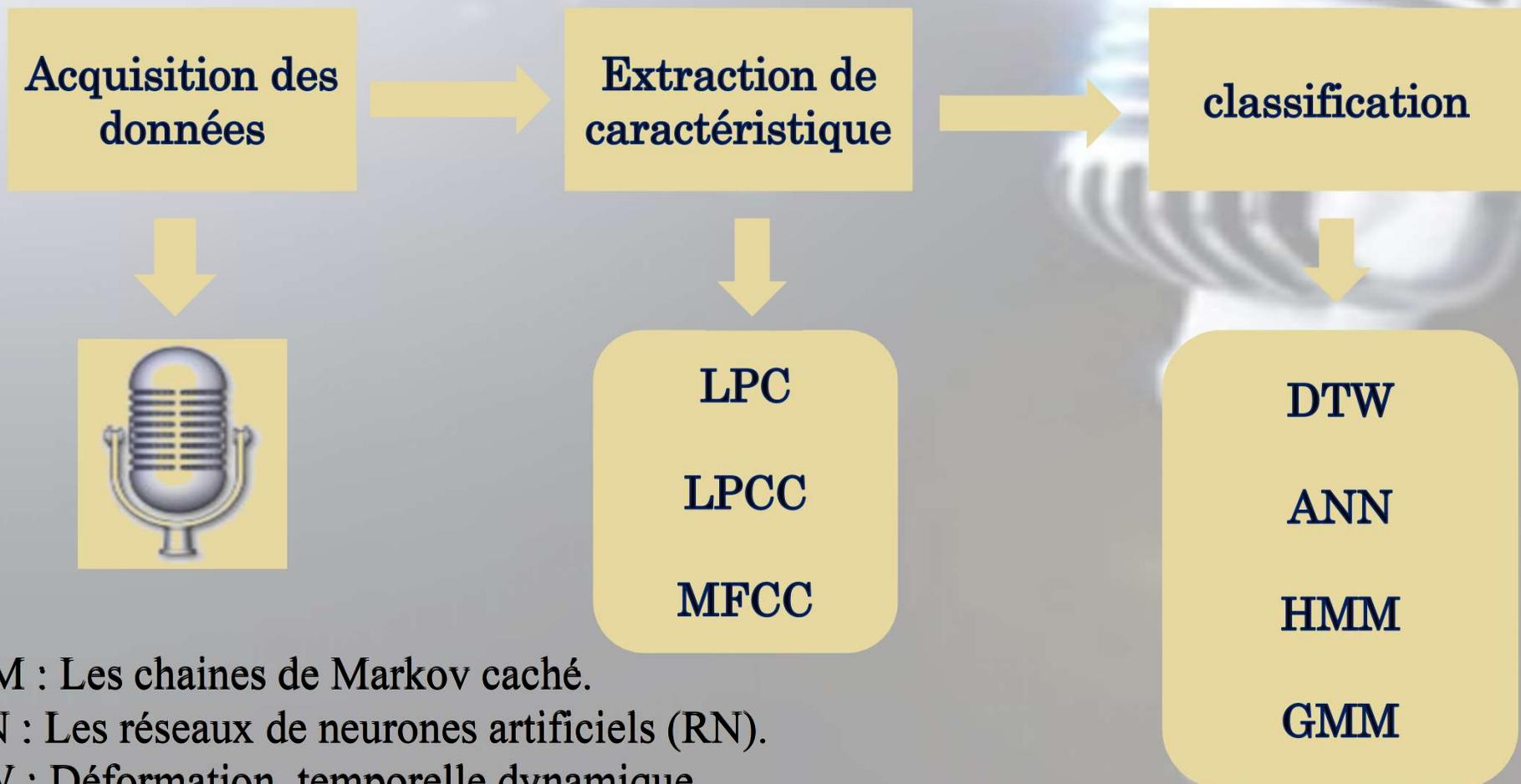
## Les différents traitements du signal de la parole



LPC (Linear Predictive Coding) ; LPCC(Linear Prédiction Cepstral Coefficients);MFCC(MEL-Frequency Cepstral Coefficients)

# Généralités sur la parole

## La chaîne de reconnaissance



HMM : Les chaînes de Markov caché.

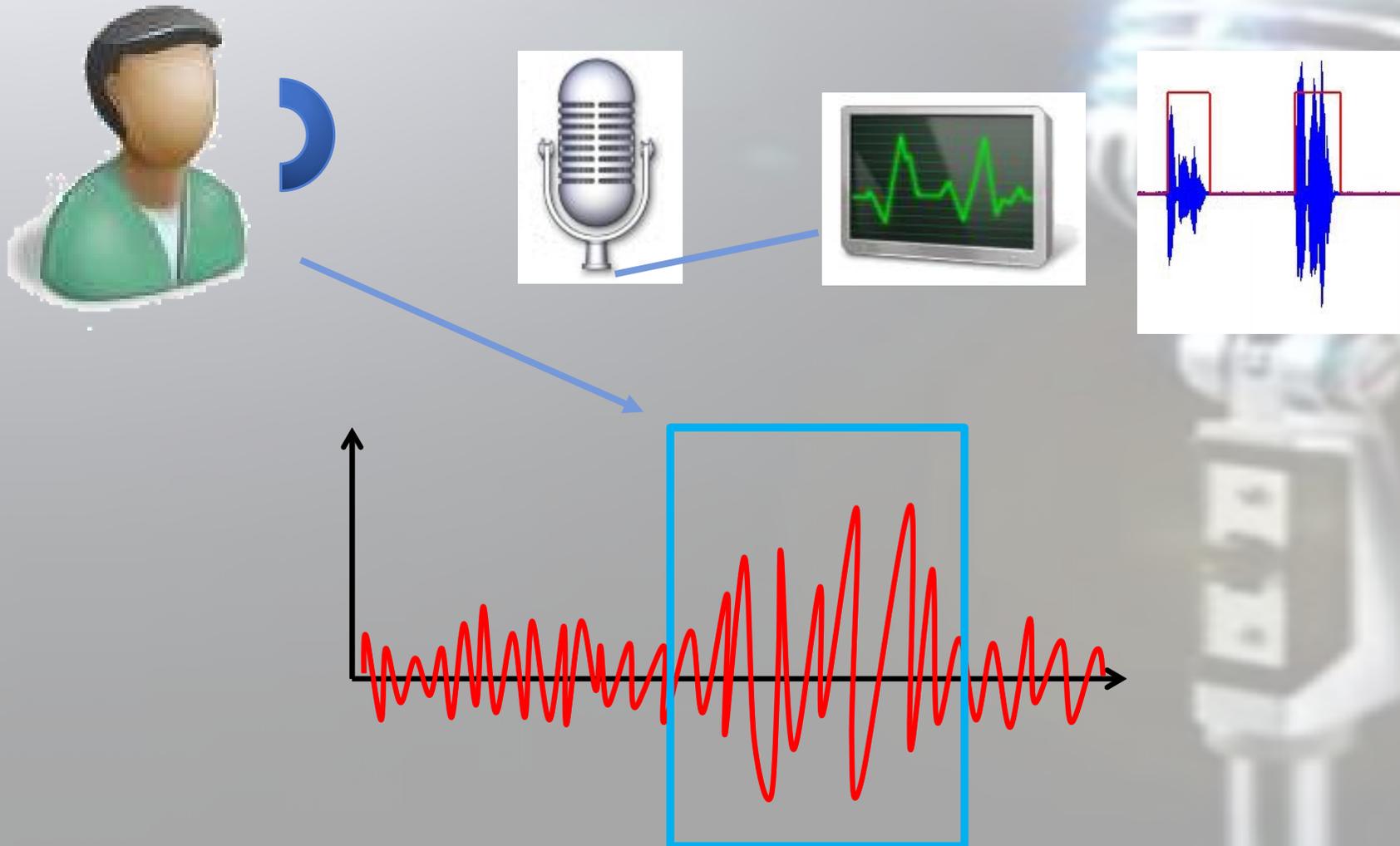
ANN : Les réseaux de neurones artificiels (RN).

DTW : Déformation temporelle dynamique.

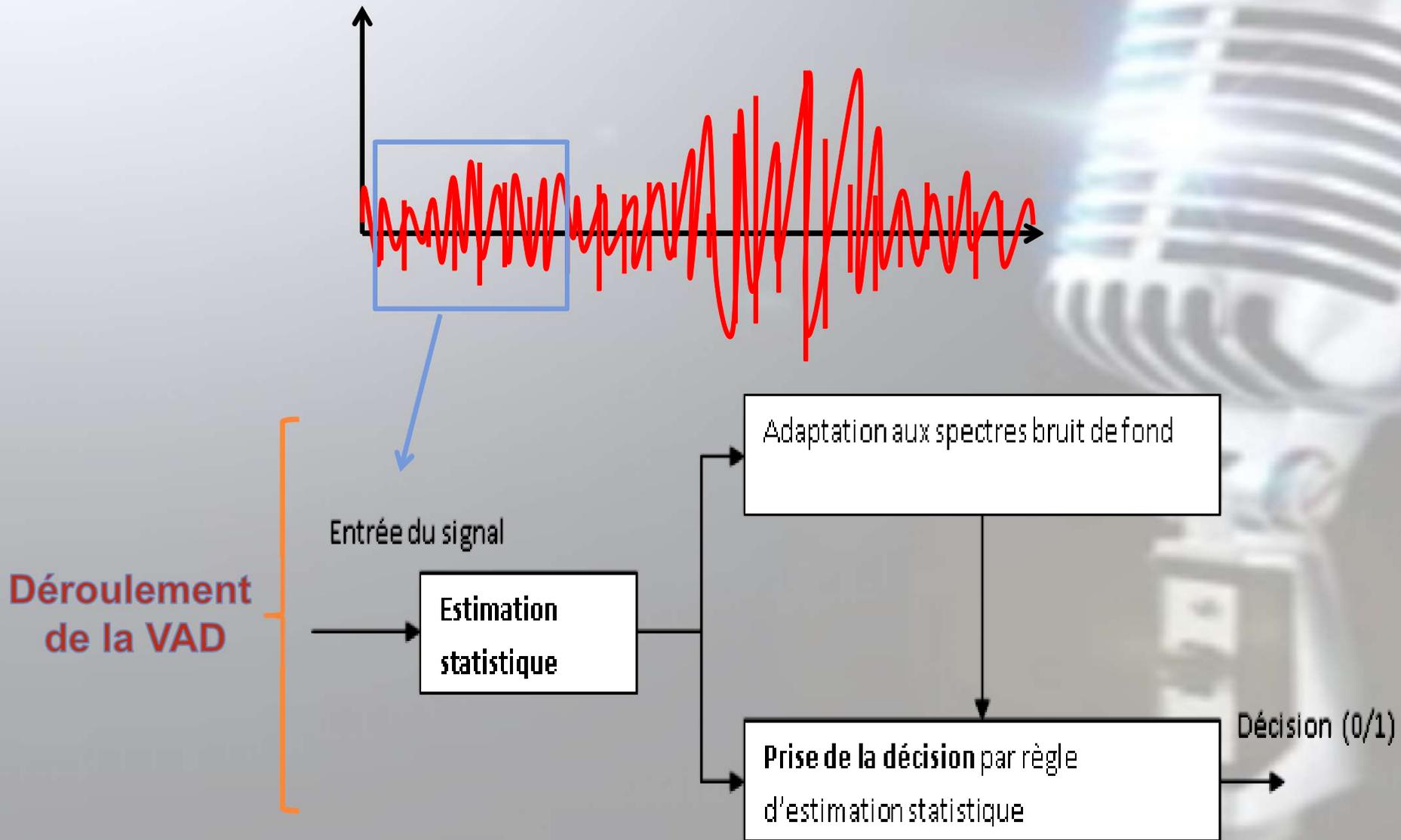
GMM : Mixture de multi gaussienne

# Les algorithmes VAD et DTW

## La Détection d'Activité Vocale VAD



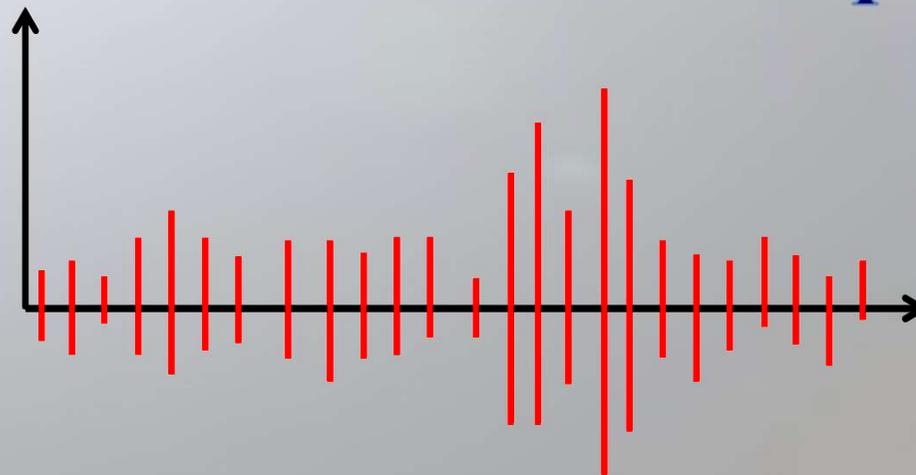
## La Détection d'Activité Vocale VAD



ADAPTATION AUX SPECTRES DU BRUIT

## La Détection d'Activité Vocale VAD

### Estimation Statistique



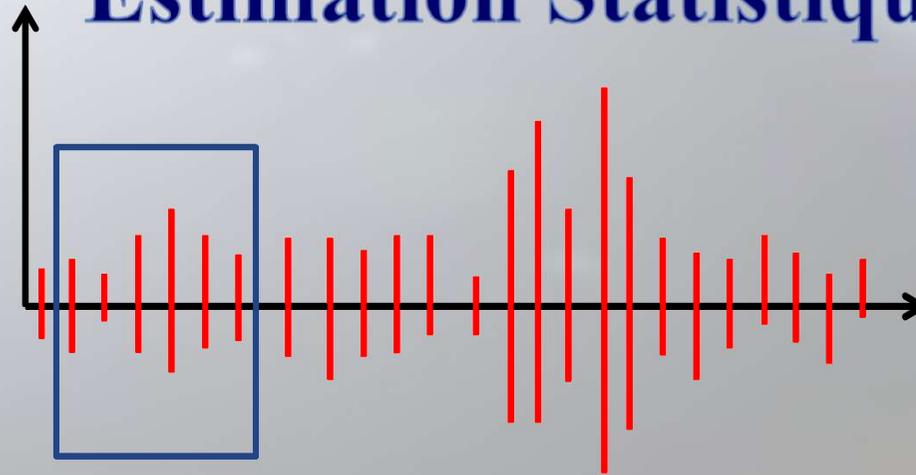
$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x}{\sigma}\right)^2}$$

- $\sigma$  : L'écart type
- $x$  : la valeur de l'échantillon
- La moyenne est nulle

# Les algorithmes VAD et DTW

## La Détection d'Activité Vocale VAD

### Estimation Statistique



Hypothèse 0 ( $H_0$ ) :  $X(t) \equiv N(t)$

Hypothèse 1 ( $H_1$ ) :  $X(t) \equiv N(t) + S(t)$

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)}$$

$P(A|B)$  : Loi a posteriori

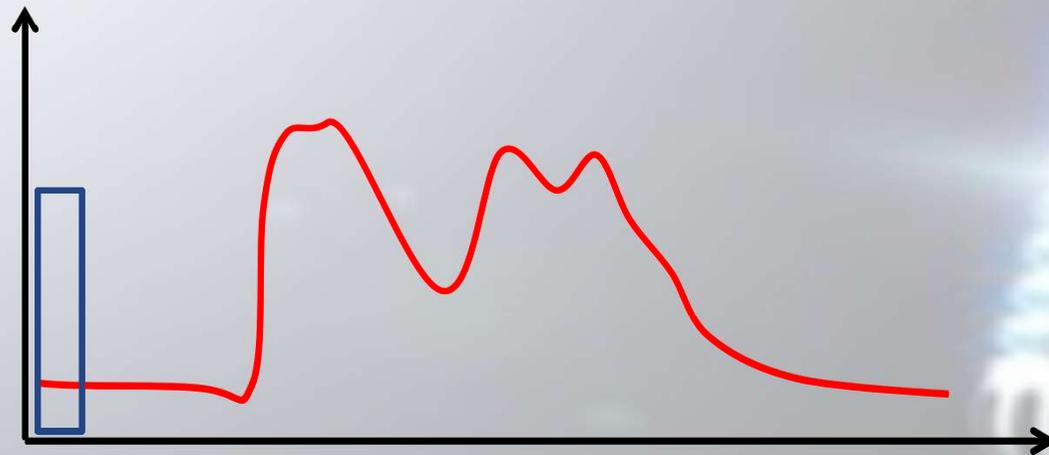
$P(B|A)$  : Vraisemblance

$P(A)$  : Loi a priori

$P(B)$  : Evidence (constante de normalisation)

# Les algorithmes VAD et DTW

## Initialisation et Prédiction du bruit



## Spectre énergétique

$$\hat{\lambda}_N(k) = E(\lambda_N(k) | H_0) \cdot P(H_0 | X_k) + E(\lambda_N(k) | H_1) \cdot P(H_1 | X_k)$$

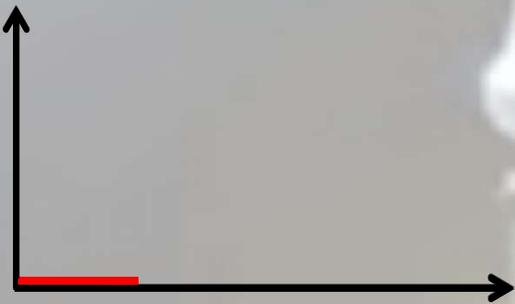
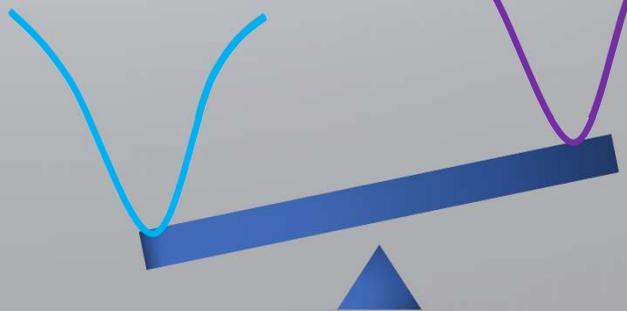
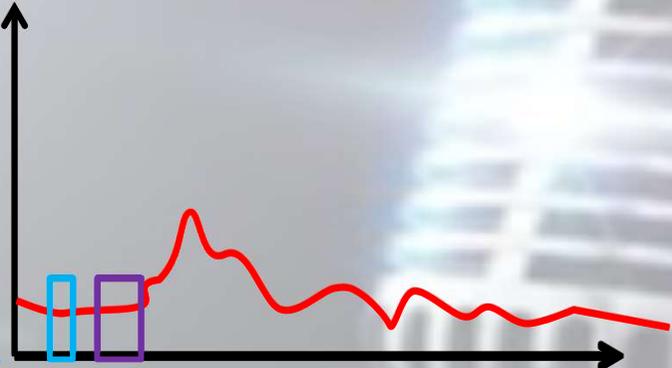
$$\Lambda(k) = \frac{P(X_k | H_1)}{P(X_k | H_0)}$$

$$\varepsilon = \frac{P(H_1)}{P(H_0)}$$

# Les algorithmes VAD et DTW

## Règle de décision

$$\Lambda_g(\mathbf{r}) = \frac{\max_{\theta} P(X | \hat{\Theta}, H_1)}{\max_{\theta} P(X | H_0)} > \gamma \quad H_1$$
$$\Lambda_g(\mathbf{r}) = \frac{\max_{\theta} P(X | \hat{\Theta}, H_1)}{\max_{\theta} P(X | H_0)} < \gamma \quad H_0$$



Mise a jour  
du bruit de fond

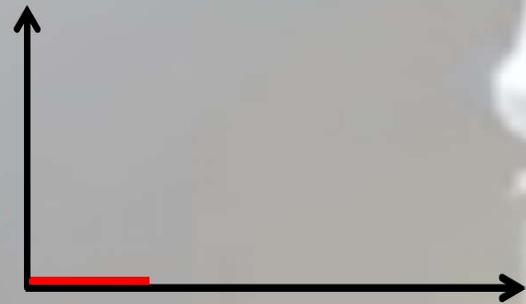
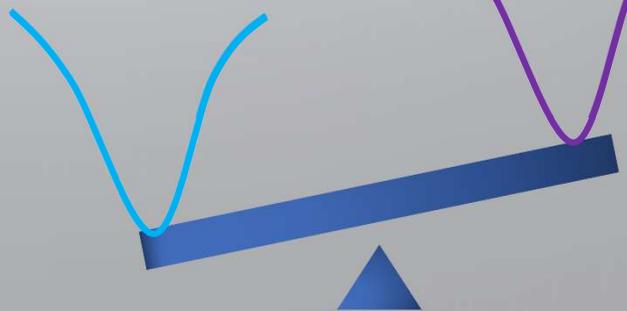
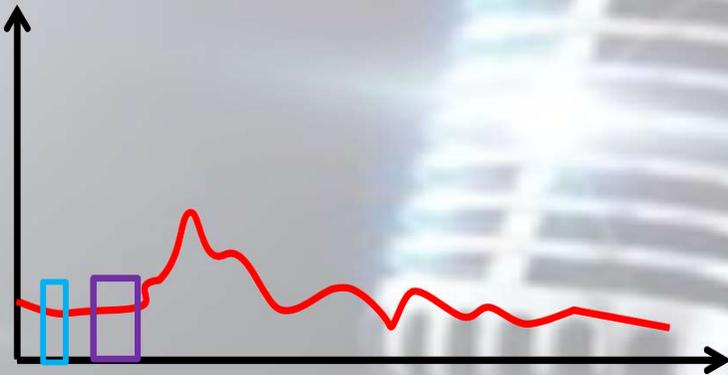
Déclanchement  
De la VAD

# Les algorithmes VAD et DTW

## Règle de décision

$$\Lambda_g(\mathbf{r}) = \frac{\max_{\theta} P(X | \hat{\Theta}, H_1)}{\max_{\theta} P(X | H_0)} > \gamma \quad H_1$$

$$\Lambda_g(\mathbf{r}) = \frac{\max_{\theta} P(X | \hat{\Theta}, H_1)}{\max_{\theta} P(X | H_0)} < \gamma \quad H_0$$



$$\hat{\lambda}_N^{(m)}(k) = \frac{1}{1 + \varepsilon \cdot \Lambda_g^{(m)}} \cdot |X_k^{(m)}|^2 + \frac{\varepsilon \cdot \Lambda_g^{(m)}}{1 + \varepsilon \cdot \Lambda_g^{(m)}} \cdot \hat{\lambda}_N^{(m-1)}(k)$$

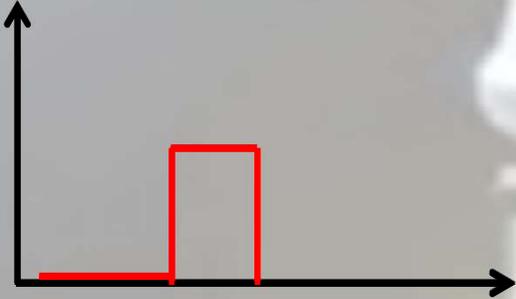
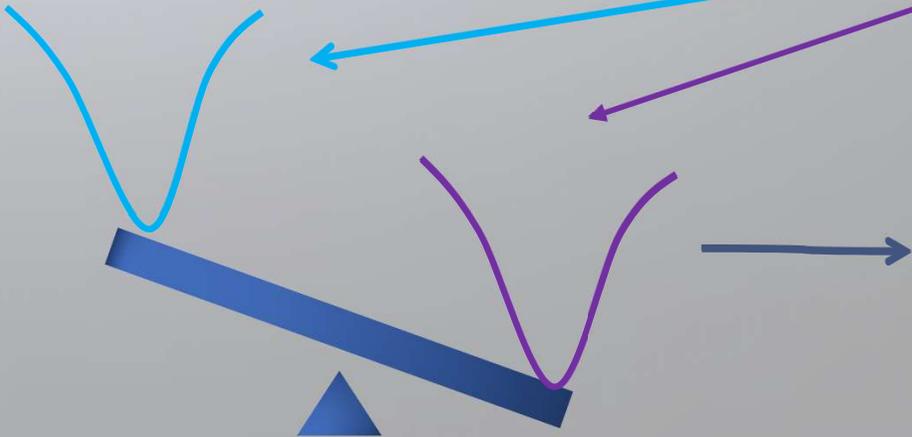
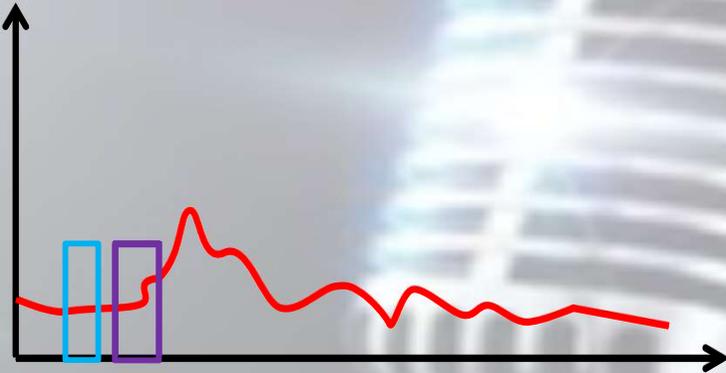
Mise a jour  
du bruit de fond

Déclanchement  
De la VAD

# Les algorithmes VAD et DTW

## Règle de décision

$$\Lambda_g(\mathbf{r}) = \frac{\max_{\theta} P(X | \hat{\Theta}, H_1)}{\max_{\theta} P(X | H_0)} > \gamma \quad H_1$$
$$\Lambda_g(\mathbf{r}) = \frac{\max_{\theta} P(X | \hat{\Theta}, H_1)}{\max_{\theta} P(X | H_0)} < \gamma \quad H_0$$



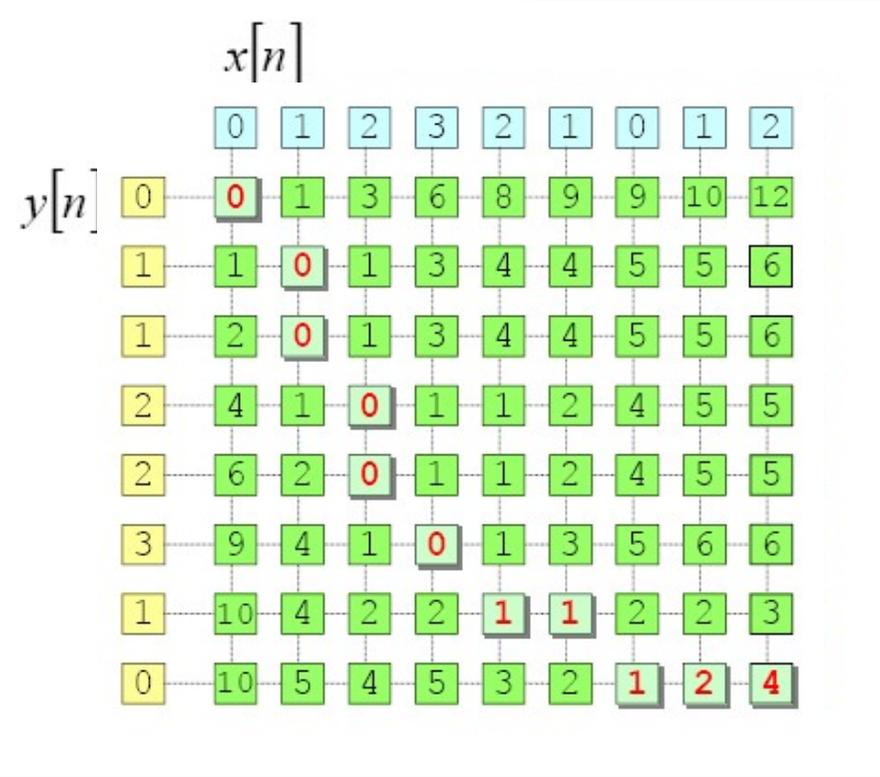
Mise a jour  
du bruit de fond

Déclanchement  
De la VAD

# Les algorithmes VAD et DTW

## La DTW

### Principe:

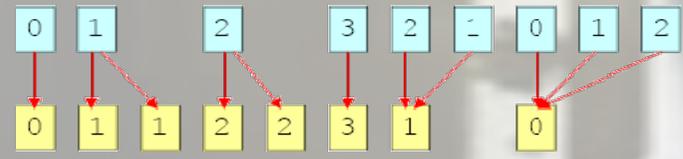


Commencer par calculé les distances locales:

$$d(A(:, i), B(:, j)) = \sqrt{\sum_{l=1}^p (A(l, i) - B(l, j))^2}$$

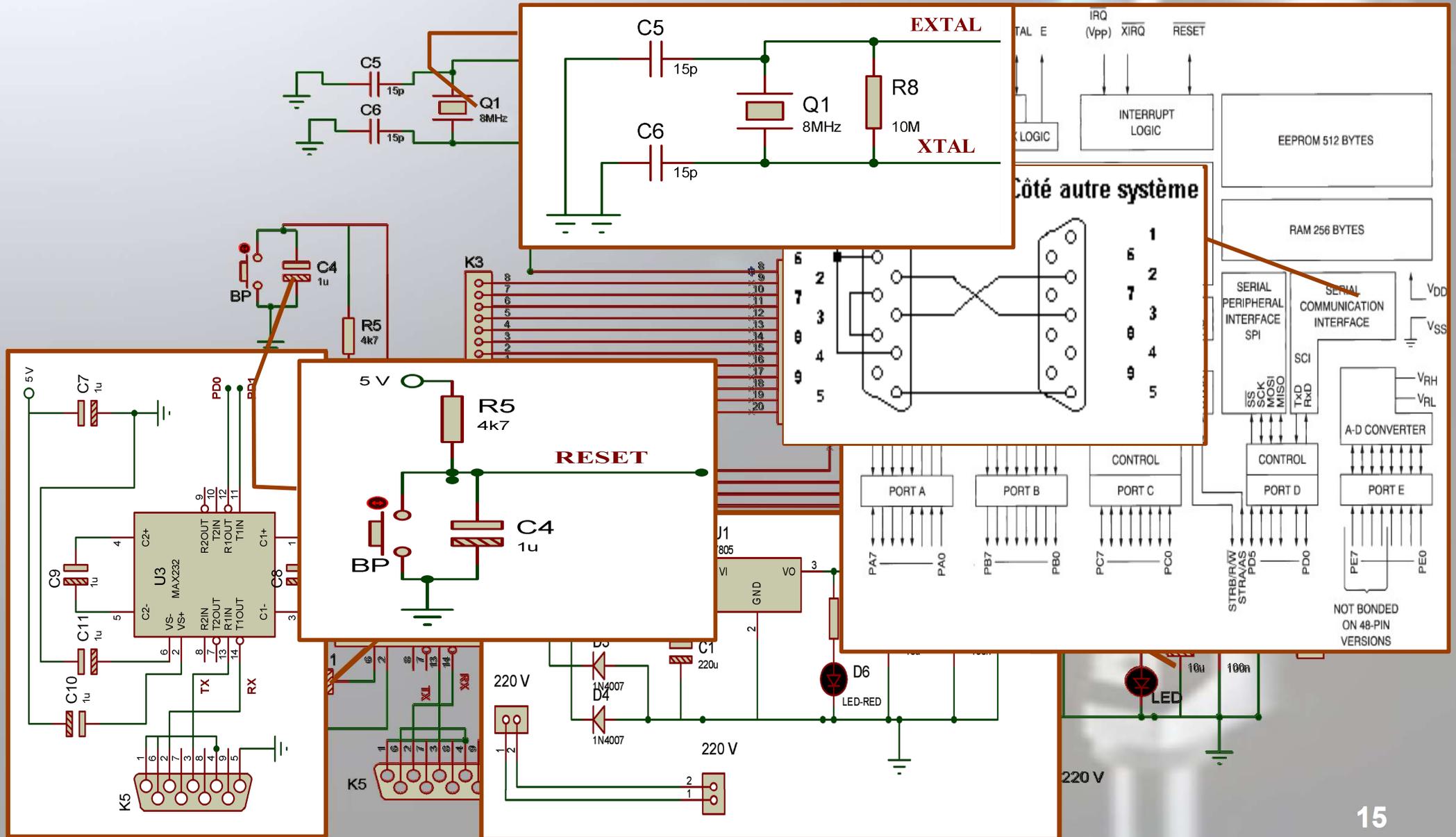
Puis calculé les distances globales:

$$g(i, j) = d(i, j) + \min(g(i-1, j), g(i-1, j-1), g(i, j-1))$$



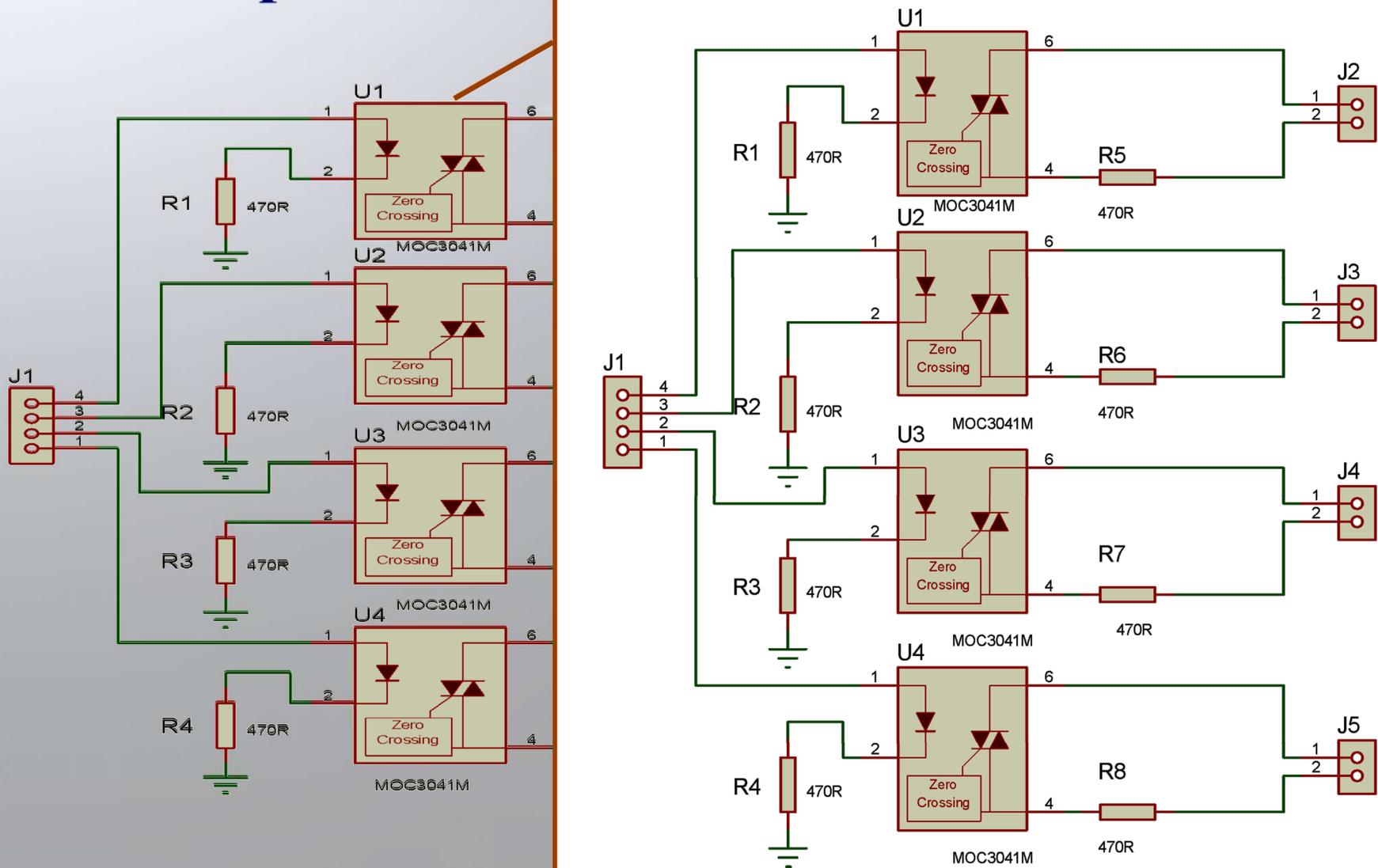
# Carte d'interface et mode de transmission

## La carte d'interface



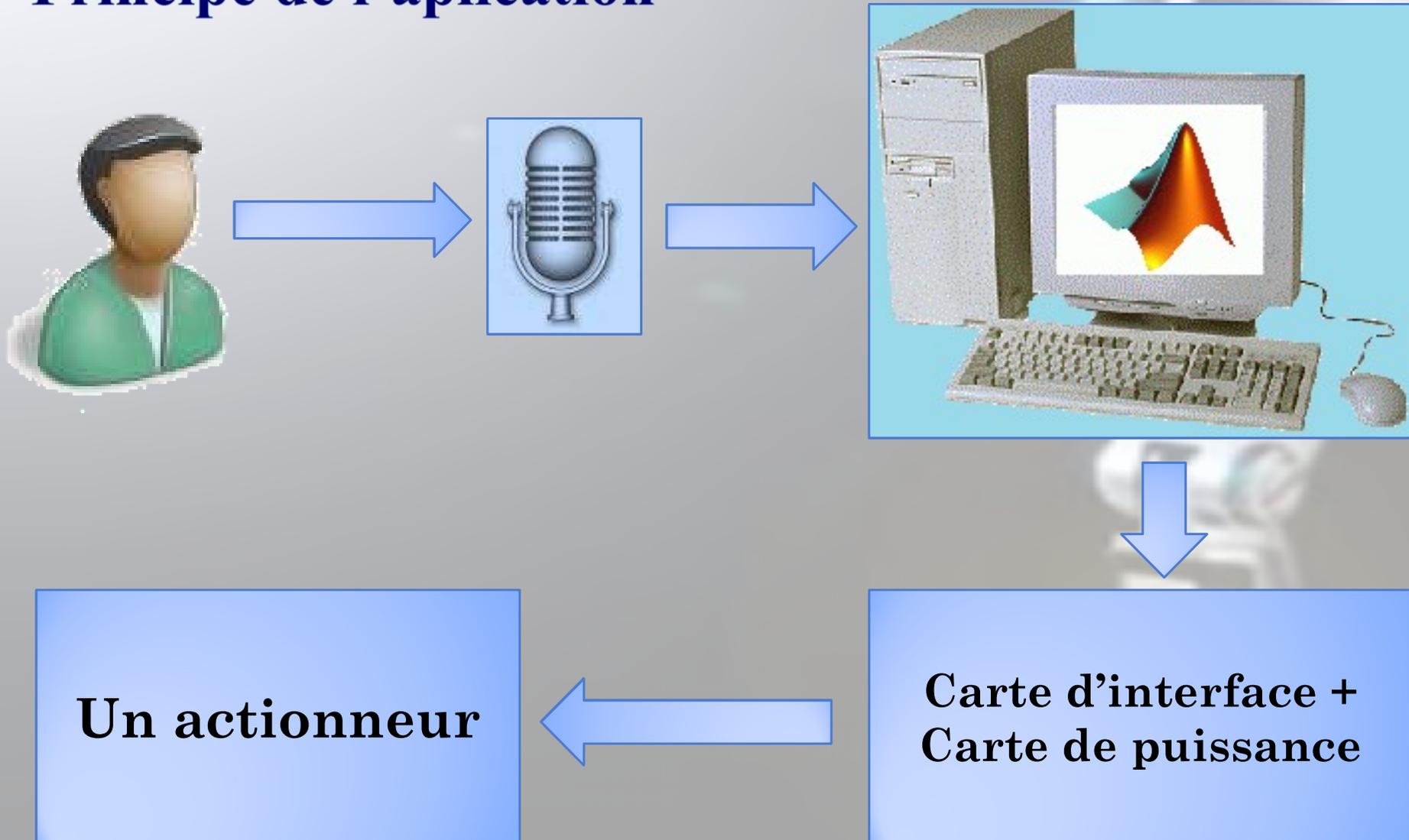
# Carte d'interface et mode de transmission

## Carte de puissance



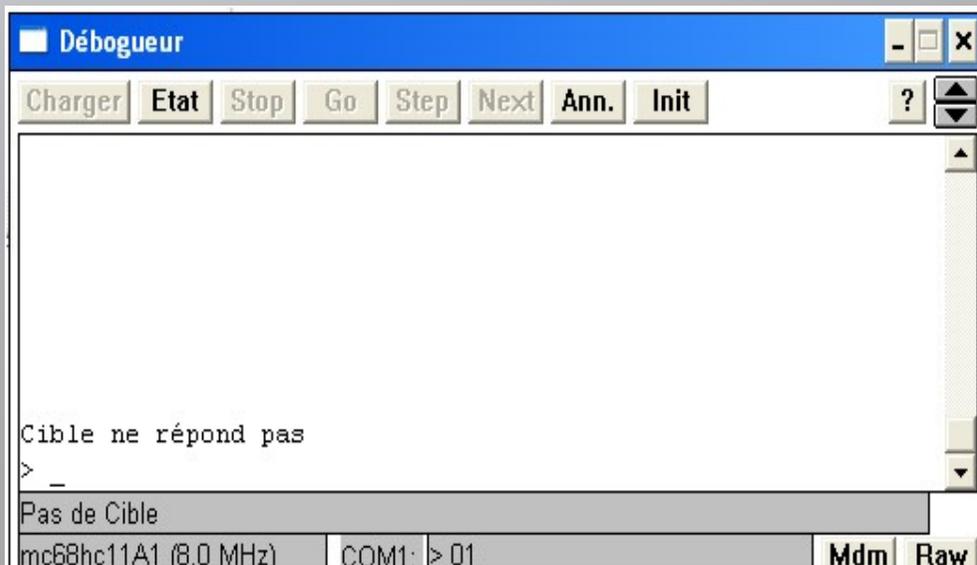
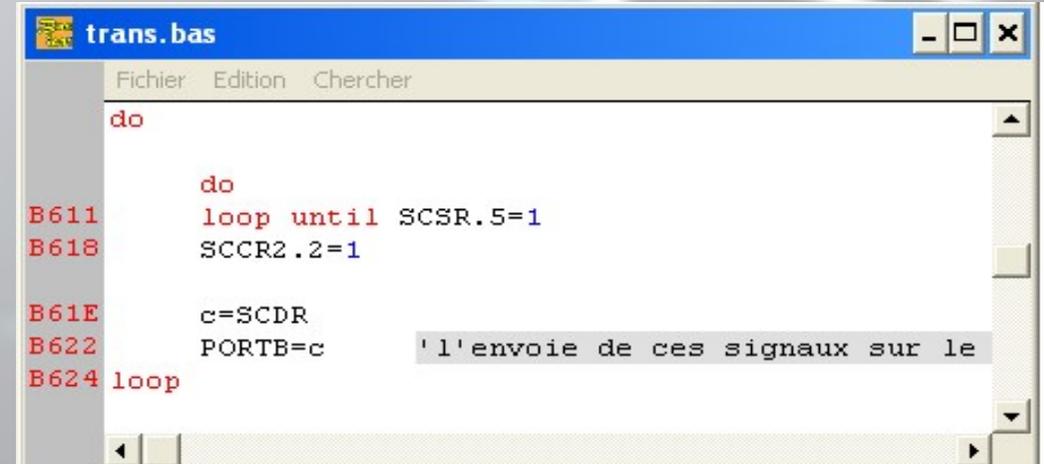
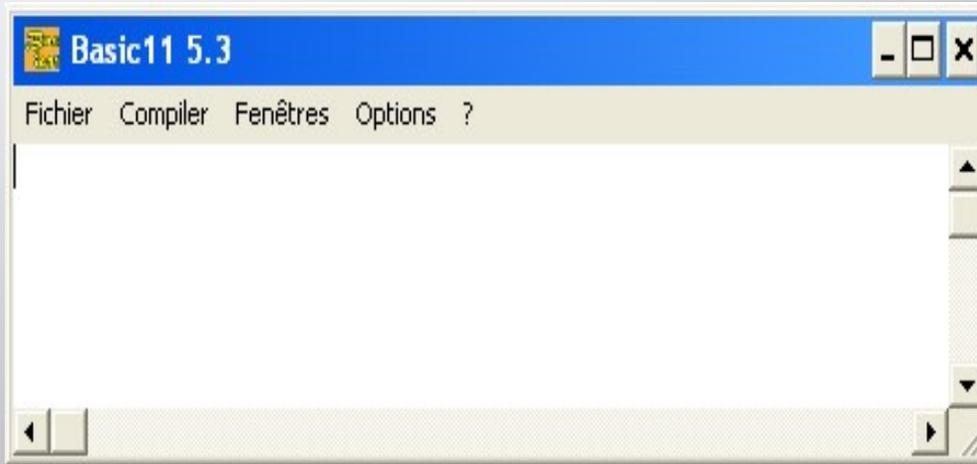
# Application et résultats

## Principe de l'application



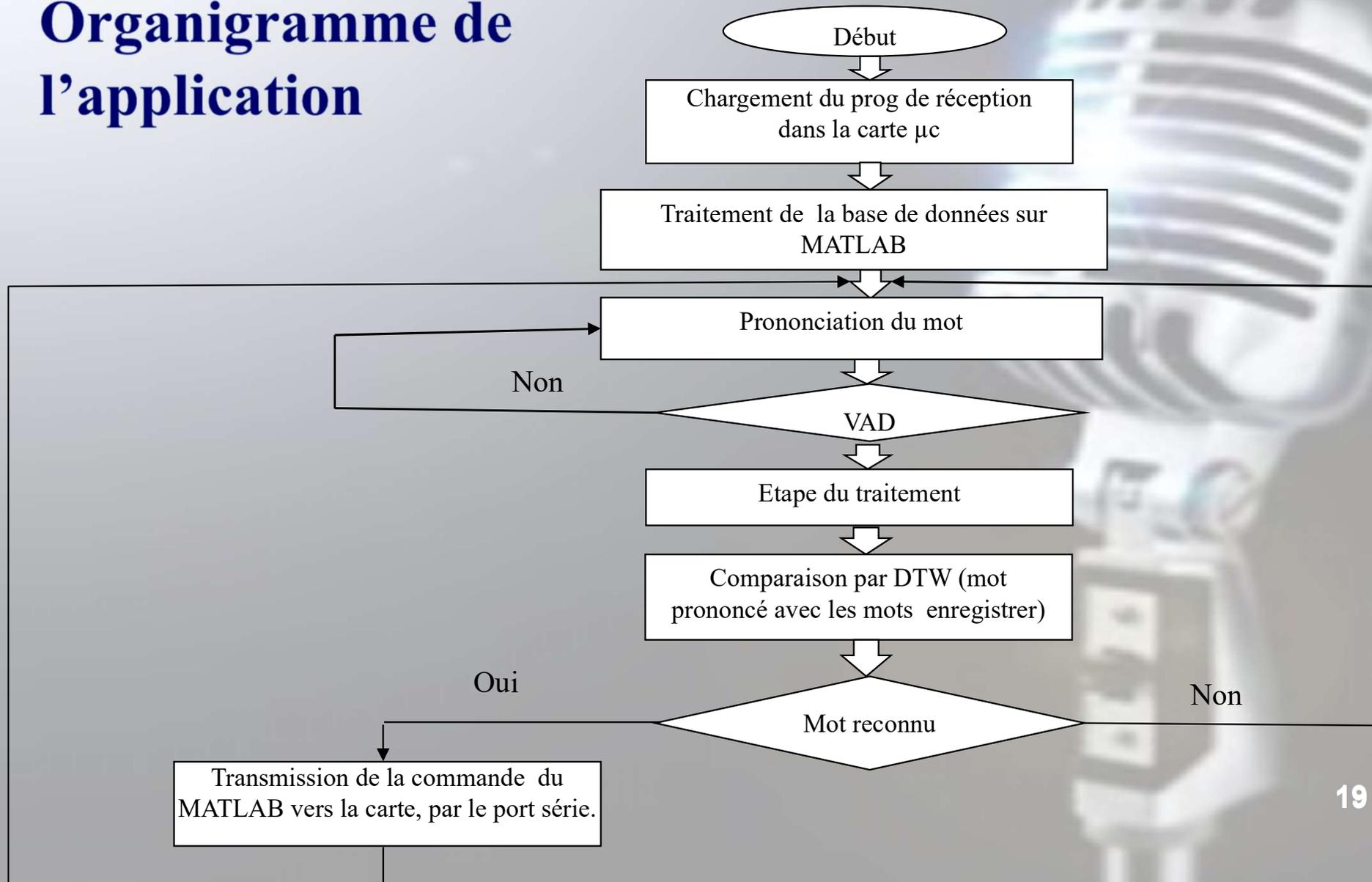
# Application et résultats

## Environnement de communication avec le microcontrôleur



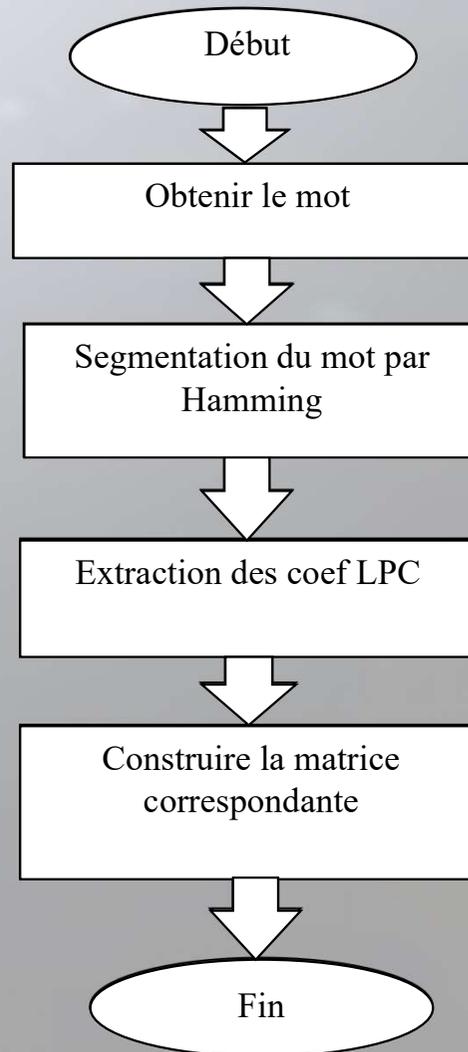
# Application et résultats

## Organigramme de l'application



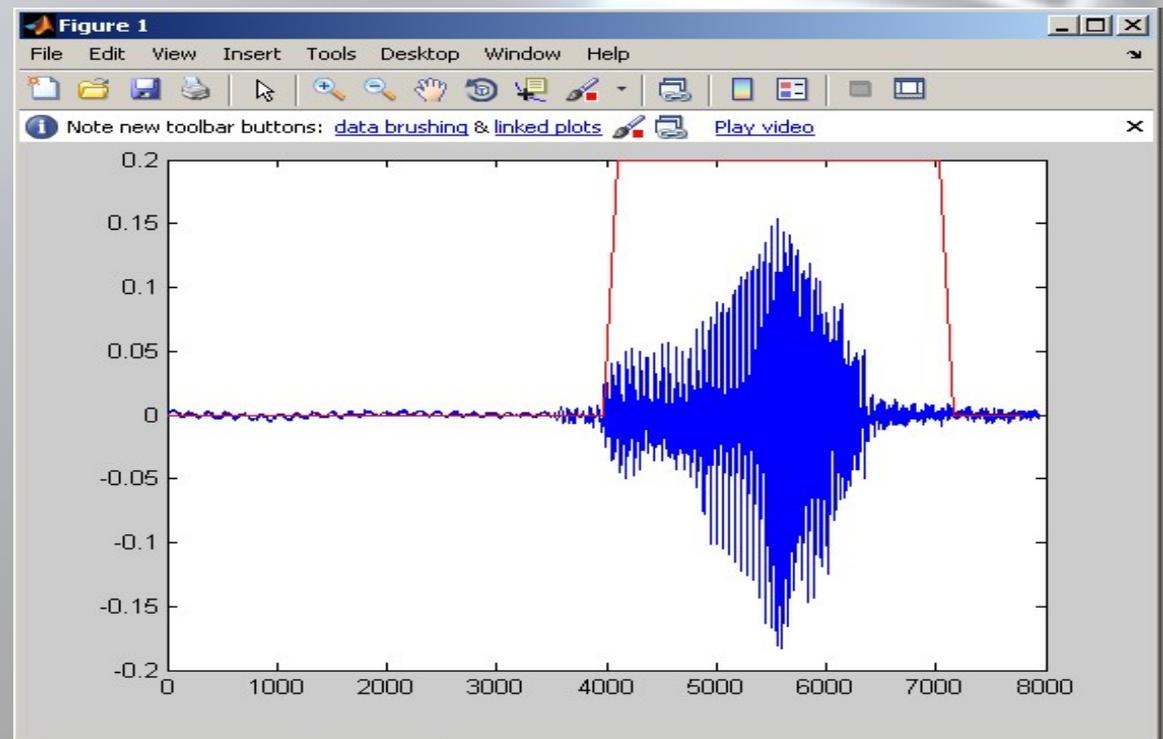
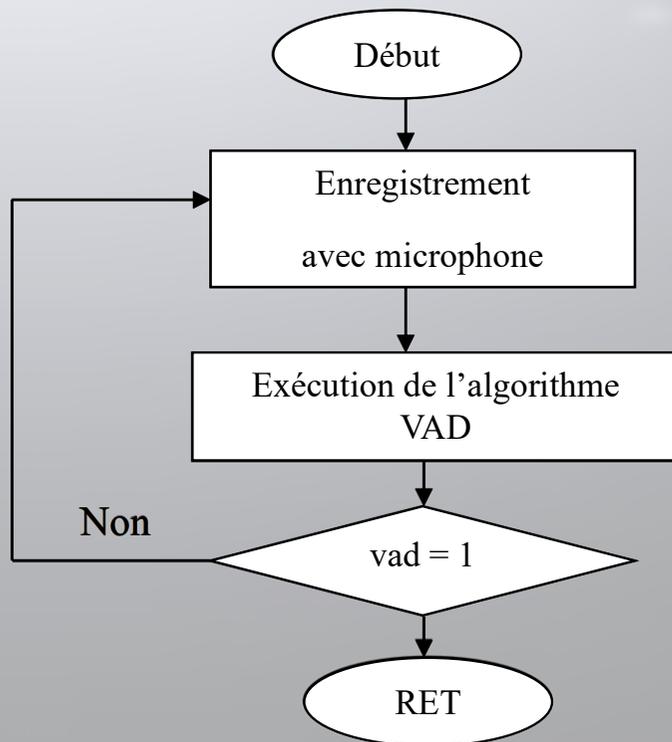
# Application et résultats

## Construction de la base de donnée



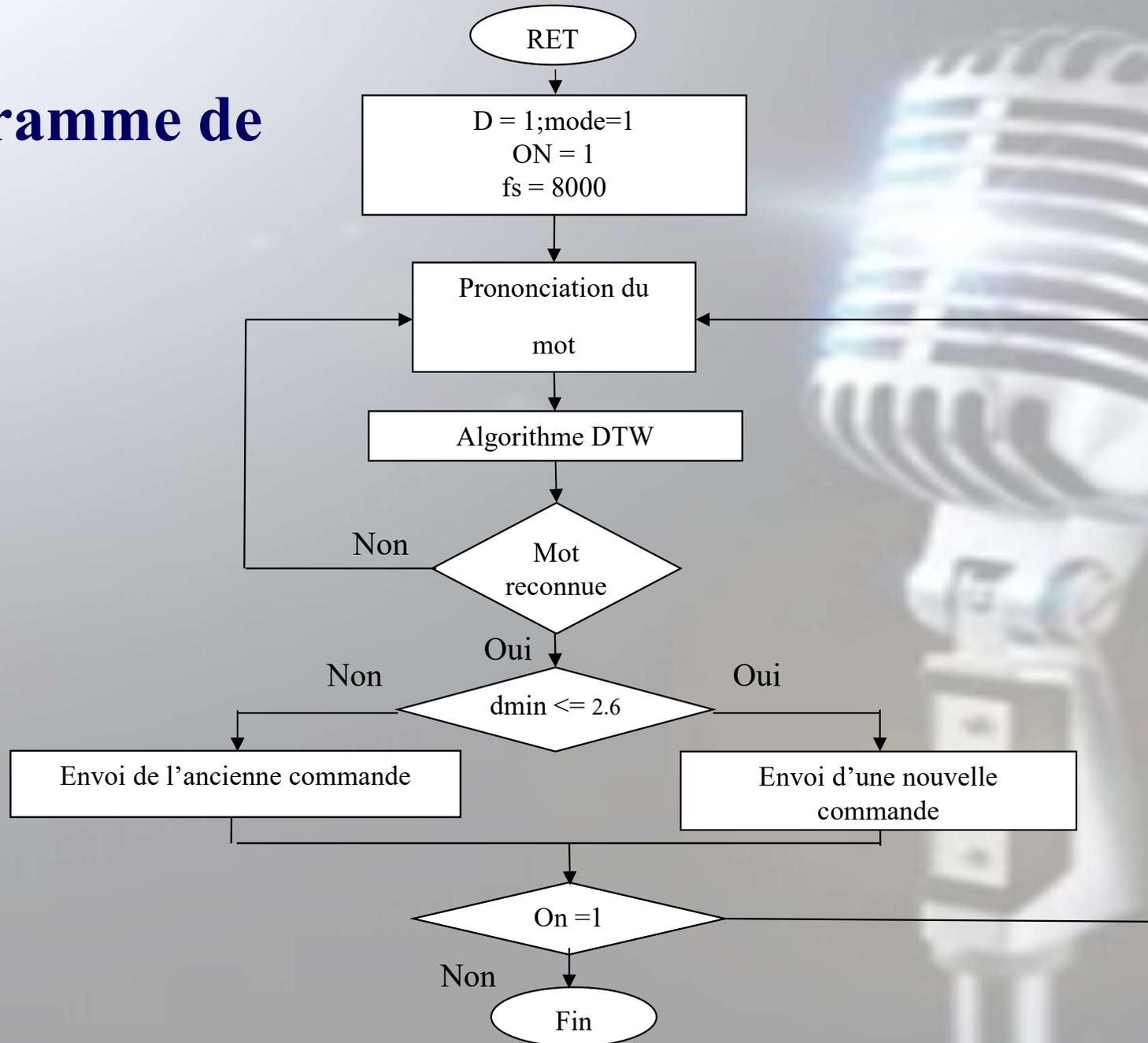
# Application et résultats

## Organigramme de la VAD



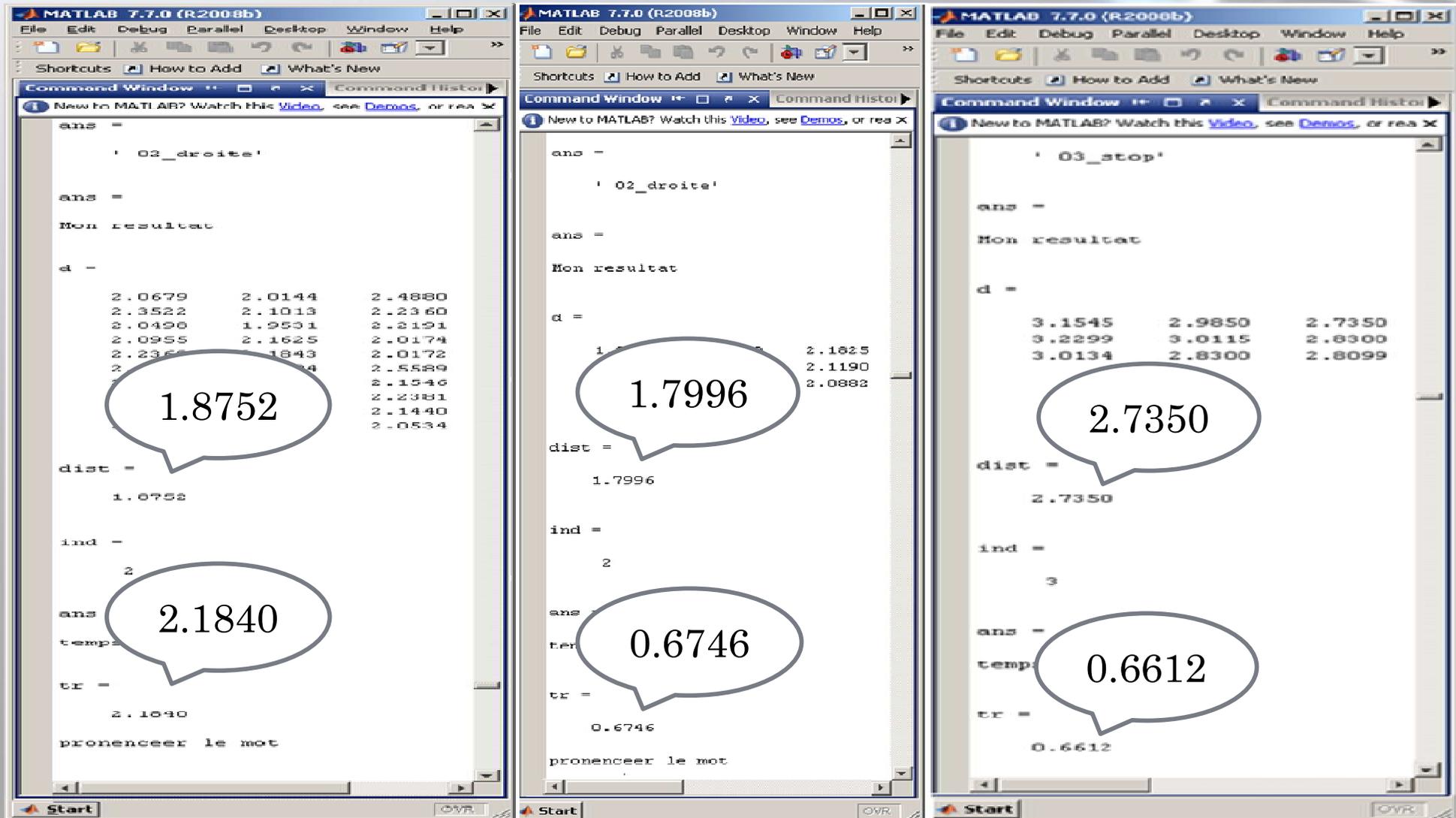
# Application et résultats

## Organigramme de la DTW



# Application et résultats

## Résultats de l'application





*Problème rencontré*

# *Conclusion*





**MERCI pour votre attention**

MERCI pour votre attention

# Liste des Figures

|  |    |
|--|----|
| <b>Figure I.1</b> : L'appareil phonatoire.   | 05 |
| <b>Figure I.2</b> : Vue de dessus de la section du larynx.   | 06 |
| <b>Figure I.3</b> : Le système auditif.  | 07 |
| <b>Figure I.4</b> : Enregistrement numérique d'un signal acoustique.   | 11 |
| <b>Figure I.5</b> : Evolution temporelle d'un son voisé et son spectre obtenu par transformée rapide de Fourier (FFT). | 12 |
| <b>Figure I.6</b> : Fenêtre de Hamming.  | 13 |
| <b>Figure I.7</b> : Schéma générale de la reconnaissance de forme.   | 16 |
| <b>Figure I.8</b> : Schéma de la reconnaissance du mot isolé.  | 16 |
| <b>Figure II.1</b> : Bruit de fond + signal vocal.   | 19 |
| <b>Figure II.2</b> : Energie du signal.  | 19 |
| <b>Figure II.3</b> : Fenêtrage de la parole par un seuil énergétique.  | 20 |
| <b>Figure II.4</b> : Bruit de fond + Parole.   | 20 |
| <b>Figure II.5</b> : Energie du signale.   | 20 |
| <b>Figure II.6</b> : Déroulement global de l'algorithme de détection.  | 21 |
| <b>Figure II.7</b> : Déroulement de la détection d'hypothèse.  | 21 |
| <b>Figure II.8</b> : Signal avec sa fonction de densité de probabilité (pdf).  | 25 |
| <b>Figure II.9</b> : Distribution gaussienne.  | 26 |
| <b>Figure II.11</b> : Organigramme de la VAD.  | 31 |
| <b>Figure II.12</b> : Déformation temporelle linéaire.   | 32 |
| <b>Figure II.13</b> : Matrice des distances.   | 33 |
| <b>Figure II.14</b> : Etablissement d'une zone de recherche.   | 34 |
| <b>Figure II.15</b> : Les contraintes locales.   | 35 |
| <b>Figure II.16</b> : Contrainte de parcours.  | 35 |
| <b>Figure II.17</b> : Comparaison entre deux signaux A et B par DTW.   | 36 |
| <b>Figure II.18</b> : Organigramme de la DTW.  | 37 |
| <b>Figure III.1</b> : Schéma interne du 68HC11A1.  | 41 |
| <b>Figure III.2</b> : Format de données.   | 44 |
| <b>Figure III.3</b> : Connecteur RS 232.   | 47 |

|   |    |
|---|----|
| <b>Figure III.4</b> : Schéma d'un connecteur DB9 et signification des broches.      | 48 |
| <b>Figure III.5</b> : Liaison à trois fils.   | 51 |
| <b>Figure III.6</b> : Synoptique d'une liaison série.                               | 52 |
| <b>Figure III.7</b> : Schéma de fonctionnement de l'UART.                           | 52 |
| <b>Figure III.8</b> : Schéma électrique de la carte de commande a base de 68HC11A1. | 54 |
| <b>Figure III.9</b> : Le schéma développé de la carte d'isolation optique.          | 56 |
| <b>Figure III.10</b> : La carte de puissance.                                       | 57 |
| <b>Figure IV.1</b> : Schéma illustrant les étages fonctionnels du projet.           | 59 |
| <b>Figure IV.2</b> : Schéma significatif des étapes.                                | 60 |
| <b>Figure IV.3</b> : Les fenêtres de logiciel BASIC11.                              | 61 |
| <b>Figure IV.4.a</b> : L'organigramme de réception de données.                      | 62 |
| <b>Figure IV.4.b</b> : Le programme de réception de données sur BASIC11.            | 62 |
| <b>Figure IV.5</b> : Le compilateur de BASIC11.                                     | 63 |
| <b>Figure IV.6</b> : Chargement de programme de BASIC11.                            | 63 |
| <b>Figure IV.6</b> : Organigramme du bloc de la détection d'activité vocale.        | 65 |
| <b>Figure IV.7</b> : Détection d'une activité vocale.                               | 65 |
| <b>Figure VI.8</b> : Organigramme de la reconnaissance et d'envoi de la commande.   | 67 |
| <b>Figure IV.9</b> : Résultats du MATLAB pour la reconnaissance du mot « droite ».  | 68 |
| <b>Figure IV.10</b> : Résultats du MATLAB pour la reconnaissance du mot « stop ».   | 69 |

# LISTE DES ABRÉVIATIONS

|      |  |
|------|--|
| VAD  | Voice Activity Detection (détection d'activité vocale).                              |
| DTW  | Dynamic Time Warping (déformation temporelle dynamique).                             |
| FFT  | Fast Fourier Transform (transformée rapide de Fourier).                              |
| LPC  | Linear Predictive Coding (codage prédictif linéaire).                                |
| LPCC | Linear Prediction Cepstral Coefficient (coefficient linéaire de prédiction Cepstral) |
| MFCC | Mel-Frequency Cepstral Coefficient (Mel-Frequency coefficient Cepstral)              |
| HMM  | Hidden Markov model (modèle de Markov caché)   |
| ANN  | Artificial Neural Network (réseau de neurones artificiel).                           |
| GMM  | Gaussian Mixture Model (modèle de mélange gaussien)                                  |
| SNR  | Signal to Noise Ratio (rapport signal-bruit)   |
| LRT  | Likelihood Ratio Test (rapport de vraisemblance)                                     |
| PDF  | Probability Density Function (fonction de densité de probabilité)                    |
| MLE  | Maximum Likelihood Estimation (Estimation du maximum de vraisemblance)               |
| MMSE | Minimum Mean Square Error (l'erreur quadratique moyenne)                             |
| LTD  | Linear Time Deformation (déformation temporelle linéaire)                            |
| SCI  | Serial Peripheral Interface (interfaces séries asynchrones)                          |
| SPI  | Serial Communications Interface (interfaces séries synchrones)                       |
| NRZ  | Non Retour à Zéro  |
| RS   | Recommanded Standard   |
| EIA  | Electronique Industries Association  |
| DCD  | Data Carrier Detect  |
| RxD  | Receive Data   |
| TxD  | Transmit Data  |
| DTR  | Data Terminal Ready  |
| GND  | Ground   |
| DSR  | Data Set Read  |
| RTS  | Request To Send  |
| CTS  | Clear To Send  |

|       |  |
|-------|--|
| RI    | Ring Indicator   |
| CCITT | Comité Consultatif International des Téléphones et Télégraphes                         |
| ETTD  | Equipement Terminal de Traitement de Données   |
| DTE   | Data Terminal Equipment  |
| ETCD  | Equipement de Terminaison de Circuit de Données  |
| DCE   | Data Communication Equipment   |
| UART  | Universal Asynchronous Receiver Transmitter (émetteur-récepteur asynchrone universel). |

### **Conclusion générale**

Nous avons entamé ce mémoire de fin d'étude avec la perspective de réaliser un système autonome qui permet de commander un actionneur à l'aide du signal vocal, pour cela nous avons commencé notre travail par faire un rappel sur le signal de la parole et les principaux axes de travail, en s'intéressant de près à la reconnaissance vocal et particulièrement à la reconnaissance du mot isolé.

Nous avons centré notre étude en premier lieu sur le développement de l'algorithme VAD « **V**ocal **A**ctivité **D**étection » cela pour rendre notre système autonome avec une réponse plus aux moins à temps réel en faisant ressortir le mot de commande, du bruit de fond et ça nous a donné des résultats très satisfaisants, par la suite nous avons implémenté l'algorithme DTW « **D**ynamic **T**ime **W**arping » ceci nous a permis de réaliser la reconnaissance du mot isolé avec taux de reconnaissance assez satisfaisants mais qui peut être amélioré avec l'implémentation d'autres algorithmes, tel que les chaînes de MARKOV caché, les ANN ; et l'utilisation du matériel adéquat, un microphone sensible au bruit, un milieu insonore pour la construction de la base de donnée.

Afin de commander l'actionneur nous avons utilisé les algorithmes énoncés dans le paragraphe précédant implémenté sur un PC grâce aux langages de programmation MATLAB et pour l'envoi de la commande du PC vers l'actionneur nous avons utilisé une carte d'interface basée sur le microcontrôleur 68HC11A1 de MOTOROLA

A la fin de notre mémoire on peut dire que nous avons réussi à atteindre l'objectif fixé au départ quelque lacune, la réalisation de cette application nous a permis de nous familiariser avec différents thèmes, du traitement du signal de la parole à savoir le Codage de la parole, Extraction des caractéristiques, la Synthèse vocal, la Reconnaissance de la parole, la Carte d'interface et la Transmission série.

# Annexes

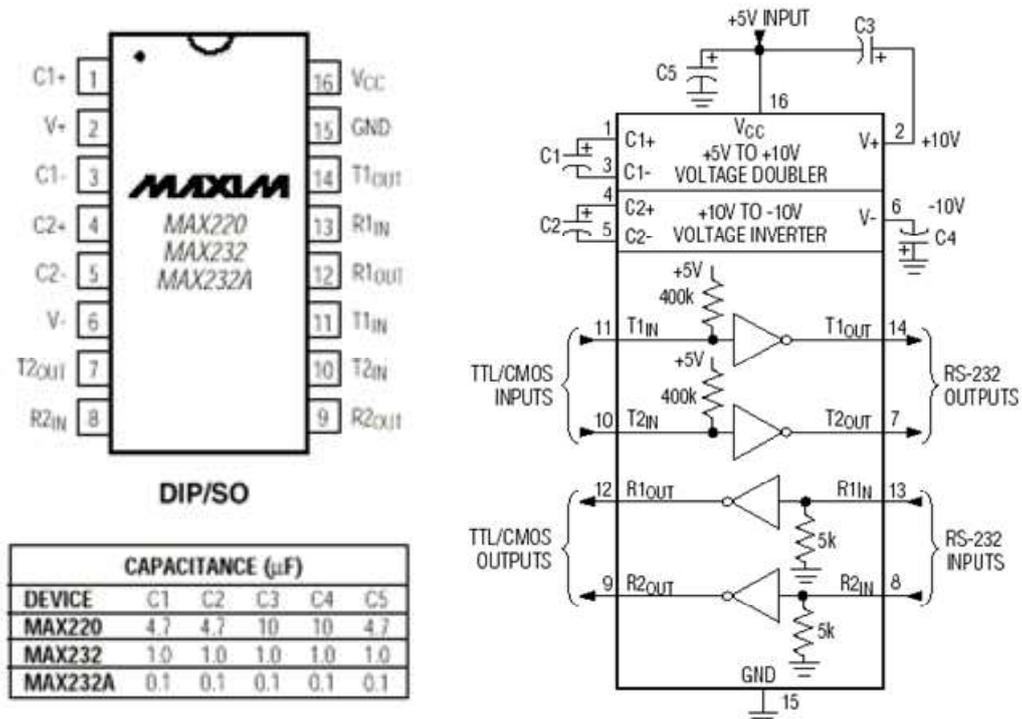


Figure : Schéma représentant le MAX 232.

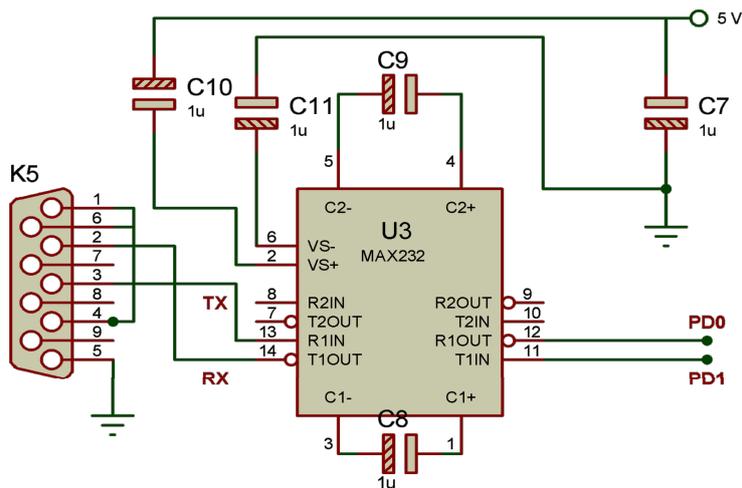


Figure : Schéma de principe de la carte interface.

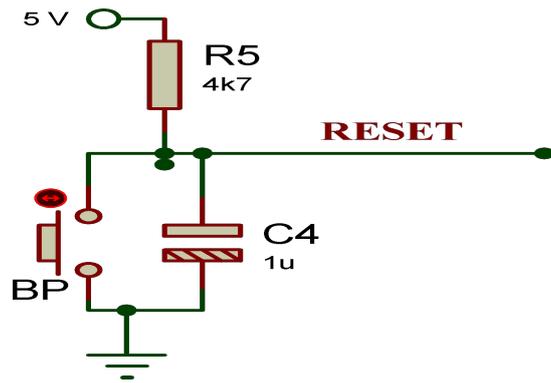


Figure : Circuit de RESET.

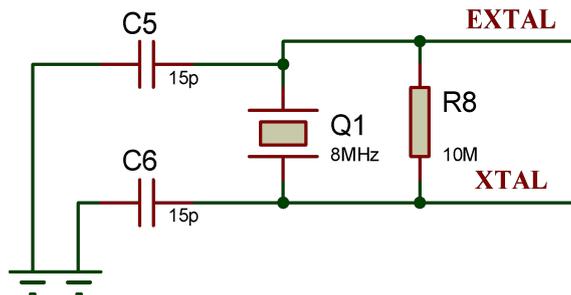


Figure : Circuit d'horloge.

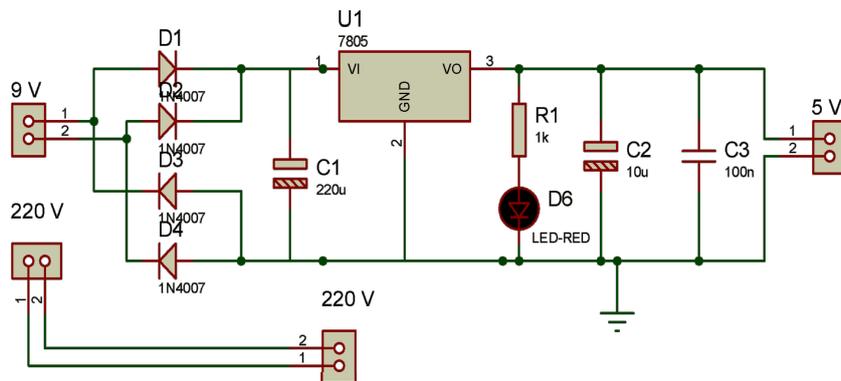


Figure : Schéma développé de l'alimentation.

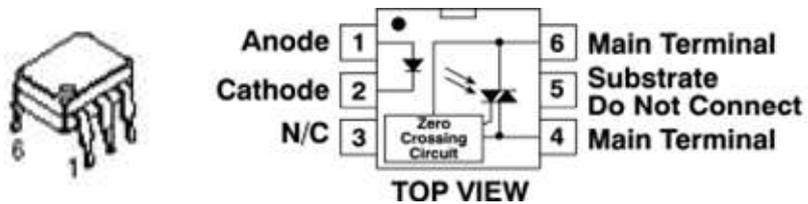


Figure : Schéma interne du MOC3041.

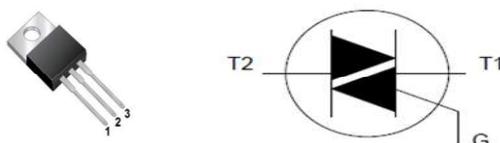
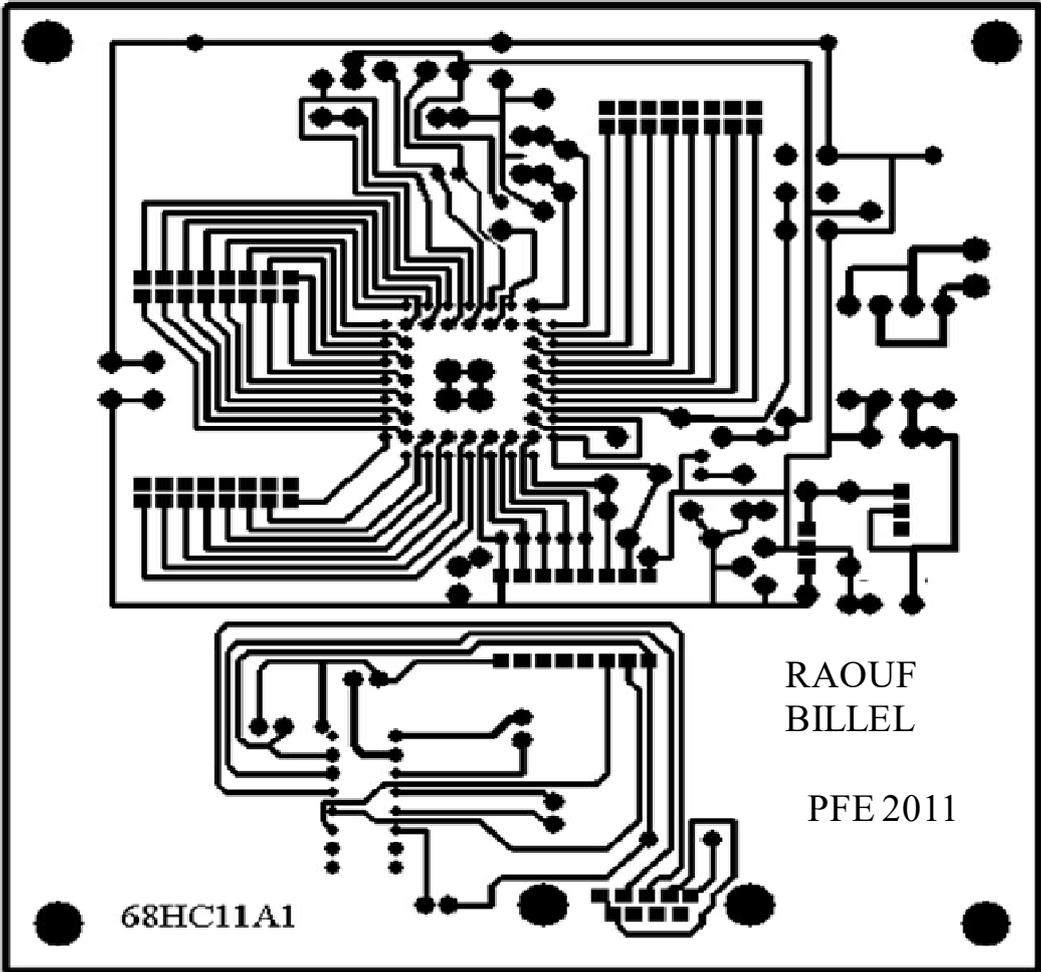
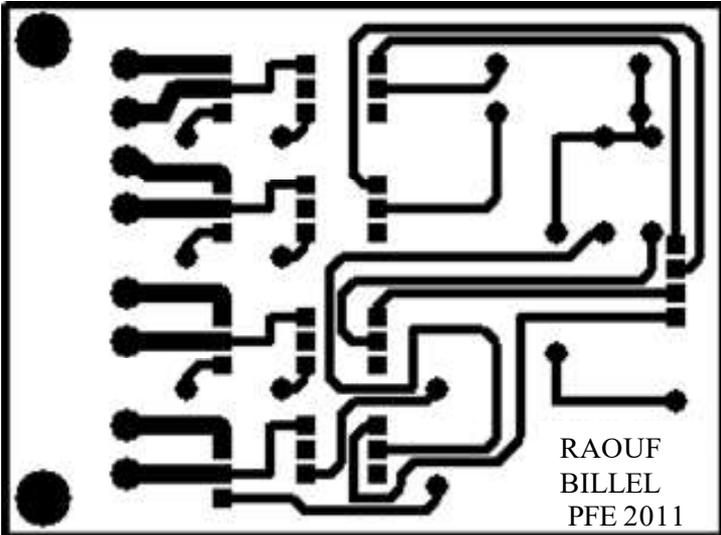


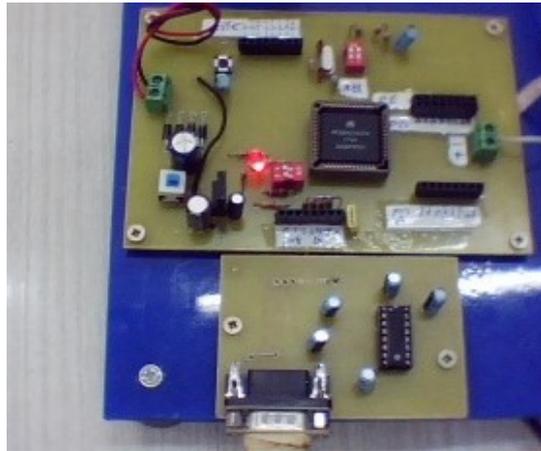
Figure : Schéma interne du triac.

Circuit imprimé de la carte de commande :



Circuit imprimé de la carte de puissance :





**Figure :** La carte de commande.



**Figure :** La carte de puissance.

- **La carte de commande**

R1:1 K $\Omega$ .

RES : pont de 4 résistances des 10 K $\Omega$ .

R2, R3, R4, R5, R6, R7, R9, R10 : 4.7 K $\Omega$ .

R8 :10 M $\Omega$ .

C1=220  $\mu$ F 16 V.

C2: 10  $\mu$ F 16 V.

C3: 100 nF.

C4, C7, C8, C9, C10, C11: 1  $\mu$ F 100 V.

C5, C6: 15 pF.

LED rouge 3 mm.

U1: 68HC11A1.

U2: 7805.

U3 : MAX 232.

2 Commutateur.

1 Oscillateur 8 MHz (Quartz).

1 Bouton poussoir.

2 Connecteur de 8 pins.

2 Connecteur de 16 pins.

1 Support 52 broches.

1 Support 16 broches.

2 connecteurs DB9 femelle.

1 connecteur DB9 male.

D1 à D4 : diode 1N4007.

Transformateur 9 V.

- **La carte de puissance**

R1 a R8 : 470  $\Omega$ .

U1 a U4 : MOC3041M.

U5 a U8 : BT137.

