Abstract

“In this project we are interested in the design of an automatic voice commands system which allows us to transfer commands to a machine or robot arm using microcomputer, this system can be used by people who can’t use keyboard or mouse to give simple commands to a computer due to a physiological disability or someone his hands are in operating conditions. The mathematical tools to be employed in this system are the Hidden Markov Models (HMM), its models are generally used for the processing of random physical signals with sequential nature vis-à-vis the information they conduct in this case is speech signal.”

Key words: HTK, HMM, HTK Toolkit, voice commands, Voice Signal, Speech recognition, Speech Production.

Résumé :

“On s’intéresse dans ce travail à la conception d’un système de reconnaissance automatique de la parole permettant de donner des commandes vocales à une machine ou un bras de robot via un micro-ordinateur un tel système peut être utilisé par une personne qui ne peut pas utiliser un clavier ou une souris pour donner des commandes simples a un ordinateur suite à un handicape physiologique ou parce qu’elle a les mains occupées. Les outils mathématiques à employer dans un tel système sont les Modèles de Markov Cachés (HMM), ses modèles sont largement utilisés pour le traitement des signaux physiques aléatoires ayant une nature séquentielle vis-à-vis l’information qu’ils conduisent ce qui es le cas du signal parole. ”

Mots clés: HTK, HMM, HTK Toolkit, commande vocale, signal parole, reconnaissance de la parole, production de la parole.

ملخص:

” نحن مهتمون في هذا المشروع بتصميم نظام تلقي أوامر صوتية يسمح لنا بنقل الأوامر إلى آلة او ذراع روبوت باستخدام معالج كمبيوترمصغر ، ويمكن استخدام هذا النظام من قبل الأشخاص الذين لا يستطيعون استعمال لوحة المفاتيح أو الفارة بسبب إعاقة فسيولوجية أو في حالة شخص يداه مشغولتان. الأدوات الرياضية التي سيتم استخدامها في هذا النظام هي نماذج ماركوف المخفية(HMM) وتستخدم نماذجها بشكل عام لمعالجة الإشارات الفيزيائية العشوائية ذات الطبيعة المتسلسلة بخصوص الاشارات التي يتم نقلها في هذه الحالة هي إشارة الكلام“ .

الكلمات المفتاحية: HTK , HMM , HTK Toolkit , التحكم الصوتي , إشارة الكلام , التعرف الصوتي .

Table of contents

Table of figures

Abbreviations

MFCC Mel-scale Frequency Cepstral Coefficient

DFT Discrete Fourier Transform

PLP Perceptual Linear Predictive.

LPC Linear Predictive Coding

AR Auto Regressive

ARMA Auto Regressive Moving Average

MLE Maximum Likelihood Estimation:

LDA Linear Discriminant Analysis

NLDA Non-linear Discriminant Analysis

HMM Hidden Markov Models

HTK Hidden Tool Kit

General introduction.

Speech processing nowadays is a fundamental part of science the engineer. This science is a combination of digital signal processing and language processing, this field of science known since 1960s, that part of time seen a huge development of communications systems and tools. The particular importance of speech processing in this general context explained because speech is known as the carrier of information in our human societies. Continuous speech recognition presents new difficulties because we do not know the number of words that make up a sentence, or the boundaries of each word. For these reasons, during this thesis, our work consists in building a system of continuous speech recognition using special software named HTK. This study is based on a probabilistic approach, where we use hidden Markov models (**HMM**) to model the vectors of the acoustic parameters, and give the decision after calculating the maximum likelihood between the acoustic model and the linguistic model, which will provide a recognition error. For its part, the HTK will give us a report of different recognition rate, substitutions, deletions and insertion.

In danger environments inaccessible to humans, such as deep underwater environments, toxic environments, etc., the remote command of robots is a tool of paramount importance. As part of this work, we propose to examine a voice command from an operator. This command must be executed by the manipulator by a movement action in the three axes. The stages of this work can be summarized as follows:

Development of a speech acquisition system. This system should allow the recording of commands of the type: UP, DOWN, LEFT, RIGHT, FORWARD, BACKWARD, START, STOP, etc. It should be open to allow the database to be enriched with new orders as they arise.

Development of a module for extracting the characteristics of each order. The most discriminating characteristics should be retained as the output of this module to ensure a clear separation between the different orders

This dissertation contains 4 chapters,

- The first chapter discuss the all the process of generating the voice signal from human's mouth and the steps to generating speech till receiving by the ear and processing the received speech signal.

- The second chapter interests in signals processing and how the voice signal is being produced.

- The third chapter will discuss the theory of the HIDDEN MARKOF MODELS (HMMS) and the processing of the voice signals with its characteristics.

- The fourth chapter discuss the practical side of the HMMS by building a model used to command a robot, then some other examples of speech recognition utilities in real life.

Chapter I Speech production

I.1 Introduction

Speech is the human being's ability to communicate thought through articulate sounds. Due to its importance, the word has always preoccupied scientists. So, some of the sciences concerned with the study of speech are already hundreds of years old.

In this chapter we will see a general approach on the identity of the speech signal starting with everything related to its production including the architecture of the vocal apparatus, the mechanism of phonation and the classification of phonemes will also be discussed in the content of this chapter. We will also see the acoustic level of the speech signal, particularly its spectrum, its fundamental frequency, its energy…. etc.

I.2 Speech production

I.2.1 Architecture of the vocal device

The vocal device (Figure I-1), or phonatory system, consists of four fundamental elements that work in close synergy to produce acoustic signals. These are, in the order they are developed:

- The blower.
- The vibrator.
- The sound bodies.
- The articulator system.

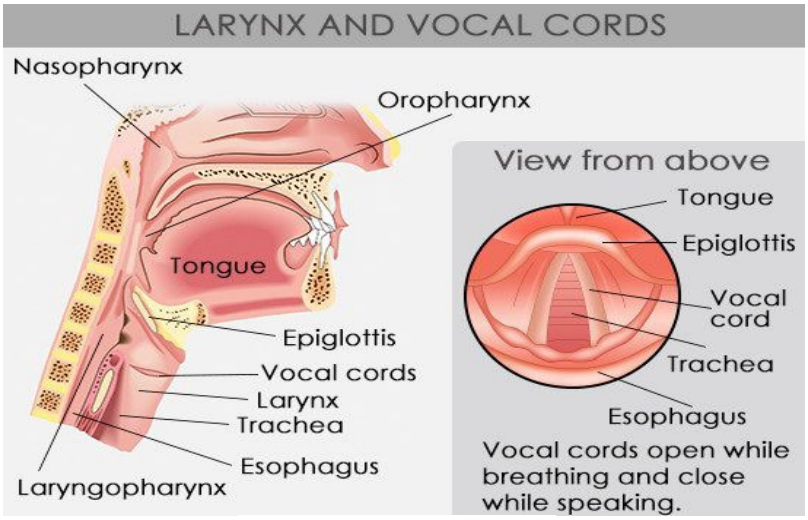

Figure I-1 the vocal device

The blower is made up of a reservoir of air, the lungs, powered by the muscles of the thorax and abdomen, and a tube, the trachea artery, which conducts air to the vocal cords; the vibrator is the larynx, which generates air waves; the sound body is made up of a complex set of resonators, of which the pharynx and the mouth are the main ones; the articulator system, finally, is made up of fixed and mobile elements which allow the shape of the laryngeal wave to be greatly modified. All these elements are placed under the close dependence of the central nervous system, which ensures synchronism and coordination [1].

I.2.1.1 **The blower and the vibrator**

Air is the raw material of the voice. While the functioning of our vocal apparatus is often compared to that of a musical instrument, it should be described as that of a wind instrument. Indeed, by expelling pulmonary air through the trachea, the respiratory system acts as a wind tunnel. This is the "phonatory breath" produced either by lowering the rib cage or as part of vocal projection by the action of the abdominal muscles [2].

I.2.1.2 **The Larynx**

The larynx is the upper end of the trachea artery, located at the height of the sixth cervical vertebra (in adults) (Figure I-2). It is an assembly of articulated cartilages, linked together by ligaments and muscles (including the vocal cords), the whole being lined with a mucous membrane [1].

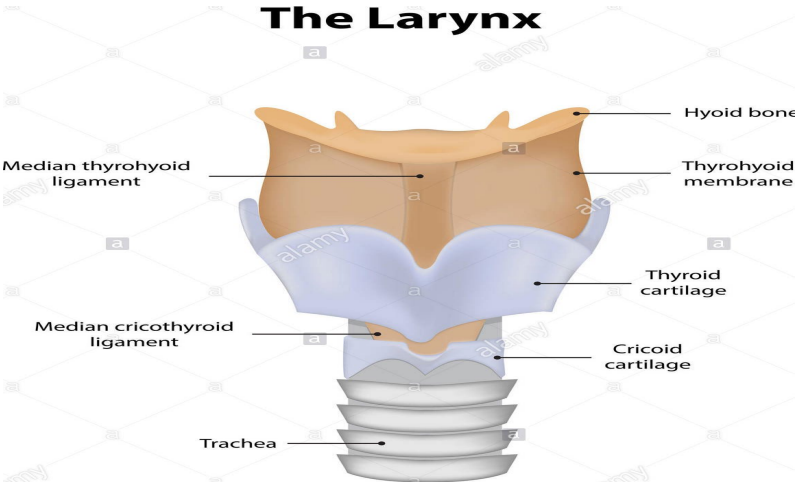

Figure I-2 Schematic anterior view of the larynx (Left) and its section, top view (right)

I.2.1.3 The sound bodies

The resonators of the phonatory system are primarily responsible for the tone of the voice. Their originality compared to the resonance boxes of traditional musical instruments is their ability to change, thanks to a dense and elaborate muscular network, - in large proportions, and very [1].

I.2.2 Mechanism of phonation

One of the most important characteristics of the speech signal is the nature of the arousal. There are two basic types of excitations which produce voiced and unvoiced sounds.

I.2.2.1 Voiced sounds phonation

Voiced sounds are produced from an excitation which activates on the vocal tract and which consists of a series of periodic pulses of air supplied by the larynx. The vocal cords at the beginning are closed. Under the continuous pressure of the air coming from the lungs they gradually open delivering this potential energy. During this opening the air speed and the kinetic energy increase until the elastic tension of the vocal cords equals the force of separation of the air stream. At this point the opening of the glottis is maximum. The kinetic energy that has been accumulated as elastic tension in the vocal cords begins to narrow this opening and furthermore Bernoulli's force further accelerates the abrupt closure of the glottis.

This periodic process is characterized by a frequency specific to each person, known as the fundamental frequency and it can vary from 80 to 200 Hz for a male voice, from 150 to 450 Hz for a female voice and from 200 to 600 Hz for a child voice [4]. This fundamental frequency can vary due to factors related to stress, intonation and emotions. The timbre of the voice is determined by the relative amplitudes of the harmonics of the fundamental. The intensity of the sound emitted is related to the air pressure upstream of the larynx. All these aspects for a voiced sound can be observed in (Figure I-3).

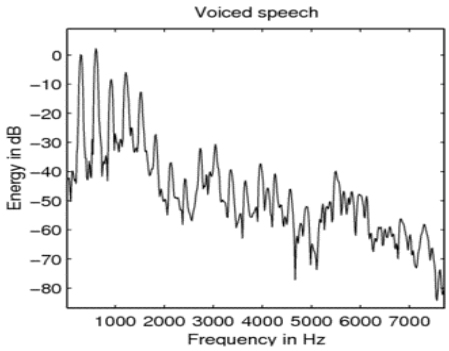

Figure I-3 spectrum of voiced sound

I.2.2.2 **Phonation of non-voiced sounds**

Unvoiced sounds are generated by the passage of air in a narrow constriction located at a point in the vocal tract. They are generated without the contribution of the larynx and do not have a periodic structure [3]. These characteristics of an unvoiced sound can be seen in (Figure I-4).

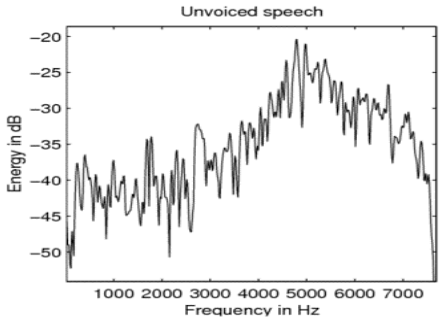

Figure I-4 spectrum of non-voiced sound

I.2.3 Classification of phonemes

There are several ways to classify phonemes. A phoneme is stationary or continuing if the configuration of the vocal tract does not change during sound production. A phoneme is non-continuous if during its production there are changes in the configuration of the vocal tract [3].

I.2.3.1 **The vowels**

Vowels are voiced, continuous sounds, normally with the greatest amplitude among all the phonemes and they can vary a lot in duration, between 40 and 400 ms. Oral vowels are produced without the intervention of the nasal cavity while for nasal vowels, the nasal passage is coupled to the oral cavity and the production of sound occurs done through the mouth and through the nostrils at the same time. The vowels are differentiated into three groups according to the position of the

curvature of the tongue and the degree of constriction induced in the vocal tract. Differentiated into three groups according to the position of the curvature of the tongue and the degree of constriction induced in the vocal tract. Time and frequency domain analysis reveals several characteristics acoustics that help in the classification of each sound. Time domain analysis shows that vowels are quasi-periodic sounds due to excitement.

Vowels can be identified by the locations of their closers in the field frequency. The position of the first two closers is sufficient to characterize the majority of vowels, the third forming is needed just for a few. The position of closers of higher frequency remains almost unchanged and does not provide useful information for identification.

I.2.3.2 **Diphthongs**

Diphthongs involve movement from an initial vowel to another vowel final. So, diphthongs are essentially non-continuous sounds. The difference between a diphthong and the two component individual vowels is that the duration of the transition is greater than the duration of each vowel. In addition, the initial vowel is more along as the final vowel. In speech the two vowels making up a diphthong may not be fully realized which accentuates the idea of non-stationarity which characterizes diphthongs.

I.2.3.3 **Semi-consonants**

Semi-consonants are non-continuous, voiced sounds that have spectral characteristics similar to vowels. We can see the semi-consonants as transient sounds approaching, reaching and after moving away from a target position. The duration of transitions is comparable to the time spent in target position.

I.2.3.4 **Consonants**

Consonants are sounds for which the vocal tract is narrower during production, compared to vowels. Consonants involve both forms of excitement for the vocal tract and they may or may not be continuous.

I.3 The hearing system

In the context of speech processing, a good knowledge of the mechanisms of hearing and the perceptual properties of the ear are as important as a mastery of production mechanisms. Sound waves are collected by the hearing instrument, which causes sensations hearing. These pressure waves are analyzed in the inner ear which sends to the brain the resulting nerve impulse; the physical phenomenon thus induces a psychic phenomenon thanks to a complex physiological mechanism. The hearing aid consists of the outer ear, the middle ear, and the inner ear (Figure I-5). The ear canal connects the pinna to the eardrum: it is an acoustic tube of uniform section closed at one end, its first resonance mode is located around 3000 Hz, which increases the sensitivity of the hearing system in this frequency range. The inner ear mechanism (hammer, caliper, anvil) allows impedance adaptation between the air and the liquid medium of the inner ear. The vibrations of the caliper are transmitted to the liquid the cochlea.

This contains the basilar membrane which transforms the vibrations mechanical nerve impulses. The membrane widens and thickens as it goes as you approach the apex of the cochlea.it is the support of the organ of Corti which is made up of about 25,000 hair cells connected to the auditory nerve.

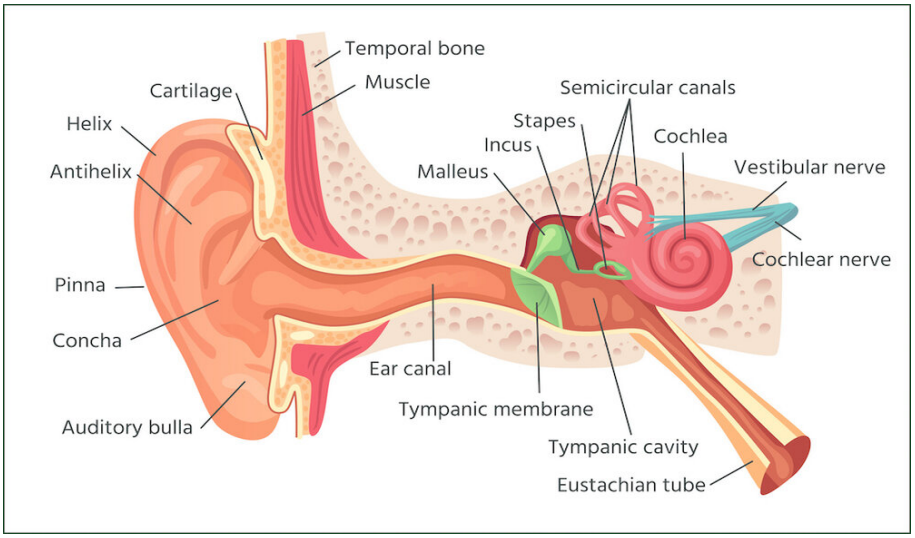

Figure I-5 Cross section of the human hearing system.

It remains very difficult nowadays to say how auditory information is processed by the brain. On the other hand, we were able to study how it was ultimately perceived, in the context of a specific science called psychoacoustics. Without wishing to go into too much detail on the major contribution of psychoacousticians in the study of speech, it is interesting to know the most significant results.

Thus, the ear does not respond equally to all frequencies. (Figure I-4) shows the human hearing field, delimited by the threshold curve of hearing and that of the threshold of pain. Its upper limit in frequency (≈ 16000 Hz, variable according to the individuals) fixes the maximum useful sampling frequency for an auditory signal (≈ 32000 Hz) [5].

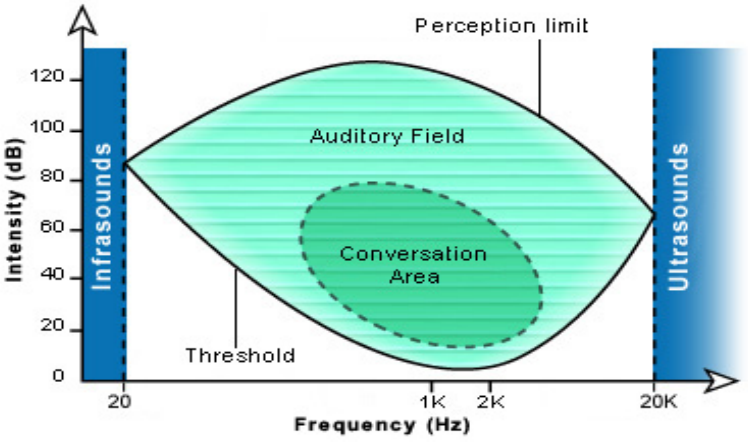

Figure I-6 The human hearing fields

I.4 Speech Sound level

Speech physically appears as a change in the air pressure caused and emitted by the articulatory system. Acoustic phonetics studies this signal by firstly transforming into an electrical signal using the appropriate transducer: the microphone (itself associated with a preamplifier).

Nowadays, the resulting electrical signal is most often digitized. It can then be subjected to a set of statistical treatments which aim to highlight the features acoustic: its fundamental frequency, its energy, and its spectrum. Every acoustic trait is itself intimately linked to a perceptual quantity: pitch, intensity, and timbre. The digitization operation requires successively: filtering guard, sampling, and quantification [5].

I.4.1 audiograms

Sampling transforms the continuous-time signal into a discrete-time signal defined at sampling times, an audiogram is a representation of sounds and words spoken, each sound we hear

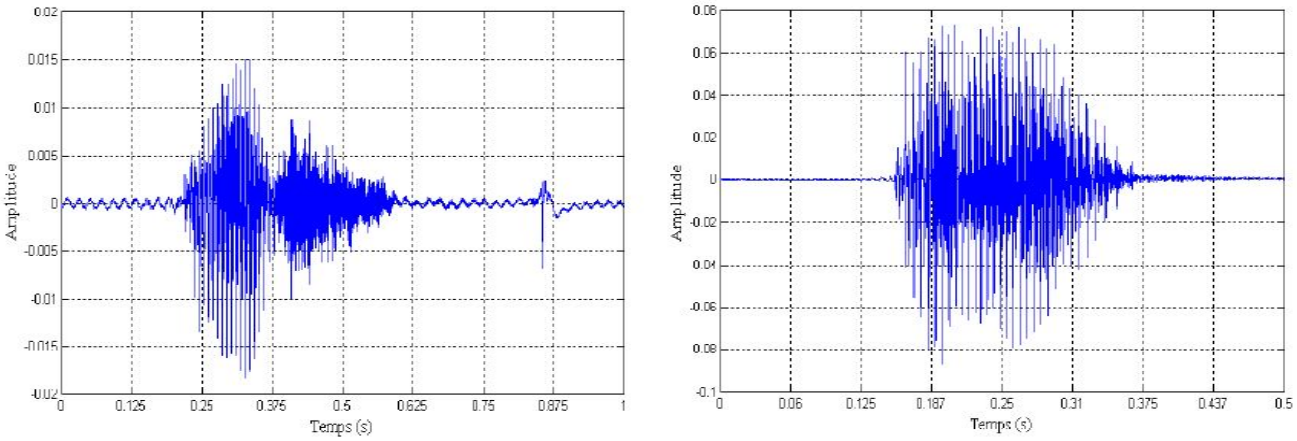

Figure I-7 Audiogram on the left the word "left" and on the right the word "down"

has a different pitch and loudness and a different representation, we have also an essential characteristic which results from the mode of representation is the bit rate, expressed in bits per second (b / s), required for transmission or recording of the voice signal. Conventional telephone transmission requires a bit rate of 8 kHz x 8 bits = 64 kb / s, in principle, the transmission or recording of an audio signal requires the order of 48 kHz x 16 bits = 768 kb /s [5]. (Figure I-6)

represents the temporal evolution, or audiogram, of the voice signal for the words 'left', and 'down'.[5]

I.4.2 Fundamental frequency

An analysis of a speech signal is not complete until it is measured the temporal evolution of the fundamental frequency [5]. In acoustics, the fundamental frequency is the first order harmonic of a sound. The fundamental frequency or F0 is the frequency at which vocal chords vibrate in voiced sounds. This frequency can be identified in the sound produced, which presents quasi-periodicity, the pitch period being the fundamental period of the signal. Pitch is more often used to refer to how the fundamental frequency is perceived.

I.5 Conclusion

Speech processing is now a fundamental component of the science of engineer, but before treatment it is necessary to know the characteristics and behavior of the speech signal. In this chapter we have reviewed the main features acoustico-phonetics of the speech signal. At the phonetic level we have seen the principle general speech production, the architecture of the vocal apparatus, the phonation… etc. At the acoustic level, we were mainly interested in the evolution time of the speech signal and frequency fundamental.

Chapter II The Speech Signal

II.1 Introduction

Speech processing is a science at the intersection of digital signal processing and language processing. Speech has the peculiarity, compared to others information processing signals, to be produced and perceived instantly by the brain, and for that the speech processing tends to replace these functions by automatic systems (Figure II-1):

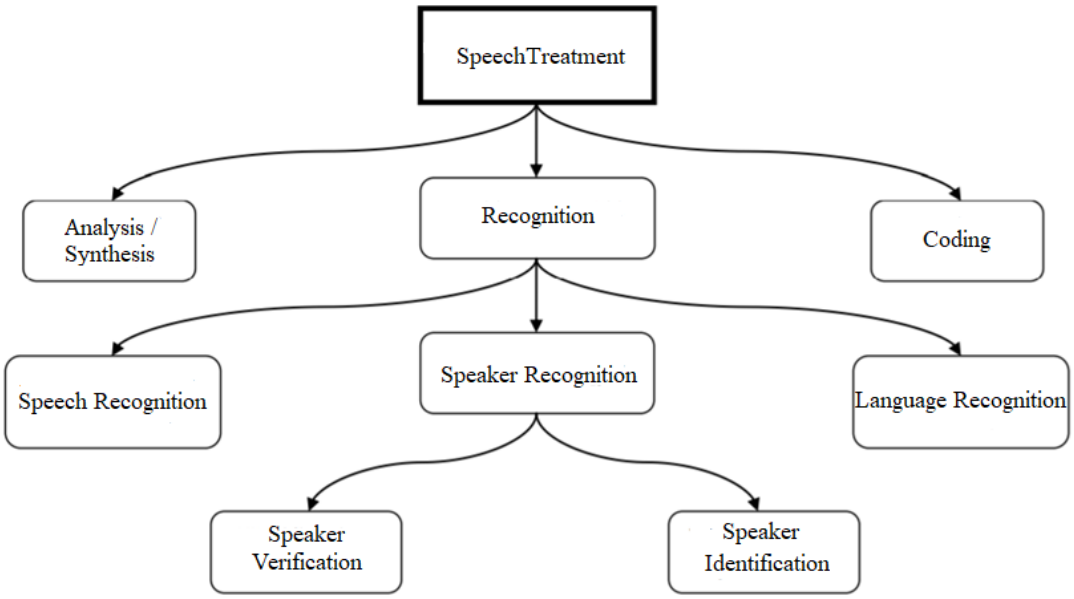

Figure II-1 speech processing

Speech analyzers highlight the characteristics of the voice signal such as that it is produced. They are used either as a basic component of coding systems, recognition or synthesis.

Speech recognizers decode the information carried by the voice signal from data provided by the analysis. We classify them according to the information we seek to extract from the voice signal. The recognition of the speaker which is the identification or speaker verification (determines who, among a finite and predetermined number of speakers, produced the analyzed signal).

There is also the recognition of the speaker dependent on the text (the sentence to be pronounced to be recognized is fixed from the design of the system), recognition with dictated text (The sentence to be pronounced is fixed during the test), and independent recognition of the text (The sentence to be pronounced is not specified). In addition to the single-speaker, multi-speaker, or speaker-independent speech recognizer, speaker to recognize the voice of a person, a finite group of people, or recognize anyone.

Finally, the recognizer of isolated words, recognizer of connected words, and continuous speech recognizer, depending on whether the speaker separates each word by a silence, whether he continuously pronounces a series of predefined words, or whether he pronounces any sequence of words continuously. Speech synthesizers is a computer technique of sound synthesis which allows you to create artificial speech from any text. To get this result, it is based both on linguistic processing techniques, in particular for transform the spelling text into an unambiguously pronounceable phonetic version, and on signal processing techniques to transform this phonetic version into sound digitized listenable on a loudspeaker. There are two types of synthesizers: synthesizers of speech from a digital representation, the reverse of analyzers, whose mission is to produce speech from the digital characteristics of a voice signal such as that obtained by analysis, and speech synthesizers from a representation symbolic (text or concept) ¸ inverse of speech recognizers and capable in principle of pronouncing any sentence without it being necessary to have it pronounced by a human speaker beforehand.

Finally, the role of coders is to allow the transmission or storage of speech with a reduced flow, which naturally requires careful consideration of the speech production and perception properties [17].

II.2 The Speech

Speech is a continuous signal, finite energy, not stationary. Its structure is complex and variable over time.

II.3 Classification of speech sounds

A simplified decomposition of the speech signal should show two types of sounds: voiced and unvoiced.

Voiced sounds, such as vowels, are produced by the passage of air from lungs through the trachea which vibrates the vocal cords. This mode, which represents 80% of phonation time, is generally characterized by near-periodicity, high energy and fundamental frequency (pitch). Typically, the fundamental period of the different voiced sounds varies between 2ms and 20ms.

Unvoiced sounds, such as some consonants, in this case strings vocal cords do not vibrate, air passes at high speed between the vocal cords. The signal produced is equivalent to white noise.

II.3.1 Speech signal parameters

The voice signal is generally characterized by three parameters: its frequency fundamental, its energy and its spectrum.

II.3.1.1 **Energy**

It is represented by the intensity of the sound which is related to the air pressure upstream of the larynx. The amplitude of the speech signal varies over time depending on the type of sound, and its energy in a frame is given by:

$E = \sum_{n=0}^{N-1} S^2(n)$ With N: the size of the frame.

II.3.1.2 **Spectrum**

The spectral envelope or spectrum represents the intensity of the voice according to the frequency, it is generally obtained by a short-term Fourier analysis. The quasi-stationarity of the speech signal makes it possible to implement methods efficient analysis and modeling used for short-term speech signal processing over windows of duration generally between 20ms and 30ms called frames, with an overlap between these windows which ensures the temporal continuity of

the characteristics of analysis. The short-term Fourier transform (TFCT) of a sampled signal is by definition of the transform of the weighted signal.

$$\hat{S}(k)=\hat{S}\left(f=\frac{k}{N}\right)=\sum_{n=0}^{N-1}s(n)\cdot w(n)\cdot\exp(-j2\pi nk/N)\qquad 0\le k\le N-1$$

With: N: The number of points collected, S(k): Complex spectrum, S(n): Segment analyzed.

II.4 Modeling Speech Production

The absence of coupling between the glottis and the vocal tract makes it possible to model separately the source and the production system. For voiced sounds, the source is a train periodic wave of a particular form (rapid rise in pressure followed by more gradual). This wave train is modeled by the response of a low pass of order 2 at real poles and whose cutoff frequency is of the order of 100Hz. [18]. For unvoiced sounds the source is white noise. The sound is emitted through the opening of the lips, which represents an acoustic load, lip radiation can be modeled by transmittance:

$$R(z)=C\left(1-z^{-1}\right)$$

Which expresses that the pressure of the wave observed at a certain distance from the lips is proportional to the derivative of the volume flow at the lips.

II.5 Conclusion

The voice signal can only be considered as quasi stationary over intervals of time of limited duration. We are therefore led to consider successive slices and to estimate an AR or ARMA model for each slice.

Chapter III The HIDDEN MARKOV MODELS

Introduction

Automatic speech recognition has been an active field of study since the beginning of 1950s. It is clear that an effective speech recognition tool will facilitate the interaction between humans and machines. The possible applications associated with such tools are numerous and are set to experience a great revolution. Most applications of speech recognition can be grouped into four categories: command and control, access to databases or search for information, voice dictation and automatic speech transcription.

The most used technology for over 20 years is based on statistical models: Hidden Markov Models (HMM) capable of simultaneously model the frequency and temporal characteristics of the speech signal.

III.2 automatic speech processing

Automatic speech processing is a rich and difficult field of research, the speech signal is complex because its structure results from the interaction between the production of sounds and their perception by the ear. Researchers in the field have succeeded despite everything, to achieve some recognition or synthesis systems that can be used in real conditions. In the following sections, we are interested in automatic speech processing. with a view to its recognition. We start first with a view of the levels of complexity linked to its recognition. Then, we detail the different stages of its processing which aim to extract characteristic coefficients of the signal. We study particularly the MFCC Mel-scale frequency cepstral coefficient) used in the majority of current recognition systems. Finally, we describe the approach statistic for recognition based on the use of hidden Markov models (HMM) [6].

III.2.1 complexity levels

To fully understand the problem of automatic speech recognition, it is good to understand the different levels of complexity and the different factors that make a difficult problem [5].

First, there is the problem of intra- and inter-speaker variability. Is the system speaker dependent (optimized for a specific speaker) or speaker independent (can recognize any user)?

Obviously, speaker-dependent systems are easier to develop and are characterized by better recognition rates than independent systems of the speaker since the variability of the speech signal is more limited. This dependence to the speaker is however acquired at the cost of specific training for each user. This is not always possible, however. For example, in the case of applications of robot and drones, it is obvious that the system must be able to be used by anyone and must therefore be independent of the speaker. Although the basic methodology remains the same, this independence to the speaker is however obtained by the acquisition of many speakers. which are used simultaneously to the training of models capable of extracting all the major characteristics. An intermediate solution sometimes used is to develop systems capable of adapting quickly to the new speaker.

Does the system recognize single words or continuous speech? Obviously, it is easier to recognize isolated words well separated by periods of silence (sil) than to recognize the sequence of words constituting a sentence. Indeed, in the latter case, not only the border between words is no longer known but, moreover, the words become strongly articulated. In the case of continuous speech, the level of complexity also varies depending on whether it is read text, spoken text or, much more difficult, natural language with its hesitations, grammatically incorrect sentences, false starts, etc. Another problem, which begins to be well mastered, concerns the recognition of keywords in free speech. In the latter case, the vocabulary to be recognized is relatively small and well defined but the speaker is not forced to speak in isolated words.The size of the vocabulary and its degree of confusion are also important factors. Small vocabularies are obviously easier to recognize than large ones [5].

III.2.2 Extraction of parameters

To process a voice signal for recognition we must first extract parameters that characterizes the information hidden behind this signal. For this the Mel-scale frequency cepstral coefficient (MFCC) are used. For steps executed of the extraction of these coefficient are illustrated in (Figure III-1)

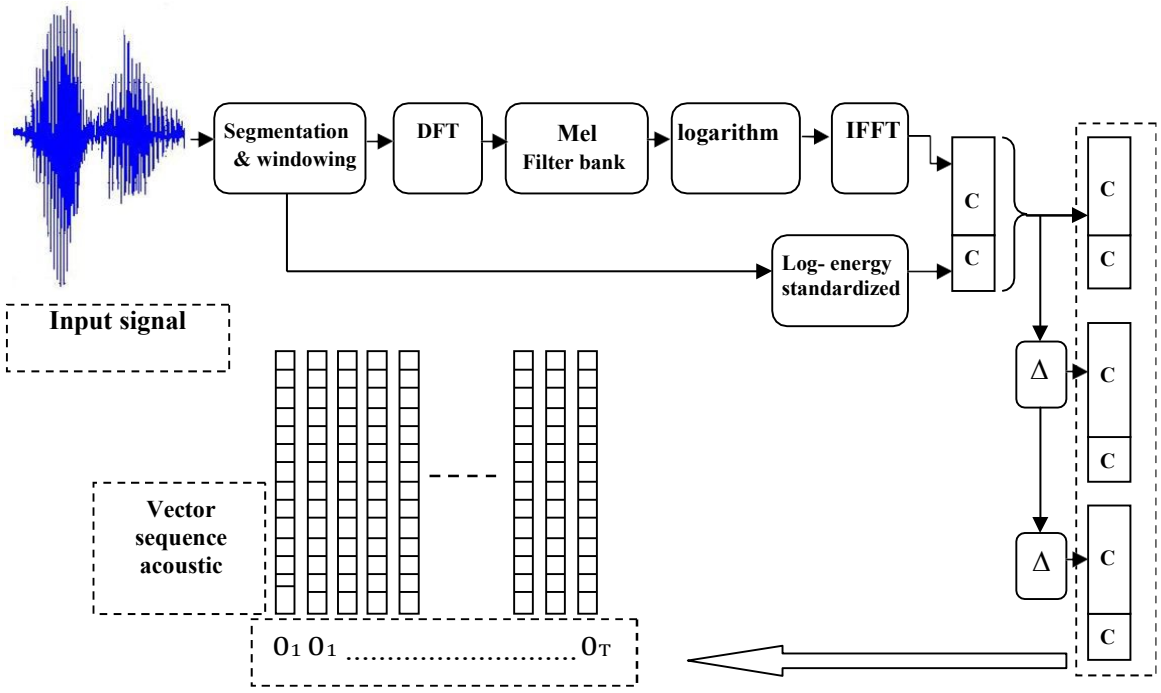

Figure III-1 The steps of an MFCC parameterization

III.2.2.1 **signal acquisition and pre-emphasis**

Acquiring the speech signal is the first step to take. It is digitizing an analog signal (speech) so that it is ready for digital processing subsequent. This step is generally carried out using a specialized acquisition card. Once picked up by a microphone, the signal is first filtered, then sampled and finally quantified. These successive operations make it possible to transform a continuous signal x (t) (where t denotes time) into a digital signal x (n) (where n corresponds to

discrete instants) [6].The format of the signal thus obtained must be further processed so that it can be used at recognition purposes. To do this, a pre-emphasis is first performed for detect high frequencies (Figure III-2) [6].Further analysis is then applied to the signal to separate the silence from the speech signal (Figure III-3), this analysis and automatic detection of speech, with the aim of reduce the signal to process it and cancel the part that represents silence. These two steps are illustrated bellow:

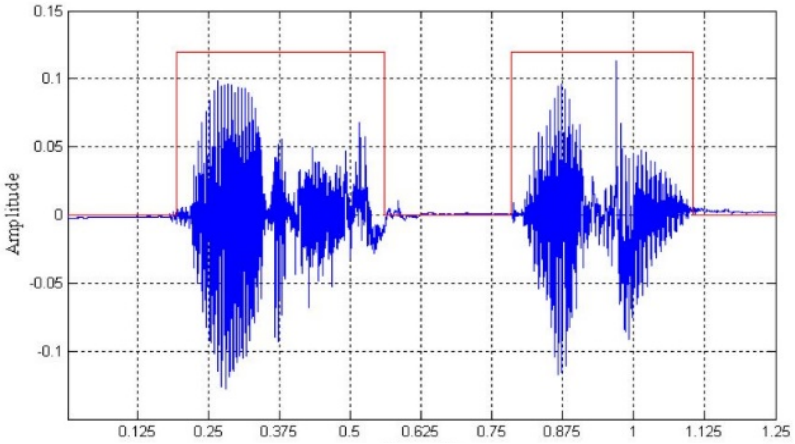

Figure III-2 detector applied to the phrase "forward"

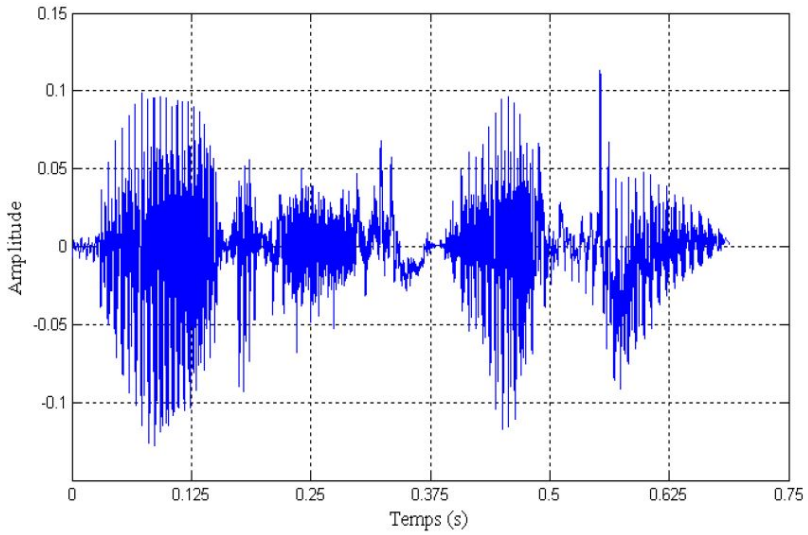

Figure III-3 the phrase "Start forward" after the silence has been removed

III.2.2.2 **segmentation and windowing**

the signal is segmented into frames where each frame consists of a fixed number (K) of speech samples. In general, (K) is fixed in such a way that each frame corresponds about 30 ms of speech. This segmentation is carried out using time windows sliding. The splitting of the signal into frames produces discontinuities at frame boundaries. To reduce these problems, weighting windows are applied (Figure III-4). These are functions that we apply to the set samples taken from the original signal window so as to reduce the effects of edge [6]:

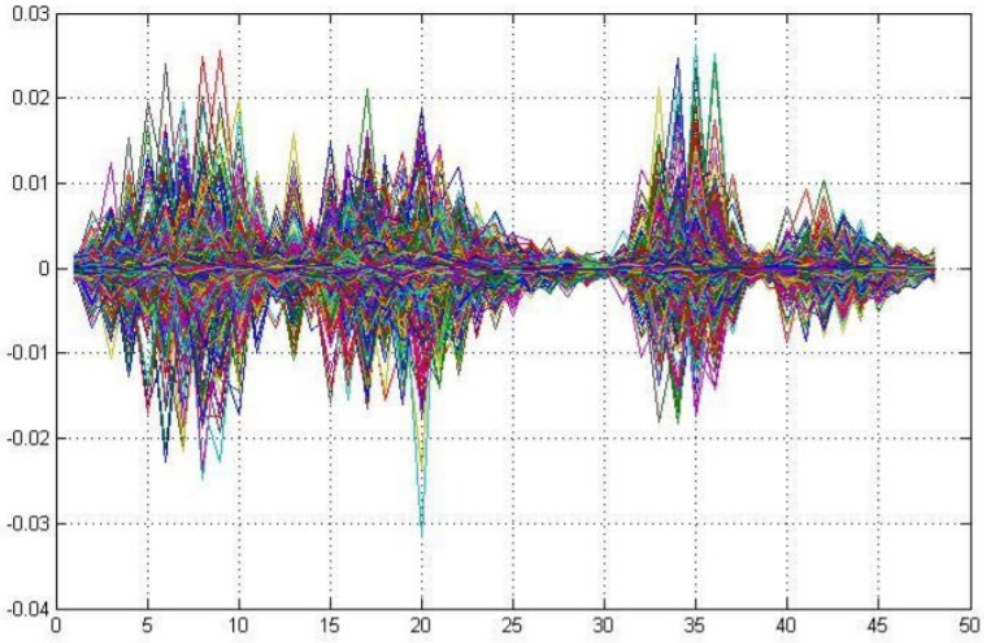

Figure III-4 signal representation of the phrase "forward" after segmentation and windowing.

III.2.2.3 **discrete Fourier transform and signal energy**

After this signal shaping (common to most methods for analyzing the speech), temporal or spectral analyzes can be applied to the signal. The energy of signal is the most intuitive parameter used to characterize the speech signal. It corresponds to the signal strength and is calculated directly in the time domain on a speech frame [6]. However, spectral analysis remains the most widely used means of characterizing the signal from word. It makes it possible to highlight certain phenomena characteristic of the production of the latter. Spectrograms have been used to represent speech from the 1940s using analog filters. Currently, the spectra are obtained numerically by a discrete Fourier transform (DFT: Discrete Fourier Transform), the fast Fourier transform algorithm

(FFT: Fast Fourier transform) is applied in this particular case. Witch allows us to go from the time domain to the frequency domain.

III.2.2.4 **the Mel Scale**

The number of spectral parameters calculated on a frame by the FFT remains too high for subsequent automatic processing. The energy of the spectrum is therefore calculated through a bench digital filters covering the bandwidth, which makes it possible to keep only one sub all of these parameters. Triangular filters are the most used. They are preferred for their simplicity and their smoothing effect on the spectrum. These filters are most often distributed on the Mel scale which is non-linear. (Figure III-5) The relation between the frequency in Hertz scale and its correspondence in mels is as follows [8]:

$$M_{mels} = x.log\left(1 + \frac{f_{Hz}}{y}\right)$$ *Equation 3_1*

Where fHz is the frequency, x = 2595 and y = 700

The outputs of the filter bank (Figure III-6) can be used directly as parameters of a recognition system [9]. However, in systems more recent, other coefficients derived from these outputs are most often used. these are the MFCC cepstral coefficients. These coefficients are linearly distributed They are considered more discriminating than the outputs of the filters and more robust to ambient noise.in addition, they are less correlated with each other [6].

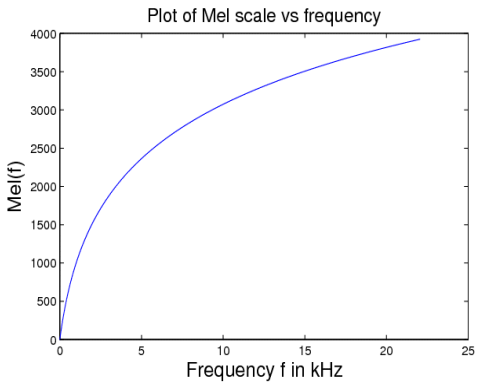

Figure III-5 Mel's scale

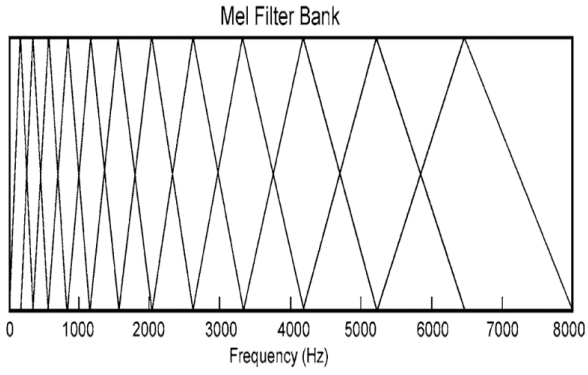

Figure III-6 A type of Mel scale filter bank using triangular filters

The spectrum of the signal is given by the inverse Fourier transform of the logarithm of the spectral density. A spectral representation has the advantage of dissociating, within the framework of the theory of a source-filter model, the glottic excitation and the resonances of the vocal tract. This separation is achieved by a homomorphism which transforms the convolution of signals in the time domain in an addition in the cepstral domain, the signal x (n) is considered as a time domain convolution of the excitation signal g (n) and the impulse response of the vocal tract h (n):

Several other types of parameterizations are present in the literature [10]. We can cite, among others, LPC coding (Linear Predictive Coding) and parametrization PLP (Perceptual Linear Predictive). In addition, we can find other techniques aiming to improve the quality of these parameters such as linear discriminant analysis (LDA: Linear Discriminant Analysis) and NLDA nonlinear discriminant analysis. More models' complexes for hearing and perception also exist and they have been studied in the field of speech.

III.2.3 Recognition

Automatic speech recognition, seen as a problem in the theory of communication, aims to reconstruct an unknown message from a sequence of observations (O) [11]. This amounts to finding, among all the possible messages, the one that in all likelihood, corresponds to the sequence of acoustic observations (O). The latter corresponds to a series of vectors making it possible to characterize the speech signal. The word(W) is a sequence of spoken words. It is therefore a question of finding the message the more probable knowing the sequence of acoustic observations(O). The probability P(W|O) is very difficult to determine, hence the need to decompose it. In using Bayes' rule, it is possible to reformulate the probability P(W|O) as follows:

$$P(W|O) = \frac{P(W)P(O|W)}{P(O)}$$ *Equation 3_2*

Since P (O) does not depend on W (the message), equation (3_2) will be equivalent to:

$$\widehat{W} = arg_W \, max \, P(W)P(O|W)$$ *Equation 3_3*

Thus, the recognition step consists in determining the sequence of words W which maximizes the product of the two terms P(W) and P(W|O). The first term represents the probability to observe the sequence of words W independently from the signal. This probability is determined by the language model. The second term indicates the probability of observing the sequence of acoustic vectors knowing a sequence of specific words (W). This probability is estimated by the acoustic model. The quality of such a speech recognition can be characterized by the accuracy and robustness of both models which make it possible to calculate these two terms P(W) and P(W|O). It is therefore a question of integrating the acoustic and linguistic levels in a single decision-making process allowing find the spoken message. (Figure III-7) shows the different steps required to recognize a message pronounced as an entry. First, the speech signal is subdivided into acoustic vectors. Using these vectors, the acoustic model is loaded, from the HMMs of phonemes learned on a learning corpus, to build the sequence of phoneme hypotheses of the signal pronounced. A single HMM model representing the hypothesis, will be built by the concatenation of a set of HMMs of phonemes. The sequence of words obtained will also be evaluated by the language model which makes it possible to estimate the probability P(W). In principle, this process is repeated for all possible hypotheses. The system finally gives the best N assumptions as a result of recognition.

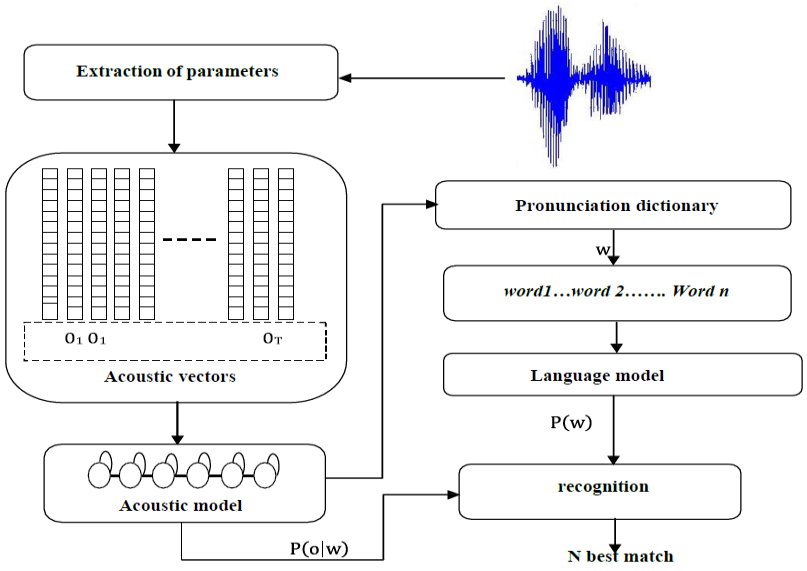

Figure III-7 The operating principle of the statistical approach for automatic speech recognition

In conclusion, speech recognition goes through three main stages. The first one concerns the acoustic modeling of the signal using HMMs. These are estimated from a learning corpus, their role is to model the phonemes in all their contexts. As a result, they must be able to evaluate the probability P(0|W). The second step is interested in language modeling. Its purpose is to estimate the probability P(W). To do this, statistical language models are most often used. These models seek to estimate the probabilities of appearance of words based on their historical. Finally, the third step uses search algorithms to deliver the (N) better solutions.

III.3 HMM-based acoustic modeling

Hidden Markov models are most commonly used in recognition automatic speech to estimate the probability P(O|W). A hidden Markov model is a stochastic automaton capable, after a learning phase, of estimating the probability, a sequence of observations has been generated by this model. The objective of the HMMs in speech recognition is to model the representative units of the speech signal. The most used units are the phonemes. However, there are also other types of units like allophones, syllables, triphones, words, etc. In this chapter, we are only interested in words.

In the following, we start by mathematically defining the hidden Markov models. Then we focus on the fundamental problems of HMMs. Then focus on the algorithmic details of the solutions provided. Finally, we show how to practice the theory of HMMs within the framework of a recognition application of the speech.

III.3.1 Definitions of hidden Markov models:

An HMM can be seen as a discrete set of states and the transitions between those states. Formally, it can be defined by the set of parameters λ [10]:

$$\lambda = (N, A, B, \pi)$$

Where: N is the number of nodes or states of the model.

$A = \{\alpha ij\} = \{p(qj|qi)\}$ is a matrix of size N x N. It contains the transition probabilities on all the states of the model. The probability of transition is the probability of choosing the transition αij to access the state qj starting from of the qi state. For an HMM of order 1, In other words, the evolution of the system between two instants t - 1 and t depends only of the state of this system at time t - 1 (order one) or of the two preceding instants t - 1 and t - 2 (order 2).

III.3.2 The fundamental problems of HMMs

Let λ be a hidden Markov model and O a sequence of acoustic observations. The recognition of this sequence is carried out by finding the model λ which maximizes the probability P (λ |O) (probability that a model λ generates a sequence of observations (O). Unfortunately, it is not possible to access this probability directly, but we can calculate the probability that a model given will generate a certain sequence of observations P (O| λ). Using Bayes' law, he is possible to link these two probabilities by:

$$P(\lambda|O) = \frac{P(\lambda)P(O|\lambda)}{P(O)}$$ *Equation 3_4*

(O| λ) Is the likelihood of the sequence of observations O given the model λ.

P (λ) is the prior probability of the model.

P (O) is the prior probability of the sequence of observations.

For a known sequence of observations: O= O1, O2…., OT, the probability P (O) can be considered constant, since it is independent of the model λ if the parameters of the latter are fixed. Thus, maximizing P (λ | O) amounts to maximizing P (O| λ) P (λ).

To do this, three fundamental problems must be solved:

III.3.2.1 **Evaluation:**

a sequence of observations Given: O= O1, O2…., OT, and the model λ = (N, A, B, π) how to calculate efficiently P (O| λ) the probability of observing the sequence O knowing the model λ?

III.3.2.2 **Decoding:**

a sequence of observations Given: O= O1, O2…., OT, and the model

λ = (N, A, B, π) how to choose the sequence of states: Q= q1, q2…., qT, which has the more chance to emit the sequence of observations O?

III.3.2.3 **Learning:**

How to determine the parameters of the model λ = (N, A, B, π) so to maximize P (O| λ)?

III.3.2.4 **Evaluation Problem**

Let the model λ = (N, A, B, π), O= O1, O2…., OT, be a sequence of observations and Q= q1, q2…., qT, a sequence of states. The probability of observing the sequence O for a sequence of states Q is [13]:

$$P(O|Q,\lambda) = b_{q1}(O_1).b_{q2}(O_2)\ldots b_{qT}(O_T)$$ *Equation 3_5*

The joint probability of the path Q and the observations (O) is:

$$P(O,Q|\lambda) = P(Q|\lambda).P(O|Q,\lambda)$$ *Equation 3_6*

The probability of the sequence of observation O knowing the model λ is obtained by the summation of P (O, Q| λ) over all possible sequences of states Q. So, the probability emission of observations is:

$$P(O|\lambda) = \sum_{Q} P(O,Q|\lambda)$$ *Equation 3_7*

$$P(O|\lambda) = \sum_{q_1,q_2,\ldots q_T} \pi_{q1} b_{q1}(o_1) a_{q1q2} b_{q2}(o_2) \quad \ldots . a_{qT-1qT}\, b_{qT}(O_T)$$ *Equation 3_8*

III.3.2.5 **Decoding problem:**

Given a sequence of observations O, and a model λ = (N, A, B, π) the decoding problem comes down to the search for an "optimal" sequence of states. This can be done with different. The

difficulty is in defining the optimal sequence of states. Therefore, we must choose a criterion among several optimality criteria. For example, a possible criterion to distribute the vectors of the sequence of observations on the states of the chain, consists in optimizing each state (qt) separately.

While these criteria are perfectly suited to certain applications, the most widely used criterion is the one who seeks the best overall sequence of states (the best path), that is to say who maximizes P (Q, O $|\lambda$). which amounts to maximizing P (Q $|$O, λ). A formal technique exists to calculate this optimal path is the Viterbi algorithm.

III.3.2.6 **Viterbi algorithm**

The Viterbi algorithm is an algorithm for analyzing a series of hidden states called the Viterbi path. This model is often used in the context of a Markov source, or a source where random variables present significant unknowns. Essentially, through logical means, the Viterbi algorithm looks at a set of objects according to certain properties, and tries to demonstrate how those properties could affect others(Figure III-8). This is often referred to as a Markov chain [18].

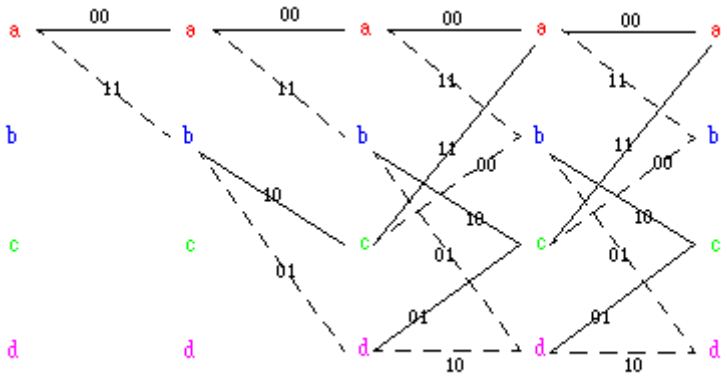

Figure III-8 the trellis representation of the encoder

The Viterbi algorithm is a maximum-likelihood decoding procedure for convolutional codes. We observe that (apart the first levels), there are two branches, which means two paths, entering each of the states. This means that a minimum distance decoder may make a decision at each level, choosing one of the two entering paths entering each state. The Viterbi algorithm computes a metric for every possible path, and chooses the one with the smallest metric. The paths that are

retained are called the survivors, and the maximum-likelihood path is always guaranteed to be among them. The algorithm performs step-by-step as follows:

- Initialization:

Set the metric of the left-most state of the trellis at 0.

- Computation step j+1:

We suppose that at the previous step we have identified all survivor paths and stored each state's survivor path. For each state at level j+1, we compute the metric of the incoming paths as the addition of the metric of the incoming branch and the metric of the survivor path. We then choose the path with the smallest metric for each state.

- Finalization:

We continue the computation until the algorithm reaches the termination node (the all-zero state), at which time it makes a decision on the maximum-likelihood path. Then the decoded sequence is the sequence of bits corresponding to this optimum path's branches [19].

III.3.2.7 **Learning problem**

The third problem is to find a method to adjust the parameters of the model $\lambda = (N, A, B, \pi)$ in order to maximize the probability of a sequence of observations given, knowing the model λ. This problem has no known analytical solution and there is no optimal technique for estimating model parameters. However, we can choose $\lambda = (N, A, B, \pi)$ such that $P(O|\lambda)$ is locally maximal using an iterative procedure such as the Baum-Welch method or the gradient technique [9].

In what follows we present an iterative procedure based on the technique of Baum-Welch. To describe how to re-estimate the parameters of the HMM, we define the probability $\varepsilon_t(i, j)$ which represents the probability of being in state i at time (t) and of making a transition to state j at time

(t +1) given the sequence of observations O and the model λ

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j|O,\lambda)$$ *Equation 3_9*

Baum-Welch algorithm estimates new parameters of the hidden Markov chain as following:

$$\bar{\pi}_i = \gamma_1(i), \qquad 1 \le i \le N$$
$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad 1 \le i \le N, 1 \le j \le N$$ *Equation 3_10*

$$\bar{b}_j(k) = \frac{\sum_{t=1,ot=k}^{T} \gamma_t(j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad 1 \le j \le N$$ *Equation 3_11*

The re-estimation of π_i is the probability of being in state i at time t = 1. The re-estimation formula of a_{ij} is the ratio of the number of transitions from state i to state j on the number of transitions starting from state i. The re-estimation of $b_i\ (k)$ is the ratio of the number of times to be in state i by observing k over the number of times being in state i. We have defined the current model λ = (N, A, B, π) and we used it for recalculate these variables, so we have the re-estimated model $\bar{\lambda} = (N, \bar{A}, \bar{B}, \bar{\pi})$. We can thus affirm one to the other of these propositions:

the initial model λ defines a critical point of the likelihood function, in this case $\lambda = \bar{\lambda}$

the model $\bar{\lambda}$ is better than the model λ in the sense that $P(O|\bar{\lambda}) > P(O|\lambda)$ therefore the sequence of observations O is more probable with the new model $\bar{\lambda}$.Based on this procedure, if we iteratively use the model $\bar{\lambda}$ instead of λ.and if we repeat the step of re-estimating the parameters. We can then improve the probability that O is observed knowing the model until reaching a certain limit point.

The final result of the re-estimation procedure is called: the estimation at the maximum of Likelihood of the HMM (Maximum Likelihood Estimation: MLE).

III.4 Continuous observation densities in hidden Markov models:

Until now we have only considered the case where the observations take values in a discrete finite alphabet and we could therefore use a discrete probability law in each state of the model. Such an approach is not compatible with observations which are continuous signals. Of course, quantifying the signal could solve the problem, but it could only lead to degradation. So, it is better to use hidden Markov models with continuous observation densities.

III.4.1 Application of HMMs to the recognition of single words

In speech recognition, n-order left-right Markov models are most often used due to the sequential aspect of the speech signal. (Figure III-9) illustrates an Example of a five-state HMM characterized by a distribution of probabilities for each state associated with an observation and by transition probabilities between states. HMM used for word modeling:

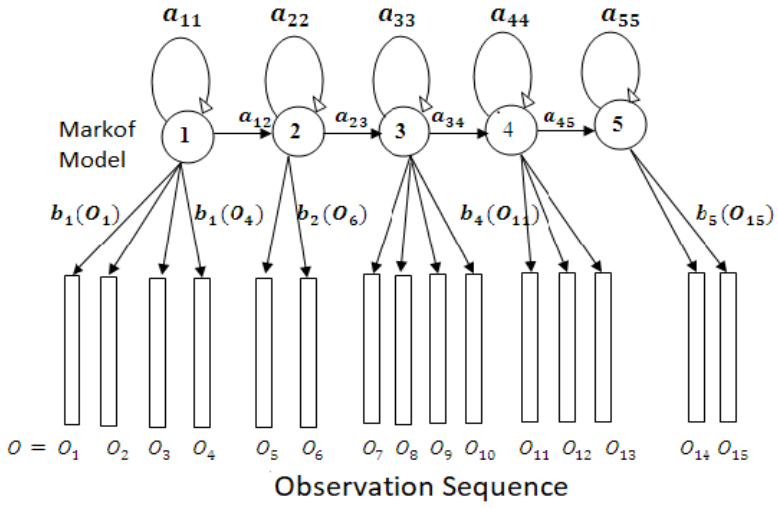

Figure III-9 Example of a five-state HMM

III.5 Application and Results:

For our work we used the HMM for the recognition of eight isolated words. So, we have a vocabulary of words (left, right, forward, backward, up, down, stop, start), each word is modeled by a distinct HMM, moreover, we have for each word of the vocabulary, a set of occurrences K (pronounced by 1 or more speakers), each occurrence of the word represents an observation. To realize the recognition of isolated words, the following steps are carried out:

For every word in the vocabulary, we had to build an HMM, we had to also estimate the Parameters of the model λ = (N, A, B, π) which optimize the likelihood instruction observation learning vectors for each word. For each unknown word the processing of Figure III-10)must be carried out, namely the measurement of observation sequence O = (O1, O2, …, OT), by an analysis of the characteristics of the speech, followed by the calculation of the model likelihood for all possible models, $P(O|\lambda)$ followed by the selection of the speech which has the highest model probability.

$$M^* = argmax[P(O|\lambda_m)] \; where \; 1 > m > M \qquad \textit{Equation 3\_12}$$

the probability calculation step is performed by the Viterbi algorithm (the maximum likelihood path is used).

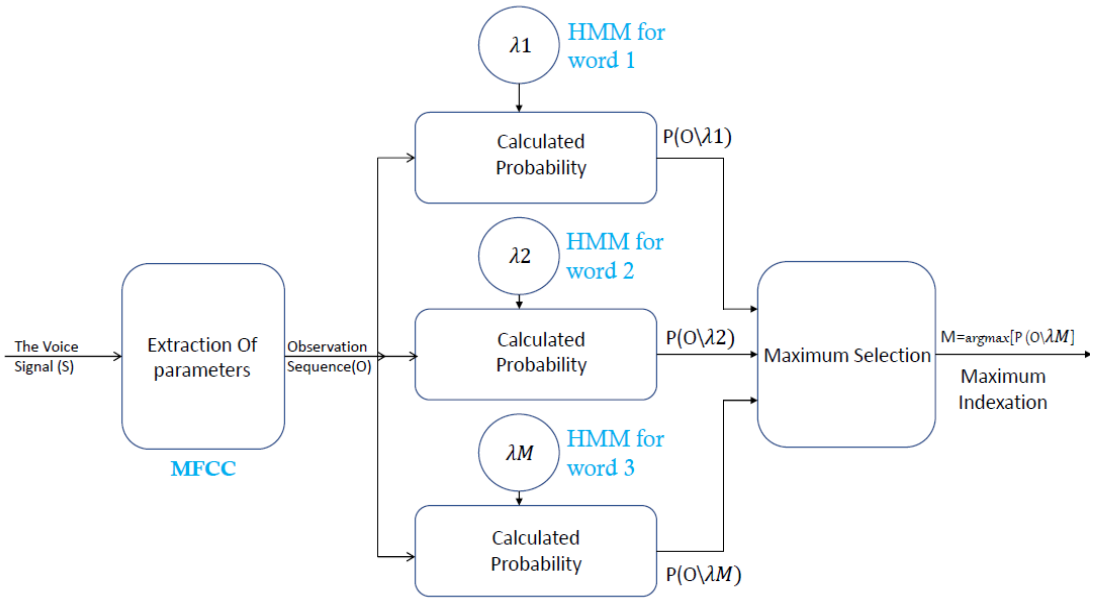

Figure III-10 isolated word recognition architecture

III.6 Conclusion

From the previous experiments and the results obtained we notice that the model of Hidden Markov ensures a good result in the field of speech recognition and in especially in the recognition of isolated words. The MFCC coefficients are behind these good results because they have a good representation of the behavior of the speech signal. It should be noted that poor results are obtained during the learning phase and recognition probably due to the quality of the sound recording or to poor pronunciations which led us to re-record the words which gave illogical results.

Chapter IV HTK – Hidden Markov Model Toolkit.

IV.1 Description

This tutorial was made in order to make easier the implementation of a sub word-based directions recognition system, for simple voice commanding applications. Moreover, after implementation finishes, the user may add new words for directions to the vocabulary by means of a modification to the pronouncing dictionary and task grammar. Our main purpose is to show the procedures through which it is possible to construct such system using HTK tools under Windows system. Therefore, we will explain in details all the steps until the recognizer gets ready to be tested, but we will not be concerned with the speech recognition theory involved. We let the reader to look for the theory basis to understand, complain and modify the obtained HMM models.

As an overview, we will start using phone models and then we will extend them to triphone models. So, the final HMMs obtained will be continuous density mixture Gaussian tied-state triphones. We used decision trees to cluster the states and then tie each cluster.

We will describe in detail the toolkit installation and the 11 steps to accomplish the speech recognition system

IV.2 Data Preparation

Initially, we have to record the data that will be used for training and the data that will be used for testing. But before recording, we have to define the training sentences and test sentences with good phonetic balance and coverage. So, in this section we will construct the task grammar and the dictionary that will be used to define such sentences. We let the user to choose to use TIMIT database or to construct his own database. Here, we will construct our database.

IV.2.1 Step 1 – The Task Grammar

A task grammar consists of a set of variable definitions followed by a regular expression containing the words to be recognized. Fortunately, HTK also provides a grammar definition language, and from that we can define our task grammar

So, make the file called “gram” containing the following lines

```
$directions = <RIGHT>|<LEFT>|<FORWARD>|<BACKWARD>|<UP>|<DOWN>;
$initialization = START;
$finalisation = STOP;
(SENT-START  $initialization UP <$directions $finalisation>    SENT-END)
```

Then execute the following command from DOS prompt

“HParse gram wdnet”

IV.2.2 Step 2 – The Dictionary

Now we have to make a dictionary containing the sorted words required in robot movement command. The dictionary for recognition task can be built using the British English BEEP pronouncing dictionary for general use but in our situation, we built our own dictionary “dict.txt” containing the word used for robot command.

Then Make a file called “wlist”, which must be a sorted list of the words to be recognized.

Copy the “dict.txt” dictionary for the C:\HTK321\ and type the command below

“HDman –e c:\htk –m –w wlist –n monophones1 –l dlog dict dict.txt” in the DOS prompt.

So, the dictionary “dict.txt” generated is as follows:

```
BACKWARD      b a k w e r d sp
DOWN          d a w n sp
FORWARD       f o r w e r d sp
LEFT          l e f t sp
RIGHT         r a y t sp
SENT-END   []   sil
SENT-START []   sil
START         s t a r t sp
STOP       s t o p sp
UP            a p sp
```

IV.2.3 Step 3 – Recording Data

The training will be recorded using HTK tools, if a database is not available. So, first we generate the sentences by executing the following command line:

"HSGen –n 130 –l wdnet dict > sentences.txt"

Then, for each sentence in the file "sentences.txt", record speech data using the following command:

"HSLab –n noname"

Note: we can record them using alternative recording software but the files recorded must be in ".wav" format.

IV.2.4 Step 4 – Creating the Transcription Files

Every file of training data must have an associated phone level transcription. Since we do not have a hand labeled data to initialize a set of models a flat-start scheme will be used instead. So, initially we will use a set that has no short-pause (sp) models between words. Then, once reasonable phone models have been generated, a short-pause (sp) model will be inserted between words to take in account any pauses introduced by the speaker

Now execute the following command line:

"Perl prompts2mlf sentences.mlf sentences.txt"

Next, make the file "mkphones0.led" containing the following lines:

```
1 EX
2 IS sil sil
3 DE sp
```

Execute the command line:

"HLEd –d dict –i phoneso.mlf mkphoneso.led sentences.mlf"

IV.2.5 Step 5 – Coding the Data

In this section we will show how to extract speech acoustics feature vectors from raw waveform data. HTK supports both FFT-based and LPC-based analysis, so we will Mel use Frequency Cepstral Coefficients (MFCC).To do so, we have to create a configuration file called "config", which contains the following lines with conversion parameters:

```
# Coding parameters
TARGETKIND = MFCC_0
TARGETRATE = 100000.0
SAVECOMPRESSED = T
WINDOWSIZE = 250000.0
USEHAMMING = T
PREEMCOEF = 0.97
NUMCHANS = 26
CEPLIFTER = 22
NUMCEPS = 12
ENORMALISE = F
```

Then create a script file (a file which has a list of files is referred to as script) called "codetr.scp" containing the lines below:

```
c:\HTK321\wav\1.wav c:\HTK321\mfc\1.mfc
c:\HTK321\wav\2.wav c:\HTK321\mfc\2.mfc
c:\HTK321\wav\3.wav c:\HTK321\mfc\3.mfc
c:\HTK321\wav\4.wav c:\HTK321\mfc\4.mfc
c:\HTK321\wav\5.wav c:\HTK321\mfc\5.mfc
c:\HTK321\wav\6.wav c:\HTK321\mfc\6.mfc
c:\HTK321\wav\7.wav c:\HTK321\mfc\7.mfc
c:\HTK321\wav\8.wav c:\HTK321\mfc\8.mfc
c:\HTK321\wav\9.wav c:\HTK321\mfc\9.mfc
c:\HTK321\wav\10.wav c:\HTK321\mfc\10.mfc
c:\HTK321\wav\11.wav c:\HTK321\mfc\11.mfc
c:\HTK321\wav\12.wav c:\HTK321\mfc\12.mfc
c:\HTK321\wav\13.wav c:\HTK321\mfc\13.mfc
c:\HTK321\wav\14.wav c:\HTK321\mfc\14.mfc
c:\HTK321\wav\15.wav c:\HTK321\mfc\15.mfc
c:\HTK321\wav\16.wav c:\HTK321\mfc\16.mfc
```

…etc.

Next, execute the command line:

"HCopy –T 1 –C config –S codetr.scp"

IV.3 Creating Monophone HMMs

In this section we will train a set of single-Gaussian monophone HMMs. We start with a set of identical monophone HMMs, in which the means and variances are identical, then we retrain these models, add short-pause (sp) and the silence models are slightly extended

IV.3.1 Step 6 – Creating Flat Start Monophones

First of all, in HMM training is to define a prototype model, whose parameters are not relevant, so its importance resides in the model topology definition. As we start with a phone-based system, an appropriate topology to use is 3-state left-right with no skips

To do so, create a file called "proto" which contains the following lines:

```
~o <VecSize> 39 <MFCC_0_D_A>
~h "proto"
<BeginHMM>
   <NumStates> 5
   <State> 2
      <Mean> 39
         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      <Variance> 39
         1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
   <State> 3
      <Mean> 39
         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      <Variance> 39
         1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
   <State> 4
      <Mean> 39
         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      <Variance> 39
         1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
   <TransP> 5
      0   1   0   0   0
      0  0.6 0.4  0   0
      0   0  0.6 0.4  0
      0   0   0  0.7 0.3
      0   0   0   0   0
<EndHMM>
```

Then, create the script file “train.scp” containing the paths of the training data, as described below:

```
c:\HTK321\mfc\1.mfc
c:\HTK321\mfc\2.mfc
c:\HTK321\mfc\3.mfc
c:\HTK321\mfc\4.mfc
c:\HTK321\mfc\5.mfc
c:\HTK321\mfc\6.mfc
c:\HTK321\mfc\7.mfc
c:\HTK321\mfc\8.mfc
c:\HTK321\mfc\9.mfc
c:\HTK321\mfc\10.mfc
c:\HTK321\mfc\11.mfc
c:\HTK321\mfc\12.mfc
c:\HTK321\mfc\13.mfc
```

…etc.

In the “config” file, change the line “TARGETKIND = MFCC_0” to “TARGETKIND = MFCC_0_D_A” and execute the command line below from DOS prompt.

“HCompV –C config –f 0.01 –m –S train2.scp –M hmm0 proto”

Then, add the following lines in the file “vFloors” and rename it to “macros”:

```
~o
   <VecSize> 39
   <MFCC_0_D_A>

```

so that the file “macros” will have the lines:

```
~o
   <VecSize> 39
   <MFCC_0_D_A>

~v varFloor1
<Variance> 39
 9.544637e-001 5.162001e-001 4.390759e-001 8.694764e-001 9.474930e-001 4.456218e-001 6.513027e-001 6.000700e-001 5.104576e-001
 3.447575e-001 2.820750e-001 2.441734e-001 3.155814e+000 1.720954e-002 1.477379e-002 1.460101e-002 1.951238e-002 2.394986e-002
 1.630496e-002 1.993453e-002 1.965995e-002 1.790006e-002 1.455124e-002 1.451952e-002 1.282476e-002 2.321478e-002 1.999643e-003
 2.037971e-003 2.203693e-003 2.812330e-003 3.271776e-003 2.682314e-003 3.144476e-003 3.220356e-003 3.021836e-003 2.577297e-003
 2.573429e-003 2.268731e-003 2.194394e-003
```

Next, for each phone in the file “monophones1”, copy the corresponding prototype in the file “proto”, including a monophone “sil”, and after that save the resulting file as “hmmdefs”,

In the last line of the file “monophones1”, include a model for the silence (sil), and delete the model for the short-pause (sp) and save this modified file as “monophones0”.

Create the directories “c:\>HTK321\hmm1”, “c:\>HTK321\hmm2” … “c:\>HTK321\hmm15”. Execute the command line:

“HERest –C config –I phones0.mlf –t 250 150 1000 –S train.scp –H hmm0/macros –H hmm0/hmmdefs –M hmm1 monophones0”

Then, for a parameter’s re-estimation execute the “HERest” command twice (each time HERest is ran, it performs a single parameter re-estimation):

“HERest –C config –I phones0.mlf –t 250 150 1000 –S train.scp –H hmm1/macros –H hmm1/hmmdefs –M hmm2 monophones0”

“HERest –C config –I phones0.mlf –t 250 150 1000 –S train.scp –H hmm2/macros –H hmm2/hmmdefs –M hmm3 monophones0”

IV.3.2 Step 7 – Fixing the Silence Models

Now we have created a 3 state left-to-right HMM for each phone and a HMM for the silence model “sil”. After that, we will add extra transitions from states 2 to 4 and from states 4 to 2. This makes the model more robust as it allows individual states to absorb the various impulsive noises in the training data (the backward skip impedes the model to transit to the following word). Moreover, we will create a 1 state short-pause (sp) model, which will have its emitting state tied to the center state of the silence model.

To accomplish this, first we created a file called “sil.hed” with the following lines:

```
AT 2 4 0.2 {sil.transP}
AT 4 2 0.2 {sil.transP}
AT 1 3 0.3 {sp.transP}
TI silst {sil.state[3],sp.state[2]}
```

Then, copy the files “hmmdefs” and “macros” from directory c:\>htk\hmm3 to directory c:\>htk\hmm4. Next, copy the central state of the “sil” model to generate the short-pause “sp” model.

After that, add the “sil” in the last line of the file “monophones1” then execute the following command lines:

“HHEd –H hmm4/macros –H hmm4/hmmdefs –M hmm5 sil.hed monophones1”

“HERest –C config –I phones0.mlf –t 250 150 1000 –S train.scp –H hmm5/macros –H hmm5/hmmdefs –M hmm6 monophones1”

“HERest –C config –I phones0.mlf –t 250 150 1000 –S train.scp –H hmm6/macros –H hmm6/hmmdefs –M hmm7 monophones1”

IV.3.3 Step 8 – Realigning the Training Data

The dictionary created in Step 2 has words with multiple pronunciations. So, the phone models created so far can be used to realign the training data and to create new transcriptions.

To do this, firstly add the line “silence sil” between the lines “**SENT-START [] sil**” and “**START s t a r t sp**” in the file “dict.txt” as illustrated below:

```
BACKWARD      b a k w e r d sp
DOWN          d a w n sp
FORWARD       f o r w e r d sp
LEFT          l e f t sp
RIGHT         r a y t sp
SENT-END   []    sil
SENT-START []    sil
silence          sil
START         s t a r t sp
STOP       s t o p sp
UP            a p sp
```

After that, execute the command line:

"HVite –l '*' –o SWT –b silence –C config –a –H hmm7/macros –H hmm7/hmmdefs –i aligned.mlf –m –t 250 –y lab –I sentencas.mlf -S train.scp dict monophones1"

Depending on the database quality, some data may be discarded during alignment. So, the missed data in the file "aligned.mlf" can be substituted manually by the corresponding non-aligned data contained in the file "phones0.mlf".

Execute the command lines:

"HERest –C config –I aligned.mlf –t 250 150 1000 –S train.scp –H hmm7/macros –H hmm7/hmmdefs –M hmm8 monophones1"

"HERest –C config –I aligned.mlf –t 250 150 1000 –S train.scp –H hmm8/macros –H hmm8/hmmdefs –M hmm9 monophones1"

IV.4 Creating Tied-State Triphones

Now we have to create monophone HMMs and we want to obtain context-dependent triphone HMMs from them. This is achieved in two steps:

Firstly, we have to convert the monophone transcriptions to triphone transcriptions. Then, a set of triphone models are created by copying the monphone models and re-estimating.

Secondly, states of the triphones are tied to ensure that all state distribution can be robustly estimated.

IV.4.1 Step 9 – Making Triphones from Monophones

Context-dependent triphones can be made by cloning monophones and then re-estimating using triphones transcriptions

So, create the file "mktri.led" which has the following lines:

```
1 WB sp
2 WB sil
3 TC
```

Then execute the command line:

"HLEd –n triphones1 –l '*' –i wintri.mlf mktri.led aligned.mlf"

Execute the sequence of command lines:

"perl maketrihed monophones1 triphones1"

"HHEd –B –H hmm9/macros –H hmm9/hmmdefs –M hmm10 mktri.hed monophones1"

"HERest –C config –I wintri.mlf –t 250 150 1000 –s stats –S train.scp –H hmm10/macros –H hmm10/hmmdefs –M hmm11 triphones1"

"HERest –C config –I wintri.mlf –t 250 150 1000 –s stats –S train.scp –H hmm11/macros –H hmm11/hmmdefs –M hmm12 triphones1"

IV.4.2 Step 10 – Making Tied-State Triphones

Now we have a set of triphone HMMs with all triphones sharing the same transition matrix. So, many of these models' variances in the output distribution will have been floored due to the lack of that associated with many of the states. From now we have to tie states within triphone sets in order to share data and be able to make robust parameter estimation.

Execute the command line from DOS prompt:

"perl mkclscript.prl TB 350 monophones1 > tree.hed"

The file "*tree.hed*" will contains questions about the left and right context of each triphone, and these question will be used during the process of clustering states and tying each cluster.

Make the file "*fullist*" containing the monofones and the triphones together so the file will contain:

```
b
a
k
w
e
r
d
sp
n
f
o
l
t
y
sil
s
p
s+t
s-t+a
t-a+r
a-r+t
r-t
a+p
a-p
r+a
```

etc...

Execute the sequence of command lines:

"HHEd –H hmm12/macros –H hmm12/hmmdefs –M hmm13 tree.hed triphones1 > log"

"HERest –C config –I wintri.mlf –t 250 150 1000 –s stats –S train.scp –H hmm13/macros –H hmm13/hmmdefs –M hmm14 tiedlist"

"HERest –C config –I wintri.mlf –t 250 150 1000 –s stats –S train.scp –H hmm14/macros –H hmm14/hmmdefs –M hmm15 tiedlist"

IV.5 Recognizer Evaluation

Now we can evaluate the performance of our recognizer. In order to do so, first we will prepare the testing data. As we did not have the acoustic data for a testing set, we used the training data only to verify if the procedures developed in this tutorial are correct. It is important to note that the recorded data quality was not observed, changes in the number of states used to model the monophones and triphones may be different from the number adopted, changes in the pruning parameters also require an adjustment and so on. But these subjects are not supposed to be discussed here.

IV.5.1 Step 11 – Recognizing the Test Data

Firstly, create a script file called "*codetst.scp*" containing the lines below:

```
1  C:\HTK321\tst\wav\1.wav     C:\HTK321\tst\mfc\1.mfc
2  C:\HTK321\tst\wav\2.wav     C:\HTK321\tst\mfc\2.mfc
3  C:\HTK321\tst\wav\3.wav     C:\HTK321\tst\mfc\3.mfc
4  C:\HTK321\tst\wav\4.wav     C:\HTK321\tst\mfc\4.mfc
5  C:\HTK321\tst\wav\5.wav     C:\HTK321\tst\mfc\5.mfc
6  C:\HTK321\tst\wav\6.wav     C:\HTK321\tst\mfc\6.mfc
7  C:\HTK321\tst\wav\7.wav     C:\HTK321\tst\mfc\7.mfc
8  C:\HTK321\tst\wav\8.wav     C:\HTK321\tst\mfc\8.mfc
9  C:\HTK321\tst\wav\9.wav     C:\HTK321\tst\mfc\9.mfc
10 C:\HTK321\tst\wav\10.wav    C:\HTK321\tst\mfc\10.mfc
11 C:\HTK321\tst\wav\11.wav    C:\HTK321\tst\mfc\11.mfc
12 C:\HTK321\tst\wav\12.wav    C:\HTK321\tst\mfc\12.mfc
13 C:\HTK321\tst\wav\13.wav    C:\HTK321\tst\mfc\13.mfc
14 C:\HTK321\tst\wav\14.wav    C:\HTK321\tst\mfc\14.mfc
15 C:\HTK321\tst\wav\15.wav    C:\HTK321\tst\mfc\15.mfc
16 C:\HTK321\tst\wav\16.wav    C:\HTK321\tst\mfc\16.mfc
17 C:\HTK321\tst\wav\17.wav    C:\HTK321\tst\mfc\17.mfc
```

After that, execute the command line:

"HCopy –T 1 –C config –S codetst.scp"

Then make the file called "*test.scp*" contains:

```
C:\HTK321\tst\mfc\1.mfc
C:\HTK321\tst\mfc\2.mfc
C:\HTK321\tst\mfc\3.mfc
C:\HTK321\tst\mfc\4.mfc
C:\HTK321\tst\mfc\5.mfc
C:\HTK321\tst\mfc\6.mfc
C:\HTK321\tst\mfc\7.mfc
C:\HTK321\tst\mfc\8.mfc
C:\HTK321\tst\mfc\9.mfc
C:\HTK321\tst\mfc\10.mfc
C:\HTK321\tst\mfc\11.mfc
```

Execute the following command line:

"HVite –H hmm15/macros –H hmm15/hmmdefs –S test.scp –l '*' –i recout.mlf –w wdnet –p 0 –s 5 dict tiedlist"

Add two parameters (**FORCECXTEXP** and **ALLOWXWRDEXP**) at the end of the "*config*" file, as described below

```
# Coding parameters
TARGETKIND = MFCC_0_D_A
TARGETRATE = 100000.0
SAVECOMPRESSED = T
WINDOWSIZE = 250000.0
USEHAMMING = T
PREEMCOEF = 0.97
NUMCHANS = 26
CEPLIFTER = 22
NUMCEPS = 12
ENORMALISE = F
FORCECXTEXP = T
ALLOWXWRDEXP = F
```

Make a copy of the file "*sentencas.mlf*" and save it as "*testref.mlf*"

As a final analysis, execute the command line:

"HResults –I testref.mlf tiedlist recout.mlf"

Then we have the recognizer performance, as we can observe below

```
====================== HTK Results Analysis =======================
  Date: Tue Jun 29 23:32:40 2021
  Ref : testref.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=2.00 [H=2, S=98, N=100]
WORD: %Corr=82.93, Acc=62.85 [H=1730, D=0, S=356, I=419, N=2086]
====================================================================
```

IV.6 Conclusion

As a conclusion we find that our Model we built Recognizes the words with a good results, so we say that This Model is used For recognizing Isolated Words and we can use it to command and control a robot arm or a machine using speech commands.

General Conclusion.

As a general conclusion we say that after these steps we have been able to build a system that can command and control a robot or a machine using only voice commands, and the system can be used and implemented in real life. For a future perspective we can develop that system by adding more languages to the dictionary vocabulary (French, Arabic, ...) in order to make it more useful for foreign users. We can also add a specific distance of moving so the movements will be more precise and accurate.

References:

[1] Gilles Léothaud, "Théorie de la Phonation"2004-2005;

[2] Thomas HUEBER, "Reconstitution de la parole par imagerie ultrasonore et vidéo de l'appareil vocal: vers une communication parlée silencieuse", doctorate thesis: Pierre et Marie Curie 2009.

[3] Deller, J.R., Hansen, J.H. L., Proakis J. G, "Discrete Time Processing of Speech Signals" IEEE Press 1999.

[4] Calliope, "La parole et son traitement automatique" Paris, Masson 1989.

[5] René Boite, Hervé Bourlard "Traitement de la parole" presses polytechniques et Universitaires de Romandes 2000

[6] Salma Jamoussi "Methodes statistiques pour la comprehension automatique de la parole" thèse Doctorat de l'universite Henri Poincare-Nancy decembre 2004

[7] https://www.svtaudiology.com/the-auditory-system (seen on: may, 15th ,2021).

[8] E. Zwicker et R. Feldtkeller. Psychoacoustique , "L'oreille, récepteur d'information", Masson, 1981.

[9] L. R. Bahl, P. F. Brown, P. V. de Souza, R. L. Mercer, et M. A. Picheny, "Acoustic Markov models used in the Tangora speech recognition system", In Proceedings of the International Conference on Acoustics, Speech and Signal Processing, 1988.

[10] L. Rabiner et B. H. Juang, "A tutorial on hidden Markov models and selected application in speech recognition", Proceeding of IEEE, 1989.

[11] F. Jelinek, "Continuous speech recognition by statistical methods", Proceeding of IEEE,1976.

[12] J. P. Haton, J. M. Pierrel, G. Perennou, J. Caelen, et J. L. Gauvain, "Reconnaissance automatique de la Parole" dunod, 1991.

[13] Yassine BEN AYED, "Détection de mots clés dans un flux de parole ", Doctorat de l'Ecole Nationale Supérieure des Télécommunications décembre 2003

[14] J. L. Gauvain et C. Lee, "Maximum a-posteriori estimation for multivariate gaussian mixture observations of Markov chains", IEEE Transactions on Speech and Audio Processing,1994.

[15] R. Cardin, Y. Normandin, et R. De Mori, " High performance connected digit recognition using maximal mutual information estimation", In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 1991.

[16] S. Kapadia, V. Valtchev, et S. J. Young, "MMI training for continuous phoneme recognition on the TIMIT database", In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, , 1993.

[17] R. Boite. Traitement de la parole. Collection Electricité. Presses Polytechniques et Universitaires Romandes, 2000.

[18] M . Kunt and R. Boite. Traitement de la parole. Presses Polytechniques Romandes, press polytechnique romandes edition, 1987.

[18] https://www.techopedia.com (seen on: may, 23rd ,2021).

[19] http://www.wirelesscommunication.nl/ (seen on: may, 23rd ,2021).