Use of Long-Term Average Spectrum for Automatic Speaker Recognition

Sergey Pauk

22.12.2006

University of Joensuu
Department of Computer Science
Master's Thesis

Abstract

In this thesis the long-term average spectrum (LTAS) performance in task of text-independent closed-set automatic speaker identification (ASI) is studied. LTAS is a statistical feature that any digital signal possesses. It answers the general question about the average signal power distribution over frequency band.

The use of LTAS for ASI can be promising for large enough speech samples since this feature collects the average general information about the speaker. In this work we firstly study the separate LTAS performance and try to find the optimal parameters tuning for the ASI task. Afterwards, the scores fusion of two features, MFCC and LTAS, is studied in case of noiseless and noisy speech samples.

Keywords: text-independent closed-set speaker identification, long-term average spectrum, speaker features scores fusion.

Acknowledgements

I would like to thank my supervisor Pasi Fränti for giving me the opportunity to write this thesis, for his help and comments during my work. I'm also very grateful to Ville Hautamäki for his guidance, comments and support in many problems that appeared during this work.

Contents

Chapter 1

Introduction

1.1 Automatic Speaker Recognition (ASR)

It is well-known that voices of different people in general do not sound similar to the listener. In our everyday life we are performing speaker recognition when we are talking by phone, listen to some mass-media or just hear some familiar voice. This is one of the most important speech features to be *speaker-dependent* that allows us to recognize the voices of the persons we know. In general, even if we do not know the speaking person we still can get some speaker-dependent information like gender, age, emotional state, physical defects, accent and others.

The ability to recognize speakers by the example of their speech is referred as *speaker recognition* [ATA76]. However, nowadays with the fast development of computers and computer science the speaker recognition is not limited only by human listening. The process of the person recognition from his voice by computer is called *automatic speaker recognition* (ASR).

The ASR is a part of more general task: person recognition by some of his biometrics. In spite of the fact that there are more precise types of biometrics (e.g. fingerprints, retina, DNA) the human voice biometric has some advantages: it is easy to get an example of speech and it can be performed almost immediately, the voice can be transferred as an electric signal over telephone or some other type of connection.

These features of human speech have defined the possible application for ASR. Human voice as a biometric is often used to control the access to services and information or in any other kind of security systems. The most common applications for ASR are [REY02, CAM97]:

- access control to physical devices (doors)
- transaction authentication in banking and e-commerce
- criminals monitoring
- perform speaker labeling in speech data management and storing (voice mail, intelligent answering machines)

- setup personal settings fro some multiuser device (condition in the car, various settings of light, radio and others in the house)
- determining of speaker belonging to some type (age, gender) for advertisement purposes or for better automatic selection of services

In this way, nowadays there are various ASR applications due to the influence of computer on different aspects of human life.

1.2 Speaker Recognition Structure

Speaker recognition problem is usually divided into two parts: ***speaker verification*** and ***speaker identification*** [SHA86, ATA76, CAM97]. The former task is to verify a claimed identity of a person from a sample of his speech. Here the decision is binary and should answer if the speaking person is the same man as it claimed. Speaker identification is a more challenging task when the system should identify whom the voice belongs to or decides that this person is unknown.

Automatic speaker identification (ASI) task can further be divided into ***closed-set*** and ***opened-set*** problems. In the first case person should be identified from the stored models and not allowed to be unknown. Thus, the random choice for the verification will have success in 50% cases but for ASI only $100/N$% for the closed-set problem [SHA86]. Errors are of two types: ***false accept*** errors when wrong person is selected in speaker identification or imposter is accepted for speaker verification, and ***false reject errors*** when the true speaker is rejected.

Depending on application and recognition algorithm the problem can be divided into ***text-dependent*** recognition and ***text-independent*** recognition. In first case the system knows the text that recognized speaker is saying. In text-independent case the system does not know the content of the speech and even does not know the spoken language. In this work ASI text-independent closed-set model is considered.

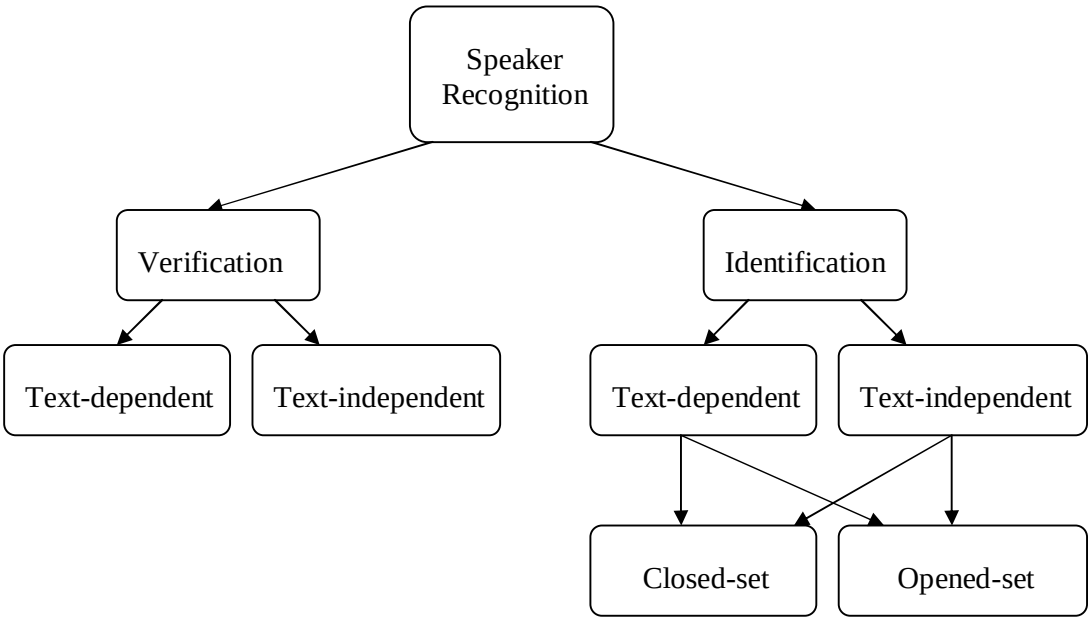

Figure 1.1. Speaker recognition subproblems.

Since the speaker identification problem is a kind of pattern recognition problem it has the two main phases. Firstly, at training phase the speech samples are collected from speakers and models are recorded into the dictionary also known as speaker database. This is not a real-time task and is performed relatively rarely.

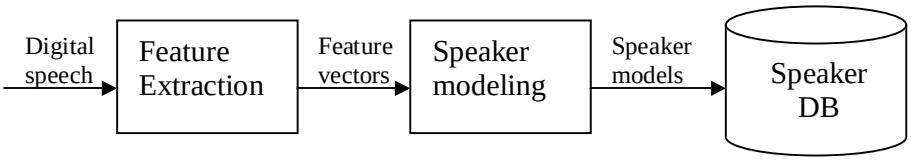

Figure 1.2. Training phase for ASI.

The second phase is automatic recognition when a new speaker should be identified. ASI's recognition phase starts with the new speaker's feature extraction. After that ***measure of similarity*** or ***distance*** between unknown speaker and stored models is calculated. The model that has the best score (closer distance) is chosen to identify the speaker in case of closed-set ASI.

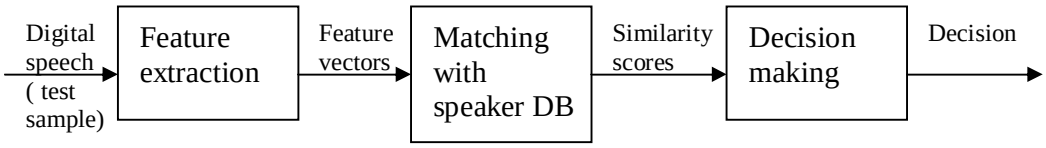

Figure 1.2. Recognition phase for ASI.

In most applications this is a real-time task and, therefore, the speed of new sample feature extraction and distance calculation is crucial.

1.3 Thesis Description

Long-term average spectrum (LTAS) is one of the basic statistical features for the arbitrary digital signal. The main goal of this study was to analyze the performance of LTAS in the task of text-independent closed-set ASI. Firstly, we intend to find optimal set of parameters for the identification task when LTAS is performing as a separate feature. Afterwards, the performance of fusion of LTAS with another speech feature, MFCC, is considered to be studied in case of noiseless and noisy speech signals.

The thesis is organized as follows. In Chapter 2 we study the physics of speech and the speech production mechanism. In Chapter 3 we describe the basics of digital signal processing starting from analog-to-digital conversion with further overview of the main DSP tool that is discrete Fourier transform. Chapter 4 firstly gives the idea of speech feature extraction in general and afterwards describes two speech features in detail: LTAS and MFCC. Chapter 5 is devoted to experiments. In that part of the thesis we study optimal LTAS tuning and fusion with MFCC performance for the task of ASI. Finally, in Chapter 6 the summary of the work is presented and conclusions are given.

Chapter 2

Speech Basics

2.1 Sound and Speech

Sound is an acoustic pressure wave; such wave is constructed by the dilution and compressions of molecules of the matter in which the wave is diffusing. The direction of these oscillations equals to stream of wave energy's application. The pressure of molecules is more than average in compressed zones due to application of energy, molecules density is less than average in dilution. These oscillations of acoustic pressure are represented as a sine wave and shown in Figure 2.1.

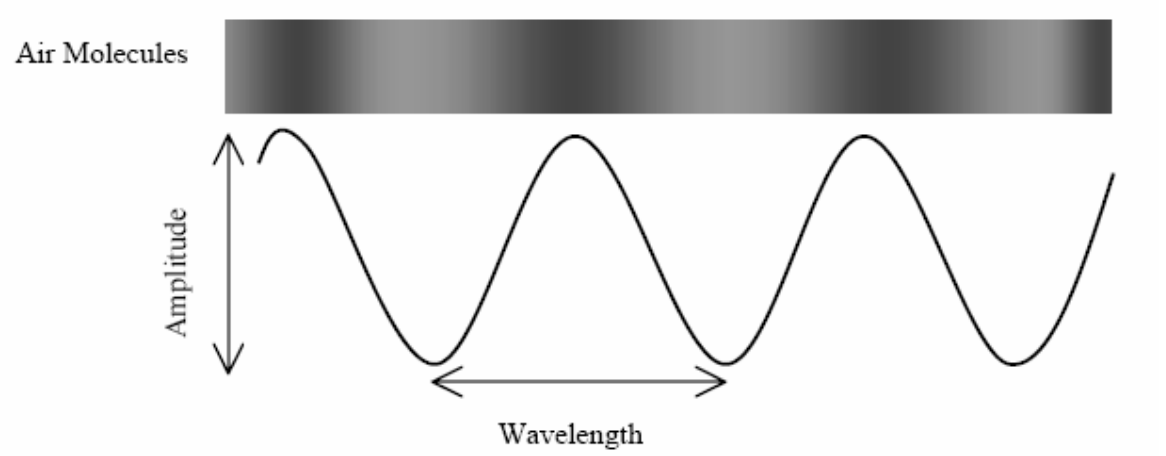

Figure 2.1. Application of sound energy causes alternating compression/refraction of air molecules, described by a sine wave. There are two important parameters, amplitude and wavelength, to describe a sine wave (after [HAH01]).

The use of the sine graph is only for notational convenience in Figure 2.1. Charting of local pressure variations does not form a transverse wave over time since sound and the air atomic particles are just oscillating in place along the direction line of energy's application. Sound pressure wave has the speed in air which can be calculated approximately as $331{,}5 + 0.6T_c m/s$, where T_c is the Celsius temperature.

The amount of movements of molecules from their standstill position reflects the amount of work that has to be done to set all the matter molecules into movement. This quantity of displacement is called ***amplitude*** of sound, which is shown in Figure 2.1. Sound amplitude has the wide range, there for the logarithmic scale in ***decibels*** (dB) is convenient measure. Actually, a decibel scale is a mean for comparing any two sounds:

$$10 \cdot \log_{10}(P_1 / P_2) \tag{2.1}$$

where P_1 and P_2 are two levels of sound power.

Sound pressure level (SPL) is the measure of absolute sound pressure P in dB:

$$SPL(dB) = 10 \cdot \log_{10}\left(\frac{P}{P_0}\right) \tag{2.2}$$

where $P_0 = 0{,}0002\ mbar$ for a tone of $1KHz$ is the threshold of hearing and it corresponds to the SPL value, which equals to zero.

Different air pressure generates different sounds for our hearing. In other words, different sounds have unique waveforms. When the guitar string is vibrating, the air pressure distribution forms the similar patterns are repeated over the time. Such kind of a sound is called *periodic* and human can hear it as a pitch. Let us consider another example: we can produce a sound by squeezing a sheet of paper. There is no periodic sound structure of acoustic wave in this situation, hence human hearing have not a feeling of a pitch.

An acoustic waveform for a word “stamp” is represented on Figure 2.2, and it is divided into separate phonetic units [HC99]. We can easily see from this figure that various phonetic units produce various repeating patterns of waveforms: between [t] and [p], the vocal tract is closure for these sounds and waveforms makes a straight line, which indicates here is near silence (such line detects minimal sound output). We can detect repeating patterns of air pressure in the structure of vowel and following [m] based on feature discussed before. In other words, waveforms [æ] and [m] are periodic sounds.

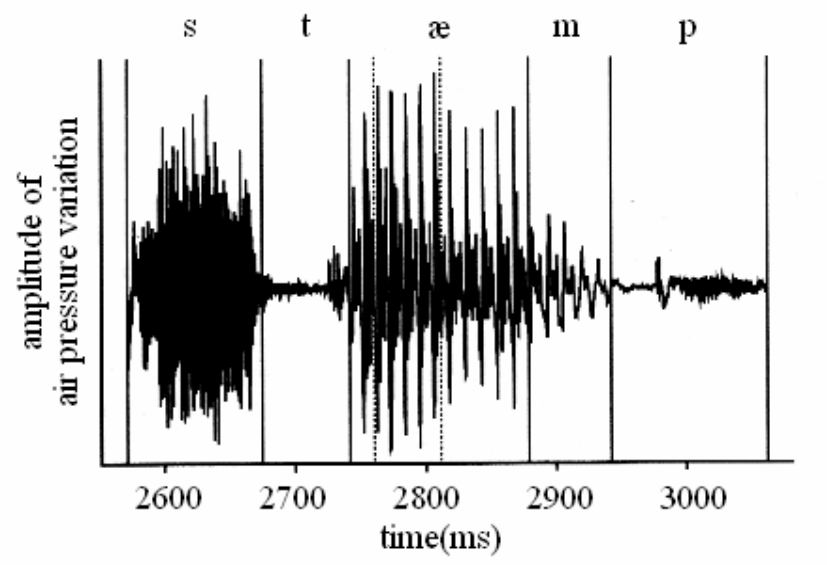

Figure 2.2. Waveform for the word “stamp”. Here we see division on elementary phonetic units with respective waveforms (after [HC99]).

The more detailed illustration of vowel [æ] periodic structure is illustrated in Figure 2.3. Here are four very similar periodic patterns. Each of this pattern forms *pitch period.* The wave is called *periodic*, if it is generated by some amount of pitch periods. Absolutely identical pitch periods do not exist in real life; all of them are little bit various from each other. Nevertheless, in speech technology sounds with such slightly different patterns are considered as periodic.

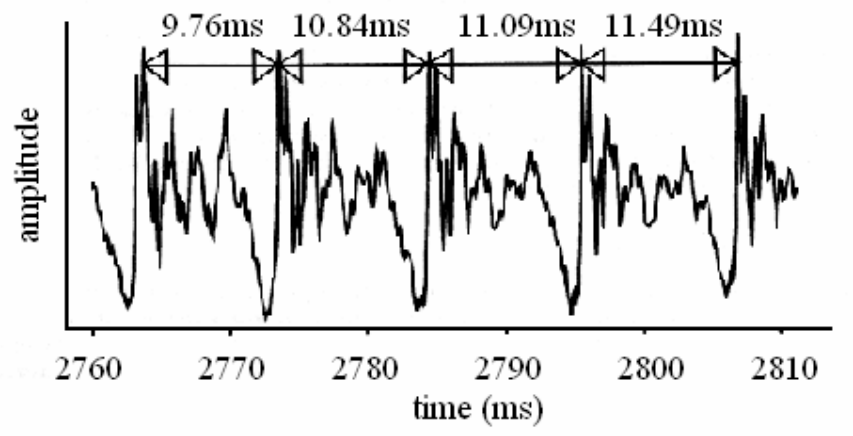

Figure 2.3. Repeating waveform for phonetic unit [æ] (after [HC99]).

For periodic sounds each pitch period is corresponding to one full period of vocal fold vibration. Hence, the duration of a cycle of vocal fold vibration can be estimated by the time

of one pitch period; in acoustics this vocal fold vibration interval is known as ***fundamental frequency*** (F_0).

$$F_0 = 1/T_{pitch} \tag{2.3}$$

where T_{pitch} is duration of a pitch period. When fundamental frequency value is increased, we hear a rising pitch while in the opposite case we hear a lowering pitch. Since in Figure 2.3 the pitch period length is increasing and according to (2.3) we can conclude that pitch is decreasing.

2.2 Speech Production

Human speech conveys three types of information: the linguistic information that represents the meaning of idea that speaker wants to say; individual information about speaker that allows to make difference between speakers; emotional information that is describing emotions of speaker [FUR01]. None of these types is the most important since the importance of information depends on the problem we have to solve. For example, in task of speech recognition the first type contains almost all information we need, but in case of speaker identification more valuable are second and third types.

The concept of information transferring from one person to another is basis of human speech production. Moving of thought to listener individual produces set of neurological processes and transferring of muscular construction of the sound wave, which is diffusing in the space, and the listener can obtain and analyze it. At first step, speaker forms the desired idea in his mind, after that human modifies this thought into linguistic form that is a representation of source idea through the words and sentences. Finally, at the last step speaker supplements individual speech features like pitch intonation to emphasize most important parts of target idea.

Human body has various internal structures, which produce acoustic sound pressure wave as person speech wave. Let us consider these structures in detail. The upper human torso with representation from the right side is illustrated in Figure 2.4. Main structures of human body are shown in this illustration; such human's parts are responsible for sound and speech production [DHP00].

The medical terms of vocal tract parts are: *lungs*, *larynx* (structure of voice output), *trachea*(windpipe), *oral cavity*(mouth), *pharyngeal cavity*(throat), and *nasal cavity*(nose).

Sometimes in other sources another terms are used; oral and pharyngeal cavities can be grouped into one entity and called as *vocal tract*. The nasal cavity (or *nasal tract*) starts at the velum and have the end point at the nostrils (part of nose). S*oft palate* or *velum, vocal folds* or *vocal cords, teeth* and *lips*, tongue form other organs, which impact the speech creation. We can see the velum soft tip (it is called *uvula* also), when the mouth is wide open and the oral cavity hangs down. Group of these components has scientific name *articulators* in the theory of speech processing. The *mandible* or *jaw* performs the control function of the size and figure of vocal tract, it produces the handling of positions for other articulators, there for this structure is also articulator.

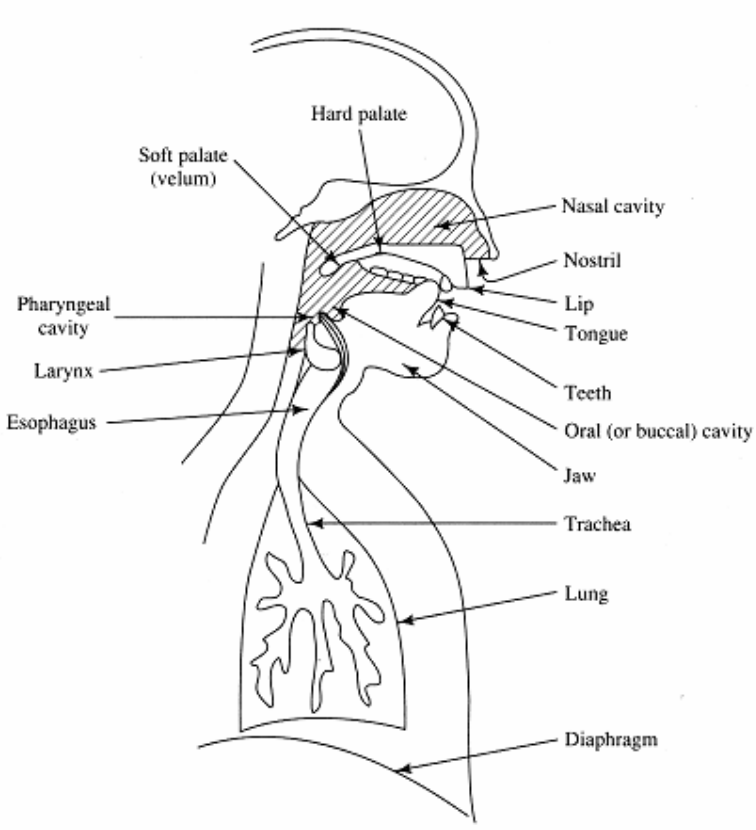

Figure 2.4. Schematic diagram of the human vocal mechanism (after [DHP00]).

A schematic illustration of the entire anatomical and physiological structure of speech creation is represented in Figure 2.5 [FLA72]. The sound creation process starts at the lungs and corresponding muscles and are the source of pressured air. The air is pushed out from lungs to the bronchi and trachea by the muscle force. The next step of speech sound creating is forced vocal cord when air stream is going through them.

After the vocal cords relaxation the air flow can perform further into two ways. It can become turbulent due to passing through the vocal tract system, or it can create an air pressure behind the closure point in the vocal tract and as a result of this closure opening air flow can transform into brief pulse sound.

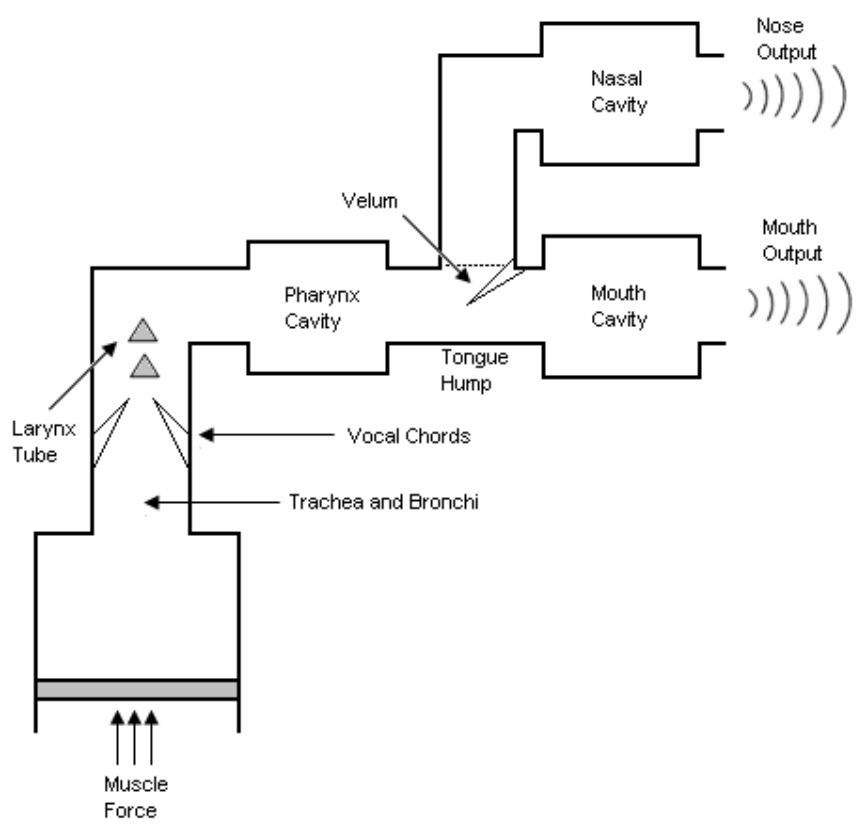

Figure 2.5. Block diagram of human speech production apparatus.

Speech is produced as the sequence of sounds. Therefore the shape, position and size of various articulators and the state of vocal cords changes over time to reflect the sound being produced.

2.3 Phonemes and Voicing

Let us consider that a speaker has already created and formulated the idea of what he want to say. The next step is to code this thought with the help of some dictionary of different sound units and produce words and sentences that consist of these units. Such elementary units that allow the speech to carry the linguistic information are referred as *phonemes*. In American English, for example, about 42 phonemes are known by linguists. They are usually divided into four general groups which are: vowels, semivowels, consonants and diphthongs. Each phoneme is produced by unique combination of *articulatory gestures*. Further, articulatory

gestures are defined by position and changes of the vocal tract articulators, and by the properties and way of air excitation.

Theoretically, the phoneme can be thought about as an ideal sound unit with unique waveform that has a bijective mapping into different combinations of articulatory gestures. However, in real life there are some additional factors that are to be taken into consideration. For different persons these ideal phonemes will be affected by different genders, ages, accents, etc. In this way, in spite of the fact that each phoneme will be correctly recognized by the listener in most cases it will have slightly different waveforms from person to person and also from time to time for one speaker. We see here, that from acoustical point of view one phoneme will represent not one real sound but a whole class of similar sounds that will be referred as one phoneme. However, despite the different ways of phoneme pronunciation, the sense will be perceived the same by most of listeners. Finally it can be summarized that the set of language phonemes is that minimal set of elementary units that are necessary and sufficient to express any possible thought by use of this language.

For further discussion we have to distinguish *phonemes* and *phones*, *phonetics* and *phonemics* [DHP00]. Previously we have defined phonemes while the sounds produced by speaker are referred in speech technology as *phones*. Similarly, the study of abstract speech units and their behavior and relations inside the language is known as phonemics, while phonetics study the actual language itself.

The basic classification of phonemes is done by detecting either they are *voiced* or *voiceless* [HAH01]. Voiced phonemes group consists of vowels and some consonants. Phonemes in this group have a structure of waveform close to periodical one. To the contrary, voiceless sounds, for example consonants “f” or “s”, do not posses any regularities. Another feature that is different between these groups is the average signal energy: for voiced phoneme it has in general bigger value compared to voiceless one as shown in Figure 2.6. In this figure we the waveform of the word *“sees”* is shown. This word consists of 3 phonemes: an unvoiced consonant [*s*], a vowel [iy] and, a voiced consonant [z] [HC99].

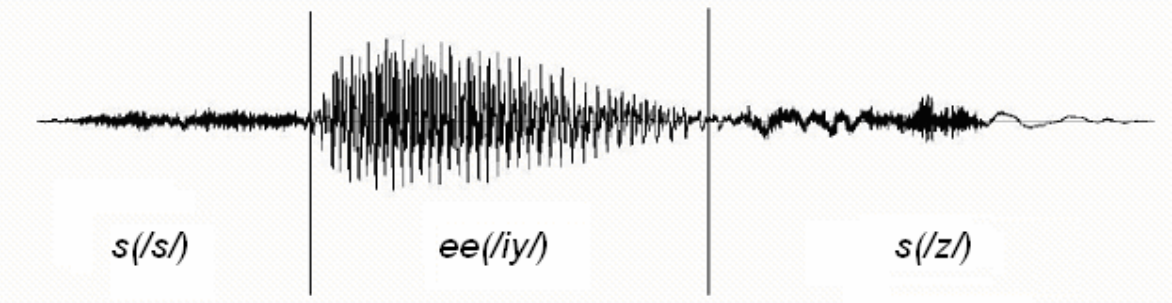

Figure 2.6. Waveform of “*sees*”, showing a voiceless phoneme [s], followed by a voiced vowel [iy]. The final sound [z] is a type of voiced consonant (after [HC99]).

Let us discuss the nature of this difference between unvoiced and voiced phonemes. During the time of the vocal folds vibration that happens in case of phoneme articulation it considered to be voiced, otherwise it is voiceless. As it was mentioned earlier, vowels belong to the voiced group and moreover they stay voiced during all their length. The various vowel *timbres* are produced with help of lips and tongue when they are forming the shape of the oral resonance cavity in different manner. The pitch range varies from person to person due to the different vocal folds vibration frequency. For a large man the lower bound is about 60 periods per second while for a child the upper bound reaches the value of 300 vibrations per second or higher. As we have defined earlier the frequency of the vocal folds opening and closing inside the larynx in the process of the voiced phonemes producing is referred by speech scientists as fundamental frequency. It plays a big role for the human perception of a sound pitch, i.e. distinguishing the higher and lower tones.

2.4 Speech Representation in Time and Frequency Domains

We can say that a speech signal is a slowly varying signal in the sense that if we will take an experiment over quite short period of time (from 5 to 100*msec*), its parameters are almost stationary. However, when taking in consideration longer time interval (more than 0.2 *sec*), the signal characteristics reflect the different sounds that are spoken.

Although speech waveforms can provide a lot of information to a speech scientist, like information about fundamental frequency or voiced property of a sound, for many other methods of speech analysis the frequency representation of speech is used. This frequency domain can be treated in the same way that amplitude domain to distinguish difference between sounds. Formerly, we have discussed *amplitude*(air pressure power)-*time* representation, but with use of frequency we can construct 2-dimentional *amplitude-frequency* and *frequency-time* representations or even 3-dimentional *amplitude-frequency-time representation* [RJ93].

En example of various types of speech representation can be seen at Figure 2.7. Here first graph is an amplitude-time presentation of a sentence “Every salt breeze comes from the sea.” Second one is 3-dimensional representation amplitude-frequency-time representation, also referred as *spectrogram*, where the amplitude strength is shown as darkness of point (higher is the amplitude the more is darkness). Third picture represents the phrase in the amplitude-frequency domain like a histogram.

There is one other method to represent time-varying parameters of speech via calculation of spectral activity based on the model of speech production. The human vocal tract can be modeled as a tube, or a system of tubes of different sizes, connected to each other. Acoustic theory tells us that the transfer of energy of air flow from the beginning of the tube to the end can be described in terms of resonates or natural frequencies of this tube. These natural frequencies are referred as *formants* in speech technology and represent the frequencies that carry the most part of the energy of air flow. There exists a good correspondence between formants and the points of high energy in spectrogram representation of speech.

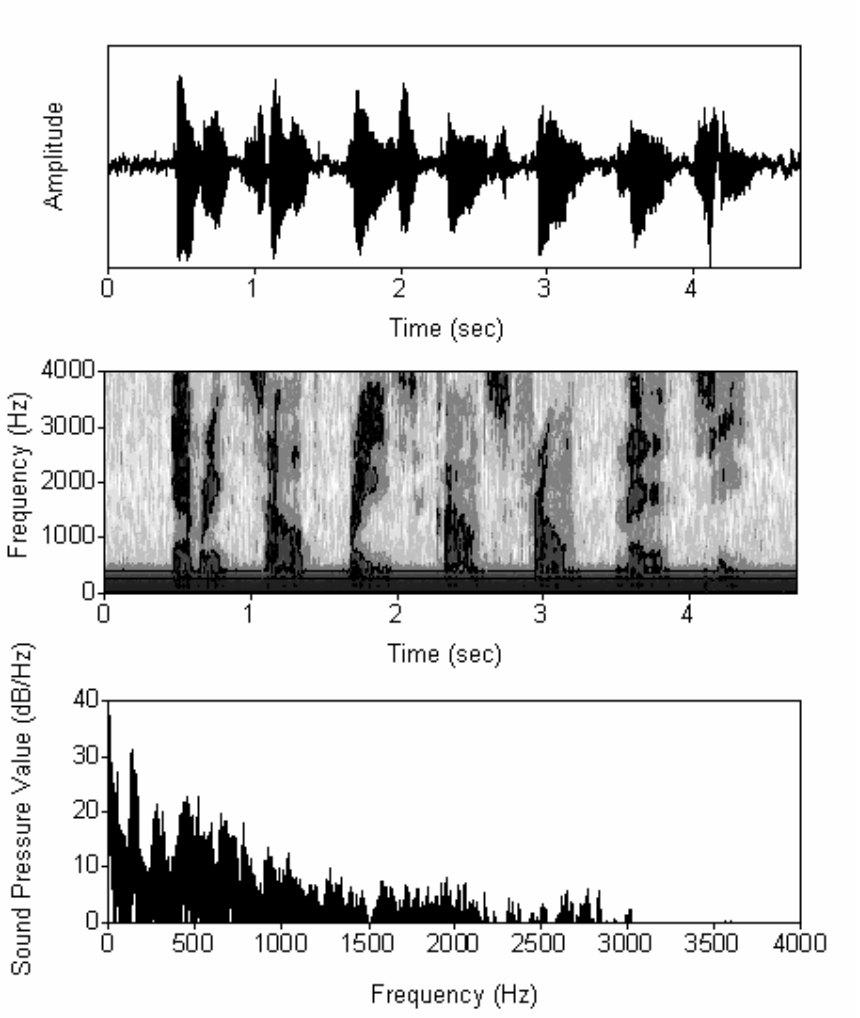

Figure 2.7. Speech amplitude, spectrogram and frequency histogram for a sentence "Each salt breeze comes from the sea."

Chapter 3

Digital Signal Processing Concepts

3.1 Introduction

By signal we mean here the dependence of one parameter from another that carries some type of information that can be conveyed, displayed and manipulated. Usually the first of these parameters is called dependant and represents quantitative characteristics of signal with respect to second parameter changing. The latter is called independent parameter (often time is taken to be an independent parameter).

According to its abbreviation, Digital Signal Processing(DSP) is an area of computer science that studies how to work with digital representation of signals. The use of DSP is to analyze, modify, extract and compare information about signals. The most of signals are analog by their nature, in other words, dependant parameter is changing continuously with respect to changing independent one. For example, sound waves are in nature analog signals, usually presented as dependence of physical quantity of signal from time. The signals that are used in the most popular DSP models are produced from analog signals that were sampled at regular intervals and were converted into a digital representation.

There exist some reasons to convert a signal to a digital form; for example, to remove interference or noise, to study the spectrum of the signal or to transform a signal into a more suitable representation. There many reasons to use DSP while handling signals [IJ02]:

- *Guaranteed accuracy*. The desired accuracy can be reached by using sufficient amount of memory.
- *Perfect reproducibility*. Identical performance from unit to unit is obtained since there are no variances due to the component tolerances. In this way, any pattern of signal can be copied or reproduced any number of time without any quality degradation (or loss of useful information).
- *No distortion* with change of temperature or the age of record.
- *Processing flexibility*. DSP systems can be programmed, reused and modified without any change in hardware.

- *Superior Performance.* DSP allows actions and methods that are even not possible to apply to analog signal. For example, complex adaptive filtering
- *Digital Devices.* At the present moment there a lot of digital devices used in science and life, hence DSP is the only approach that can be applied to deal with data from this type of devices.

However, we have to mention here the disadvantages of DSP also. The most significant are:

- *Speed and cost.* Modern ADC (analog-to-digital) and DAC (digital-to-analog) converters often do not have enough resolution for wide bandwidth applications. Wideband systems are still processed by analog methods.
- *Finite wordlength problem.* In a lot of real-life situations DSP solution can use only finite number of bits by some economic or other reasons. Hence, the signal can suffer a serious quality degradation.

Never the less, DSP is one of the fastest growing fields of computer science and electronics and is applied in many areas representing information in digital form and handling it by digital processors.

3.2 Sampling

The speech signal, being an acoustic wave, can be transformed to an electrical wave by use of microphone. After that an analog electrical wave is transformed into digital signal by analog-to-digital conversion (ADC), referred as *digitalization.* ADC consists of three processes that are *sampling*, *quantizing* and *coding*. The first process samples continuous analog signal at regular intervals, obtaining finite number of independent parameter values. Quantization task is to crate an approximation for original wave form by use of the sampled values. Coding assigns an actual number to each sample; usually binary coding is applied.

In a sampling process the continuous analog signal $x(t)$ is converted into finite sequence $\{x_k\}$, where each sample x_k is taken at time t_k as shown in Figure 3.1:

$$t_k = kT \qquad k = 0,1,... \tag{3.1}$$

Here T is called a *sampling period*. The inverse value $f = 1/T$ is called *sampling frequency.*

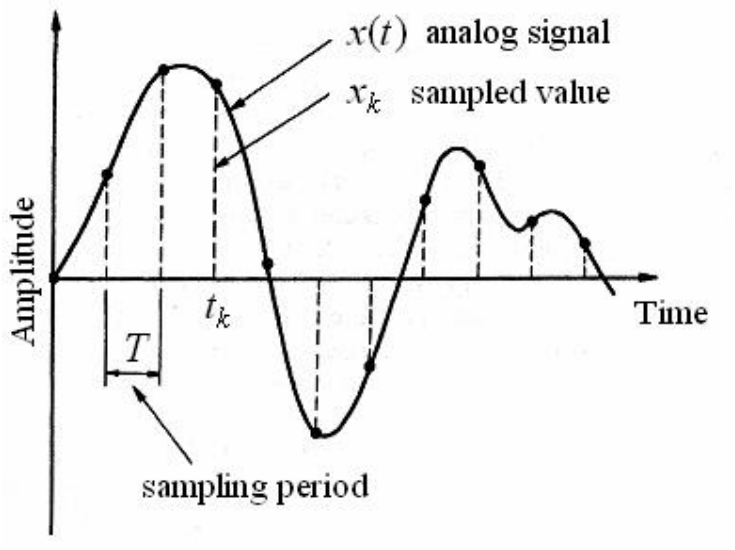

Figure 3.1. Sampling in the time domain (after [FUR01]).

It's easy to see, that if sampling period will be taken too large then the digital signal cannot represent analog signal with desired quality. From the other side, if sampling period will be taken too small, then some extra memory will be used to store redundant quantity of samples and digital processors will spend more time to handle some useless data.

Solution to this problem is proposed by Shannon-Someya sampling theorem[SW49]. This theorem allows knowing the bandwidth of initial analog signal to define the optimal sampling period. Sampling theorem says that if all frequencies of signal $x(t)$ lies in the interval $[0,W]$ and sampling period is taken as $T = \frac{1}{2W}(\text{sec})$, then the original analog signal can be completely represented by:

$$x(t) = \sum_{k=-\infty}^{\infty} x_k \frac{\sin(2pW(t-k/2W))}{2pW(t-k/2W)} \tag{3.2}$$

where samples x_k, according to (3.1), are taken at times $t_k = kT = \frac{k}{2W}$. Sampling frequency that is corresponding to such defined sampling period

$$f_N = 1/T = 2W \tag{3.3}$$

is referred by DSP scientists as *Nyquist rat*e.

3.3 Quantization

The quantization procedure divides the entire continuous amplitude range into subranges. Afterward, the waves with amplitudes from the same subrange are assigned the same new amplitude value. There are two quantization parameters: *quantization step size* Δ defines the length of one subrange and *number of levels* N defines how many different discrete values amplitude can take. When for each amplitude value there are allocated M bits, it is logically to set the number of levels equal to $N = 2^M$ since such scheme allows to use memory resources in a most efficient way [FUR01]. Further, Δ and N should be properly selected to cover exactly the interval of all possible amplitude values of particular signal. Let's define $x_{\max} = \max(|x_k|)$, then to cover the whole range of amplitudes we should set:

$$2x_{\max} = \Delta \cdot 2^M \tag{3.4}$$

The difference between the initial value x_k and quantized value $\hat{x}_k$ is called *quantization error* or *quantization noise* $e_k = x_k - \hat{x}_k$. According to (3.3) and the latter definition, quantization error satisfies following constraints:

$$-\frac{\Delta}{2} \le e_k \le \frac{\Delta}{2}, \quad for\ \forall\ k \tag{3.5}$$

A statistical model that takes into consideration three following properties can be assumed to serve as the quantization noise [RS75].

- Quantization noise is a stationary white noise.
- It is uncorrelated with the initial signal
- Distribution of quantization noise is uniform over the interval described in (3.4) and is equal to zero in other points.

$$Probability(e_k) = \begin{cases} 1/\Delta & e_k \in [-\Delta/2, \Delta/2] \\ 0 & otherwise \end{cases} \tag{3.6}$$

The parameter of *signal-to-noise ratio* (SNR) is one of the most important characteristics in DSP. For quantization noise it is defined as follows:

$$SNR = \frac{s_x^2}{s_e^2} = \frac{E[x_k^2]}{E[e_k^2]} = \frac{\sum_{k:x_k \in [-\Delta/2,\,\Delta/2]} x_k^2}{\sum_{k:e_k \in [-\Delta/2,\,\Delta/2]} e_k^2} \tag{3.7}$$

When the assumptions mentioned in (3.4) are satisfied we obtain

$$s_e^2 = \frac{\Delta^2}{12} = \frac{1}{12}\left(\frac{2x_{\max}}{2^M}\right)^2 = \frac{x_{\max}^2}{3 \cdot 2^{2M}} \tag{3.8}$$

and SNR can be transformed to the following expression:

$$SNR = 3\frac{2^{2M}}{(x_{\max}/s_x)^2} \tag{3.9}$$

Representing SNR in dB scale according to (3.8) we obtain:

$$SNR(dB) = 10\log_{10}\left(\frac{s_x^2}{s_e^2}\right) = 6M + 4.77 - 20\log_{10}\frac{x_{\max}}{s_x}\,[dB] \tag{3.10}$$

Finally, if we will set the maximum absolute value of amplitude $x_{\max} = 4s_x$, expression (3.10) can be simplified to:

$$SNR = 6M - 7.2\ dB \tag{3.11}$$

3.4 Fourier Transform Basics

The core mechanism of many DSP processors is transformation from time domain to a frequency domain and inverse one. These two representations provide complementary data about the same initial signal. The best known Fourier Transform is the Discrete Fourier Transform (DFT) and a class of fast algorithms for the computation of DFT that are generally called Fast Fourier Transform (FFT) [CT65, DV90] . Reasons to use DFT are that it allows to represent adequately all but the shortest of signal lengths (< 1*s*) in frequency domain, that the separate components are sinusoidal and they are not distorted when transmitted through linear systems, that the truncated Fourier frequency components give a more precise representation

of the data than any other exponential series, and finally, that there are the FFT algorithms allowing to process data at a high speed. We have to notice one more reason of why Fourier Transform is so popular in various DSP tasks; the first publication about it appeared in 1822 by Fourier [FOU03] and hence Fourier analysis has achieved high level of development and familiarity in modern science.

Any periodic signal (or wave) can be represented as an infinite sum of sine and cosine waves with various frequencies and amplitudes in the following way [IJ02]:

$$x(t) = a_0 + \sum_{n=-\infty}^{+\infty} a_n \cos(nwt) + \sum_{n=-\infty}^{+\infty} b_n \sin(nwt) \tag{3.12}$$

This representation is referred as Fourier series. Here t is the independent parameter (usually time), $x(t)$ is a dependant parameter (usually voltage), $w = 2p / T_p$ is called *first harmonic*, or *fundamental angular frequency* that is related with *fundamental frequency* in the following way: $f = w / 2p$. T_p is the period of waveform repetition.

Coefficients of Fourier series are calculated as shown in (3.13, 3.14):

$$a_0 = \frac{1}{T_p} \int_{-T_p/2}^{T_p/2} x(t)dt \tag{3.13}$$

is a constant term that has a physical meaning of average voltage of signal over the whole signal time.

$$\begin{aligned} a_n &= \frac{2}{T_p} \int_{-T_p/2}^{T_p/2} x(t)\cos(nwt)dt \\ b_n &= \frac{2}{T_p} \int_{-T_p/2}^{T_p/2} x(t)\sin(nwt)dt \end{aligned} \tag{3.14}$$

While w is known as first harmonic, a frequency nw is respectively called n^{th} *harmonic*. Hence, any periodic wave can be presented via enumerable amount of cosine and sine waves with frequencies that are multiples of fundamental angular frequency.

This Fourier series can be written in other form with use of complex analysis (3.15) and this representation is more easy to operate with mathematically.

$$x(t) = \sum_{n=-\infty}^{\infty} d_n e^{jnw\,t} \tag{3.15}$$

where Fourier coefficients are computed as:

$$d_n = \frac{1}{T_p} \int_{-T_p/2}^{T_p/2} x(t) e^{-inw\,t} dt \tag{3.16}$$

Here d_n are complex values, but their modules $|d_n|$ have physical sense of voltage.

The summation index goes from minus infinity to infinity; hence half of frequencies are negative. In this way, these values have no physical sense, but are just mathematical objects. The former and latter representations of Fourier series are connected to each other by the following relationships:

$$\begin{aligned} |d_n| &= (a_n^2 + b_n^2)^{1/2} \\ f_n &= -\frac{1}{\tan(b_n / a_n)} \end{aligned} \tag{3.17}$$

where f_n is the *phase angle* of the nth harmonic. Therefore, each harmonic component of series is absolutely defined by these two characteristics.

Unfortunately, most of the waves in real life are not periodic, hence we can not apply Fourier series to such type of signals, but the modification of latter can be applied to arbitrary waveform. Let's increase the period of repetition of some periodic signal to infinity; as T_p increases the distance between to neighboring harmonics becomes smaller, hence, in limiting case $1/T_p = w/2p$ is transformed to $dw/2p$. Therefore, the discrete infinite number of harmonics $\{nw\}$ goes to continuous variable w, amplitude and phase spectrum also becomes continuous [BCF73]. Considering that

$$d_n \xrightarrow[T_p \to \infty]{} d(w) \tag{3.18}$$

we obtain characteristic d of signal in the integral form:

$$d(w) = \frac{dw}{2p} \int_{-\infty}^{\infty} x(t) e^{-iw\,t}\, dt \tag{3.19}$$

Sometimes this formula is presented in other form:

$$F(iw) = \frac{d(w)}{dw / 2p} = \int_{-\infty}^{\infty} x(t) e^{-iw\,t}\, dt \tag{3.20}$$

where the complex function $F(iw)$ is called the *Fourier integral* or *Fourier transform.* The module of this value:

$$\left|F(iw)\right| = \frac{F(iw)}{e^{if(w)}} \tag{3.21}$$

is measured in [V/Hz] and shows the spectral distribution of voltage. Here phase angle is calculated as:

$$f(w) = \tan^{-1}\left(\frac{\mathrm{Im}[F(iw)]}{\mathrm{Re}[F(iw)]}\right) \tag{3.22}$$

The value $\left|F(iw)\right|^2$ has the units of [V^2/Hz^2] and has the physical sense of *energy spectral density.* The are under plot of $\left|F(iw)\right|$ versus $f = 2p\,/\,w$ in the range $[\bar{f} - df, \bar{f} + df]$ gives the value of mean voltage at frequency $\bar{f}$, and, in the same way, the area under the plot of energy spectral density in this interval is equal to the mean energy at frequency $\bar{f}$.

Finally, we have obtained the transformation of arbitrary signal from the time domain to the frequency representation. The inverse of this, that is referred as *inverse Fourier transform*, allows us to return from frequency to tome domain:

$$x(t) = \frac{1}{2p} \int_{-\infty}^{\infty} F(iw) e^{iw\,t}\, dw = \int_{-\infty}^{\infty} F(iw) e^{iw\,t}\, df \tag{3.23}$$

3.5 Discrete Fourier Transform (DFT)

As it was noticed in Sections 3.1, many practical applications the signal is converted from analog to digital form. Analog signal is continuous and we can not represent the values of all points. Therefore, the signal is sampled, as it was described in Section 3.2 and after that is quantized as it was shown in Section 3.3. The necessary sampling rate that is called Nyquist frequency was defined in (3.3). In this way, we have to transform from time to the frequency domain some discrete, in general non-periodic, signal. We can not apply the Fourier transform (3.20) since it requires the initial signal to be continuous. However, there exists the analog of Fourier transform that works with digital data, and that is Discrete Fourier Transform (DFT) [SMI99].

Let us suppose that the initial analog signal was sampled at regular intervals with the sampling period T and we have acquired the N samples $\{x(nT)\} = x(0), x(1),..., x([N-1]\cdot T)$ as it was described in (3.1). The DFT of $\{x(nT)\}$ is then defined as the finite sequence of complex values in the frequency domain:

$$\{X(k\Omega)\} = X(0), X(\Omega),..., X([N-1]\cdot\Omega)) \tag{3.24}$$

where

$$\Omega = \frac{2p}{(N-1)T} \tag{3.25}$$

is the first harmonic. As we mentioned above, DFT produces the sequence of complex values, hence, on the analogy to (3.21, 3.22) we can calculate the voltage and phase angle at frequency $k\Omega$ as follows:

$$\begin{aligned} |X(k\Omega)| &= \frac{X(k\Omega)}{e^{if(k\Omega)}} \\ f(k\Omega) &= \tan^{-1}\left(\frac{\mathrm{Im}[X(k\Omega)]}{\mathrm{Re}[X(k\Omega)]}\right) \end{aligned} \tag{3.26}$$

Therefore, the N real values $\{x(nT)\}$ are transformed to N complex numbers $\{X(k\Omega)\}$ by application of DFT:

$$X(k\Omega) = F_{discrete}(\{x(nT)\}) = \sum_{n=0}^{N-1} x(nT)e^{-ik\Omega nT}, \quad k = 0, 1, ..., N-1 \tag{3.27}$$

This equation is the discrete analogs of continuous Fourier transform (3.20) when we will consider in (3.20) that $x(t) = 0$ *for* $t \notin [0, (N-1)T]$ and put $x(t) = x(nT), k\Omega = w$ *and* $nT = t$.

The Discrete Fourier Transform has a number of useful properties that can simplify calculations and transformations [IJ02]:

- *Symmetry.*
  $\mathrm{Re}[X(N-k)] = \mathrm{Re}[X(k)]$. The amplitude spectrum has the property of symmetry.
  $\mathrm{Im}[X(N-k)] = -\mathrm{Im}[X(k)]$. Phase spectrum is antisymmetrical.

- *Simplification for even functions.*
  We call function to be *even* if for each n $x(-n) = x(n)$. For even functions the following simplification of DFT can be done:

$$X(k) = \sum_{n=0}^{N-1} x(n)\cos(k\Omega nT) \tag{3.28}$$

- *Simplification for odd functions.*
  Odd function is that one where for each n $x(-n) = -x(n)$. For odd functions the cosine component of DFT goes to zero, hence:

$$X(k) = -i\sum_{n=0}^{N-1} x(n)\sin(k\Omega nT) \tag{3.29}$$

- *Parseval's Theorem.*
  The average energy of a signal can be calculated in frequency domain by use of following formula:

$$\sum_{n=0}^{N-1} x^2(n) = \frac{1}{N} \sum_{k=0}^{N-1} |X(k)|^2 \tag{3.30}$$

In the left side of this equation we have the mean energy of the signal which is performed via time domain values. In right side of the equation (3.30) we have the mean spectral amplitude of transformed signal.

- *Convolution.*

  Convolution is one of the most important mathematical operations in DSP. It allows to perform the output signal of the system through the input signal and impulse response of the system [PM92, SMI99]. The convolution of two signals is defined as:

$$z(n) = x(n) \otimes y(n) = \sum_{k=0}^{N-1} x(k) y(n-k) \tag{3.31}$$

If z is discrete output signal of the system than it can be calculated as convolution of discrete input signal x and N samples long impulse response signal of the system y.

There is one very useful property of DFT that in frequency domain the analog of convolution is the multiplication of discrete signals:

$$\begin{aligned} z(n) &= x(n) \otimes y(n) = F^{-1}_{discrete}[X(k) \cdot Y(k)] \\ Z(k) &= X(k) \cdot Y(k) \end{aligned} \tag{3.32}$$

Hence, as shown in (3.32), the relatively complex operation of convolution in time domain becomes just a simple multiplication in frequency domain.

Chapter 4

Speaker Features

4.1 Introduction

An acoustic signal that represents human speech contains not only the content of speech but also a lot of information about the speaker: language, dialect, emotional state, gender and many others [NAI90]. History notes that even in 17th century there was attempt of speaker recognition when a witness tried to know an accused by his voice during one session that was called to reveal the details of the death of Charles I [NRC79]. Since that time a lot of work was done to create algorithms that solve problem of identification based on aural and visual methods that human do to identify each other. However, such an approach is not suitable for automatic speaker recognition since it will have too complicated calculations and measurements [NAI90].

For automatic speaker recognition it appears to be much more efficient to use some parameters or "features" that can be extracted either from raw speech signal or from the waveform that is represented in frequency domain. Among them are such acoustic parameters as intensity, pitch, short-time spectrum, predictor coefficients, formants, nasal coarticulation, spectral correlations, speaking rate and others [ATA76]. In other words, speech signal can be represented by set of feature vectors that allows the use of mathematical tools and methods for task of speaker recognition. This procedure is known among speech scientists as ***feature extraction*** [CAM97].

Raw speech signal contains quite large amount of data while not all this information is essential for speaker recognition. In this way, the main goal of feature extraction is to reduce the amount of information to be handled while keeping all necessary peculiarities that are almost unique to each person and allow distinguishing speakers from each other. Methods to extract efficient acoustic parameters were discussed extensively in literature. In paper [WOL72] there was suggested a set of characteristics that ideal feature should posses:

1) efficient to represent the information that allows to distinguish speakers
2) easy to measure
3) stable over time

4) desired feature should occur naturally and frequently in speech
5) environment independent
6) does not allow to mimicry

In real life it is not possible to produce such a feature that will satisfy all the demanded properties and, therefore, there always will be a kind of tradeoff between them that depends on the main goals and requirements of a speaker recognition application.

4.2 Windowing and Framing.

Speech signal is a slow varying signal over time or signal with *nonstationary dynamics* [DHP00]. Hence, it is useful to process not the whole signal at once but some smaller part of it considering the signal to be quite stationary and having almost nonvarying speech and speaker characteristics during this period. In Digital Signal Processing such a small portions of signal are referred as *frames* and processing of them is called *short-term analysis*.

In this way, features are extracted not from the whole signal but from small intervals. Let us describe this schema in more details: firstly the length of frame is selected. Next, window of this predefined length is moved across the signal. Each following window is overlapping with the previous one (usually for 20-50% of the frame length). This overlapping is necessary to prevent the loss of information on the sides of the frame. To prevent the big influence of neighboring frames and to avoid the abrupt ending of a frame the signal inside the frame is multiplied by the *window function* [FUR01]. The resulting windowed frame $f(n)$ of the signal $x(n)$ witch has a length of N samples is obtained as:

$$f(n) = x(n)w(n) \tag{4.1}$$

where $w(n)$ is a window function. The idea of windowing and following feature extraction are schematically shown in Figure 4.1.

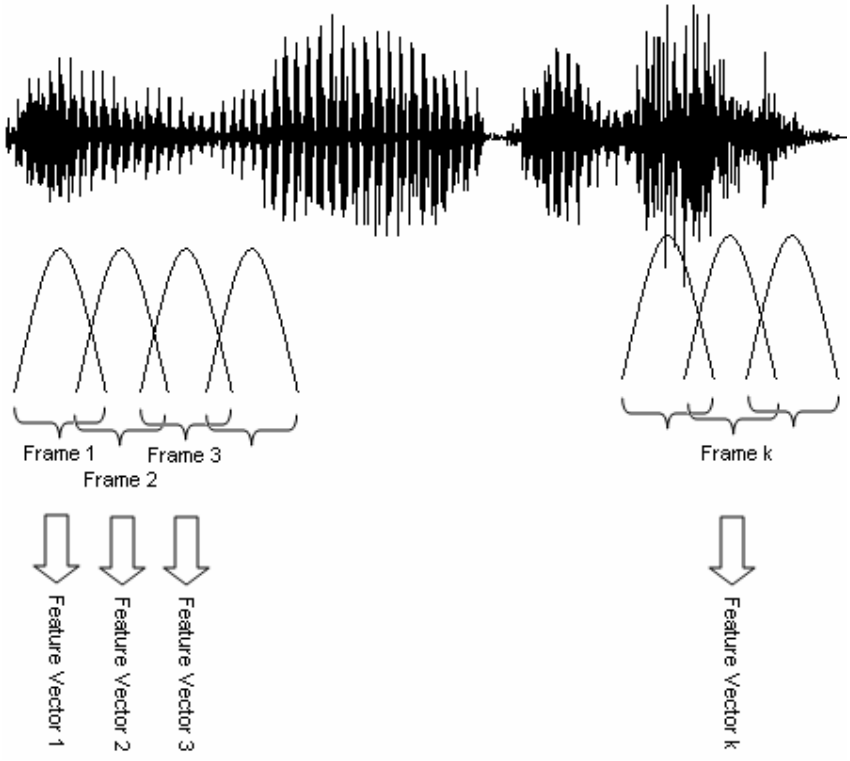

Figure 4.1. Schematic representation of windowing and feature extraction.

As it was mentioned in (3.31) DFT has a following property that multiplication of two signals in time domain is equal to convolution in frequency domain. Therefore, windowed frame in frequency domain can be calculated as:

$$F(w_n) = \sum_{k=0}^{N-1} X(w_n) W(w_n - w_k) \tag{4.2}$$

where F, X, W are corresponding DFT's of the f, x, w at frequency w.

For any windowing function it is desirable to satisfy two of the following conditions [DHP00]:

1) A narrow bandwidth main lobe
2) Strong attenuation of signal at sidelobes

The first one will allow the main lobe not to lose all the sharp details of the signal while the second feature helps to avoid spectral leak from the over parts of spectrum that are distorting the information about the signal at given frequency. Hence these desired features are almost contrary to each other every time choosing a window is a kind of trade-off between them.

The rectangular window that is preserving the initial signal unchanged has abruptly ending sidelobes. This window obtains the following properties:

- It has quite narrow main lobe which becomes smaller while the number of samples grows.
- The attenuation of sidelobes is not reducing with growing of N remaining almost constant. Typically, the influence of "noisy" side lobes is very big in case of rectangular window since they have only about 20dB spectrum magnitude if compared to main lobe.

In DSP there are used different types of windows: Kaiser, Hanning, Blackman and others [HAR78]. However, in speaker recognition area the most popular is Hamming window function which is defined as:

$$w_H(n) = 0.54 - 0.46\cos\left(\frac{2np}{N-1}\right) \tag{4.3}$$

The spectral features of the Hamming window can be described as follows:

- A main lobe is larger compared to rectangular window function. The width of main lobe is decreasing with the growth of number of samples.
- On the other hand, hamming window has much better performance on sidelobes. It reaches -30dB attenuation of "noisy" data at the sidelobes if compared with the magnitude at main lobe.

In this way, the choice of a window function is a state of art depending on what is more important for particular task. However, generally in speech processing smooth window functions are proffered still they provide much more attenuation at the side lobes preserving quite reasonable resolution at the main lobe.

We have already discussed a tradeoff between the resolution of main lobe and attenuation at sidelobes when choosing a window function. As we were considering signal to be stationary inside the frame the increase of frame size *N* should result in better performance of the system regardless of type of window function used. However, speech signal is not stationary in the larger scope. Hence, when a larger frame will slide across the signal it will be blurred more frequently by neighboring frames when such a long frame is crossing a border between different stationary parts of the signal. In this way, there occurs another trade-off when choosing the window length. The bigger frame will allow a higher resolution and more

precise spectral picture unless signal remains stationary. On the other hand, for the fast varying over time signal more narrow window should be chosen to avoid the distortion of data. This trade-off is referred as *spectral temporal resolution trade-off* [DHP00].

4.3 Long-term Average Spectrum (LTAS)

In applied mathematics and physics the one of the basic signal features is the power distribution of spectrum that allows to say what part of the whole signal energy is carried by some frequency range. There are two equivalent definitions of *power spectral distribution* (PSD) [KM81]. First is to obtain PSD as Fourier transform of autocorrelation function of time series. Second, that is usually used in speech processing is to calculate PSD as modules of Fourier transform of time series raised in square.

For analog signals last definition of PSD for the time range starting at t_1 and finishing at t_2 in terms used in (3.20) can be written as following:

$$PSD(f) = \frac{|X(f)|^2}{t_2 - t_1} = \frac{\left| \int_{t_1}^{t_2} x(t) e^{-i2p\ ft} dt \right|^2}{t_2 - t_1} \tag{4.4}$$

where $f=2\pi\omega$ is a cycle frequency. Since squared module of Fourier transform represents the energy of the whole signal starting from time t_1 to t_2 it is normalized by division on the time length of signal. In this way PSD has units of Pa^2/s.

It is easy to check that PSD is dividing the total power of a signal [SMI03]. To show this we have just to summate the total PSD and use Parseval's Theorem as follows:

$$\int_{-F}^{F} PSD(f) = \int_{-F}^{F} \frac{|X(f)|^2}{t_2 - t_1} df = \frac{1}{t_2 - t_1} \int_{-F}^{F} |X(f)|^2 df = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} |x(t)|^2 dt \tag{4.5}$$

where F is the highest frequency for Fourier transform of signal in corresponding time range.

For discrete systems the DFT is used to obtain spectral picture of signal in frequency domain. Therefore, it is useful to define PSD for discrete signal. According to the definition of DFT in section 3.5 the energy distribution for the discrete signal is calculated as follows [PTVF92]:

$$PSD(k) = \frac{|X(k)|^2}{\mathrm{N}(\mathrm{t}_2 - \mathrm{t}_1)} = \frac{\left|\sum_{n=0}^{N-1} x(n)e^{-ikn}\right|^2}{\mathrm{N}(\mathrm{t}_2 - \mathrm{t}_1)} \tag{4.6}$$

Here there are two multiplicative normalization factors: number of samples *N* which depends on sampling rate and frame length and the frame length t_2-t_1. The first one is necessary to be able to compare signals of different sampling rate when the last one allows to compare signal samples of different time duration.

In this way we have defined how to calculate power spectral density for the one frame of the signal performing short-term analysis. However, speech signal is in general a nonstationary signal. Hence, to obtain the energy distribution of the whole sample it is reasonable to calculate the *long-term average spectrum* (LTAS) or *periodogram* [WEL67]. The method of calculation average spectrum for the whole signal proposed by Welch consists of four steps:

1. The whole signal is divided into L overlapping parts, each consisting of N points.
2. For each frame the N-point DFT is performed. After that the suitable window function is applied to this frame as it was described in section 4.2.
3. For the spectrum calculated in step 2 the PSD is calculated according to (4.6)
4. Finally, the LTAS is calculated as the average value of L power spectral densities which were obtained in step 3.

In this way, the LTAS as a function of frequency can be calculated as:

$$LTAS(f) = \frac{1}{L}\sum_{i=1}^{L} PSD_i(f) \tag{4.7}$$

where $PSD_i(f)$ is a power spectral density for the i-th windowed frame of signal. Since PSD is measured in units of Pa^2/s the LTAS feature has the same units.

In acoustics PSD measure is usually related to $P_0 = 2 \cdot 10^{-5} Pa$ which is considered to be a threshold of human hearing at frequency 1kHz [LN72]. Hence, LTAS that is calculated relatively to this threshold can be calculated by following formula:

$$LTAS_{dB}(f) = 10\log_{10}\left(LTAS(f)/P_{thr}^2\right) \tag{4.5}$$

Such a normalized LTAS of speech signal is measured in dB/Hz units [PRA06]. This type of sound pressure representation via *dB* was discussed in details in subchapter 2.1. An example of original signal and its $LTAS_{dB}$ can be seen in Figure 4.2.

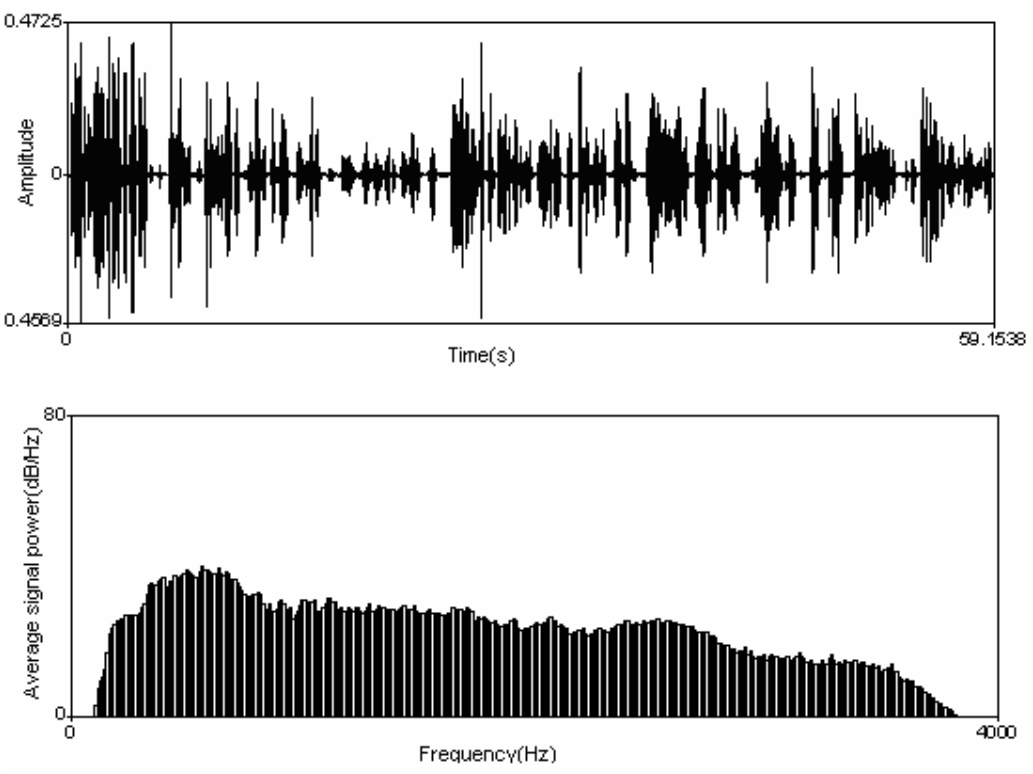

Figure 4.2. Speech signal of 59s length and its $LTAS_{dB}$ distribution.

Sometimes it is useful to know what part of total energy is carried in average by some frequency band. After calculating LTAS it is possible to answer this question as shown in following formula:

$$\text{average energy of subband (f1, f2)} = \sum_{f=f_1}^{f_2} LTAS(f) \tag{4.6}$$

Since PSD as it was shown in (4.5) is dividing the total power of signal, the LTAS is dividing the total average power of signal. In this way, summation of LTAS feature by all frequencies will give us the total average power of speech.

The choice of parameter L affects the performance in two ways. Firstly, dividing signal into smaller parts and calculating the average power of long-time signal allows to assign power to the correct frequencies and reduces noise-based variations in power distribution. However, the smaller frames provide the smaller frequency resolution since there

are less samples in original signal. In this way, there is a kind of trade-off between accuracy of assigning power to correct frequencies and frequency resolution.

4.4 Mel-Frequency Cepstrum Coefficients (MFCC)

In present time speech processing methods are based on the speech production model that present a speech signal as a composition of two: relatively fast varying pseudoperiodic source signal ***e(n)*** and relatively slow varying impulse response of a vocal tract ***h(n)***. The latter plays a role of linear digital filter with slow varying parameters [FUR01]. Hence, the discrete speech signal ***s(n)*** can be presented in time domain as convolution of these two parts:

$$s(n) = e(n) \otimes h(n) \tag{4.7}$$

In the most of speech processing applications it is needed to work with only one of these components. However, we can observe only the output signal ***s(n)*** but can not calculate easily its parts since they are convolved. Generally, in digital signal processing it is quite a challenging problem to separate the signal into additive components.

The *cepstrum* concept proposed by Bogert et al. in [BOG63] proposes the solution for this problem presenting original signal as a two additive parts. It is calculated as the discrete inverse Fourier transform of the logarithm of the magnitude spectrum.

It is easy to see that cepstrum representation allows to separate the signal into additive parts. After performing DFT and calculating spectrum we move from time domain to the frequency domain and thus convolution becomes multiplication. After that the logarithm is taken and multiplication becomes desired addition. Finally, the linear operation of inverse DFT is applied resulting in that slow varying impulse response and fast varying source signal are lying in different parts of range of the new independent variable named *quefrency*.

The main interest of speech recognition is in evaluation of the low-quefrency part of cepstrum that represents the slow varying vocal tract parameters. For this kind of a low-quefrency speaker parameters analysis usually mel-frequency cepstral coefficients (MFCC) are used.

The idea of MFCC calculation is based on cepstrum calculation. However, we have to explain where the term *mel* comes from. Stevens and Volkman in [SV40] have shown that human perception of the pitch of sound does not have linear dependence on the actual physical value of sound frequency. Thus, the term mel was used to define the sound frequency that is perceived by the human. Experiments show that mel correspond to frequency almost

linearly before 1kHz and logarithmic above this value. This relation is usually described by the following formula [FUR01]:

$$f_{mel} = 1000 \log_2 (1 + f) \tag{4.8}$$

Then MFCC's are calculated as a logarithm from filterbank magnitudes. One filter corresponds to one coefficient. These filters have a triangle shape and each of them calculates average spectrum around each central frequency. These filters have the increasing length with growth of frequency as shown in Figure 4.2:

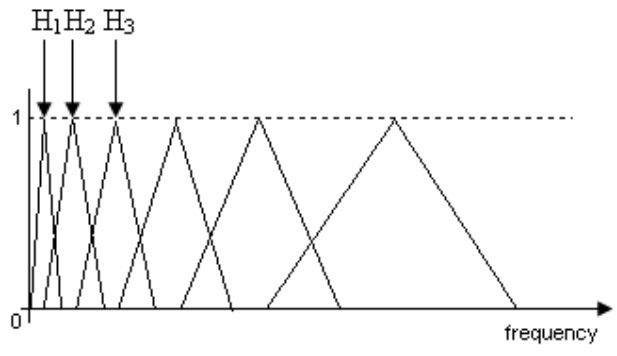

Figure 4.2. Filters of triangle shape used to compute MFCC.

The calculation of MFCC can be divided into following steps:

1. Framing of the original speech signal and windowing:

$$x(n) = s(n) \otimes w(n) \tag{4.9}$$

2. Computation of the spectrum by use of DFT (in applications FFT is used instead of DFT to improve the time performance) as shown in (3.27).
3. Filling the filterbank of each frame by the multiplication of each frame magnitude spectrum to the corresponding filter H_k. After this procedure the M values of energy are obtained, where M is a number of triangle filters in filterbank.
4. Logarithm from these energy values is taken. Hence, the logarithmic energy for each filter can be presented as follows:

$$l(m) = \log\left(\sum_{k=0}^{N-1} |X(k)| H_m(k)\right), \quad where \ m = 1,2,...,M \tag{4.10}$$

5. Finally, inverse DFT is applied (in applications discrete cosine transform is applied instead to reduce the number of calculations):

$$c(n) = \sum_{m=1}^{M} l(m) \frac{\cos(m-1,5)}{M} \tag{4.11}$$

Here *c(n)* are the mel-frequency cepstral coefficients. In speech and speaker recognition the first coefficients from 12 to 20 are chosen to represent the parameters of the slow varying impulse response of a vocal tract.

Chapter 5

Experiments and Discussions

5.1 Introduction

In this chapter we present the experimental part of the thesis containing the study of the LTAS feature performance in the task of speaker identification. The data used for experiments was quite close to everyday speech with some amount of noise and was taken from NIST 1999 speaker recognition corpus as described in subchapter 5.2.

This chapter covers the two main issues: the study of performance of LTAS feature as an independent classifier for speaker recognition including various parameters influence on the identification performance; and the study LTAS and MFCC score fusion for the task of speaker identification. These two features detailed description can be found in subchapters 4.3 and 4.4.

All the experiments were performed with the software tools developed in our department [SKHF, KKF04]. For the LTAS feature processing and automatic features scores fusion the additional modules were developed and integrated in these system. Generally, the closed-set text-independent speaker identification task was performed during the experimental part.

In the following subchapters we firstly describe dataset used for experiments, after that in 5.3 the LTAS parameters tuning is performed, in 5.4 we have studied the performance of fused MFCC and LTAS and, finally, in 5.5 the summary of the experimental results is presented.

5.2 Dataset

For the final experiments we have used dataset extracted from NIST 1999 speaker recognition corpus [NIST99]. The subset extracted from that corpus that is used in our experiments consists of 230 male persons speech samples. The detailed list of these files from NIST 1999 speaker recognition corpus can be found in Appendix A.

The basic NIST 1999 corpus comes from Linguistic Data Consortium [LDC06]. Most of the speakers were college and university students from the South of the United States. In general, they did not know each other before recording. For each speaker there was allowed only one call to send or receive per day. There was proposed a topic for conversation, however, speakers were quite free to speak about anything they want with only demand not to make big silent pieces. To avoid large pauses during the samples they were chosen closer to the end of conversation when participants were feeling themselves more relaxed and talkative [MP00].

In the Table 5.1 there are shown the main features of the male subset of NIST 1999 speaker recognition corpus.

Table 5.1. Parameters of the NIST 1999 male subset.

Subset	230 male speakers
Language	English
Speech type	Conversation with removed silent frames
Quality	Telephone lines
Environment	Indoors with some noise
Training samples length	56-65 sec.
Test samples length	56-65 sec.
Sampling rate	8 kHz
Quantization	16-bit μ-law

In our experiments we have used 2 samples for each of 230 speakers. Speech fragments marked with letter “a” were used for the training purposes when the samples marked with letter “b” were used for testing during the identification phase. The list of files that forms this NIST99 corpus subset of 230 male speakers can be found in Appendix A.

5.3 LTAS Feature Performance

5.3.1. Feature vector size

In section 4.3 it was already mentioned that power spectral density is usually calculated for some frequency subband. Therefore, one of the basic LTAS parameters is the quantization level which defines on how many subbands the whole frequency range will be divided. The increase of the number of subbands saves more information about the signal and thus allows

performing a recognition task more precisely. However, with grow of number of quantization levels the time complexity of speaker recognition based on LTAS feature is increased as *O(M)*, where *M* is a number of quantization levels.

The first experiment defines how the decrease of number of frequency subbands affects the performance of a system. For this experiment we have used one of the most frequently used set of windowing parameters in speaker recognition that are described in Table 5.2.

Table 5.2. Parameters for measurement of influence of frequency range quantization level on the performance of LTAS.

Window function	Hamming
Window size	30ms
Window shift	20ms

Results of this experiment that show the correlation between error rate and number of frequency subbands are presented in Figure 5.1.

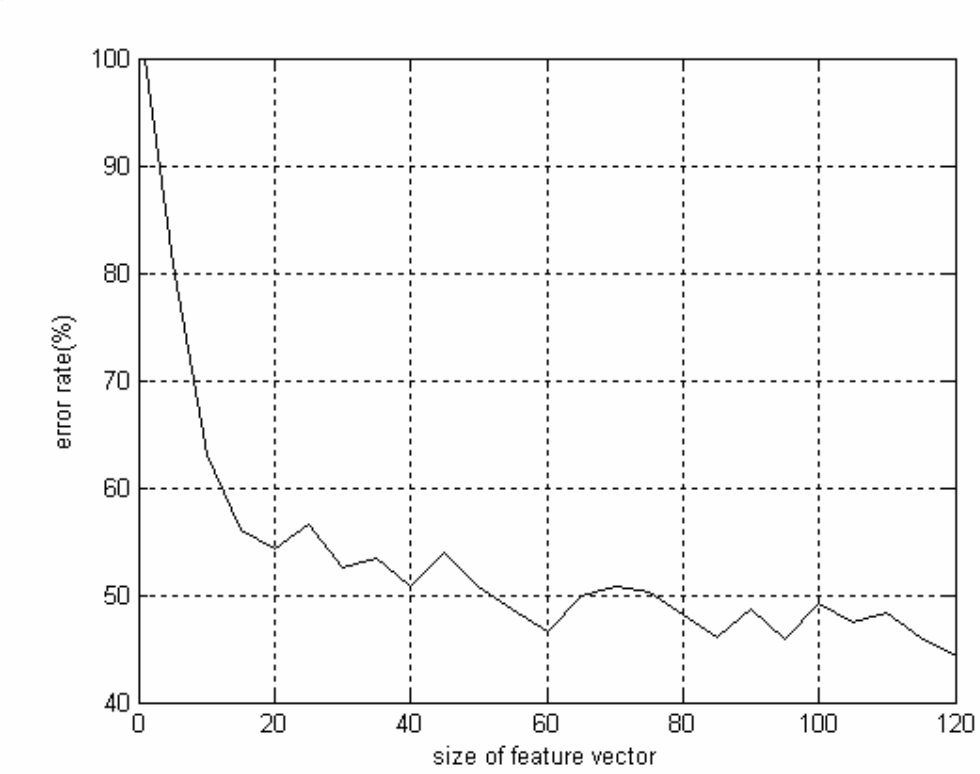

Figure 5.1. Influence of LTAS vector size on identification.

In this way, the recognition rate increases when feature vector size is growing. However, it should be noticed that there is a physical upper limit for the size of feature vector according to following formula:

$$frequency\ range\ size = \frac{window\ size}{2T_s} + 1 \tag{5.1}$$

where T_s is a sampling period as defined in (3.1). This limitation follows from symmetry of amplitude spectrum and definition of DFT which were discussed in chapter 3.5. In this way, in future experiments we will always use the maximum available size for feature vector which can be calculated as (5.1) and will vary only the window size parameter.

5.3.2. Window size

In subchapter 4.3 we have discussed that while choosing the number of frames there is a kind a tradeoff between frequency resolution and accuracy of power assignment to correct frequencies. Firstly we want to mention that knowing the total speech signal length we can easily obtain window size by simple division of signal length to the number of frames. Hence, there appears a natural question: what is the optimal window size for LTAS feature in the task of speaker identification? Next experiment allows us to establish some trends and answer to this question. For this experiment we have used the Hamming window function and we have performed no quantization while calculating power distribution over the frequency band. The shift size was calculated as 2/3 of window size. The results are presented in Figure 5.2.

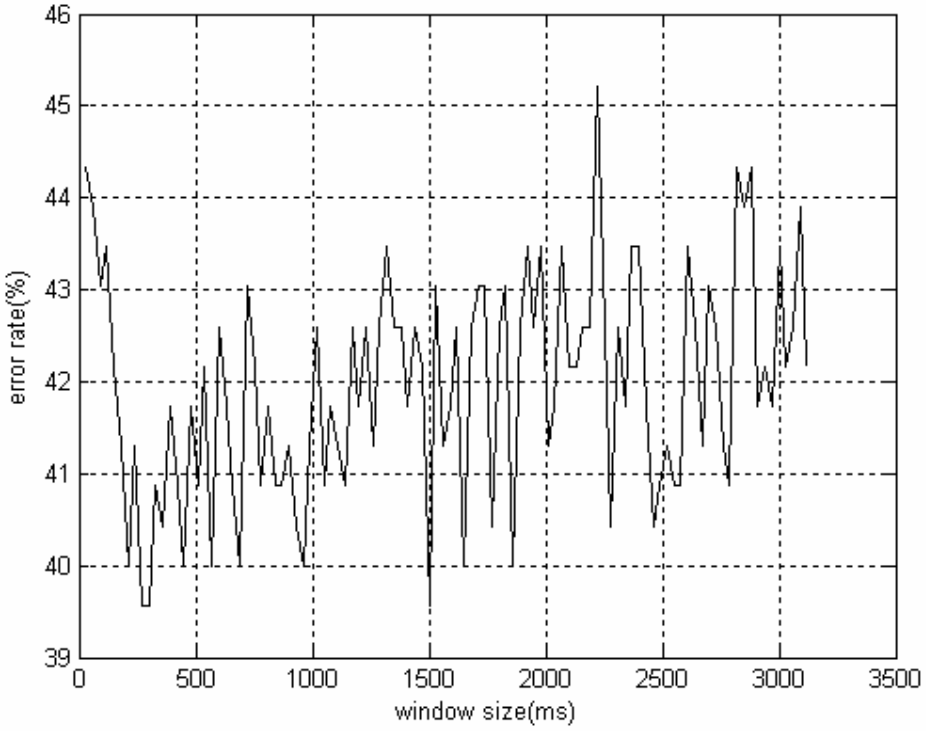

Figure 5.2. Relation between identification error rate and window size.

From this graph and Table B.2 we can see that on the subrange from 0 to 200ms window length the error rate is reducing very fast due to the lack of frequency resolution at each window. From 200 to 350ms the best performance of LTAS can be observed with 40,00% of error rate in average and 39,56% in best cases. Starting from 350ms the average identification error is growing with the increase of window length. Thus, the increasing spectral resolution can not compensate the less accurate frequency assignment due to the growth of window size. Additionally to the trade-off between resolution and assignment accuracy that was discussed in subchapter 4.3 there is another moment that should be taken into consideration. In subchapter 3.5 we have noticed that the time complexity of Fast Fourier Transform (FFT) algorithm that was used to perform DFT and obtain spectrum is *O(NlogN)* [CT65]. Thus, calculating windows of large size is not acceptable for real-time application of speaker identification. Therefore, in future experiments we will use the relatively small window of 300ms and 200ms shift size (error rate = 40,43%) that allows to extract the whole LTAS vector of signal by use of FFT quite fast (approximately 0.53s for 60s speech sample).

5.3.3. Speech sample length

Speech signal is considered to be stationary during small time frames. However, in a long speech sample the speech signal characteristics are changing with time. Since different phonemes have different waveforms for the same speaker it is necessary to collect some statistics about LTAS distribution using various phonemes. For example, if we would try to verify speaker by comparing LTAS of only two phonemes from his speech it would be very challenging task since there are some differences in power distribution that do not depend on speaker but only on phonemes themselves. An example of this can be seen in Figure 5.3 where long-term average spectrums of two phonemes recorded in the same environment and by the same speaker are calculated.

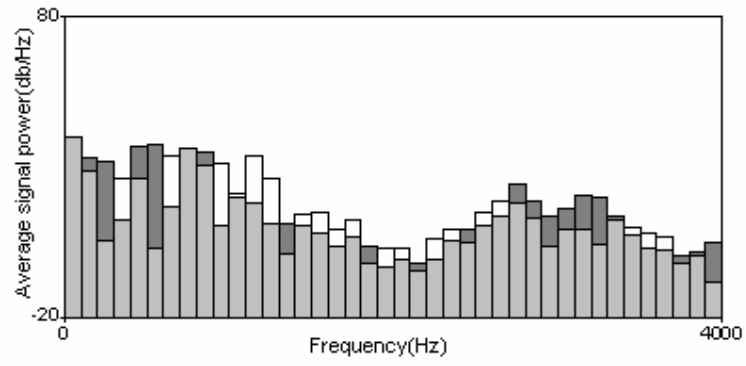

Figure 5.3. Average power distribution for sounds “a” (white) and “o” (dark grey).

However, if the speech length is quite big then the influence of separate phonemes on LTAS is reduced and the average power distribution becomes more speaker-dependent rather speech dependent.

Speaker identification application as it was discussed in subchapter 1.2 generally consists of two modules: one creates and stores speaker models while another is solving the problem of identification. The first task is not a real-time problem and is not performed frequently, hence, the calculations time is not critical parameter. However, the most usable second part is a real-time application and the identification time is crucial.

In next experiment for the role of samples for the speaker modeling the long 56-65s samples are chosen while the length of testing samples is varying from 2s to 56s. This will give us the information about how the length of testing sample affects the identification performance. The parameters for this experiment are listed in Table 5.3.

Table 5.2. Parameters for the measurement of testing speech samples length influence on the identification error rate.

Training samples length	56-65s
Testing samples length	Varying parameter from 2s to 56s
Window function	Hamming
Window size	300ms
Window shift	200ms

The results of the experiment are presented in Figure 5.4. The this graph training speech sample length is an independent parameter and identification error rate is plotted versus it.

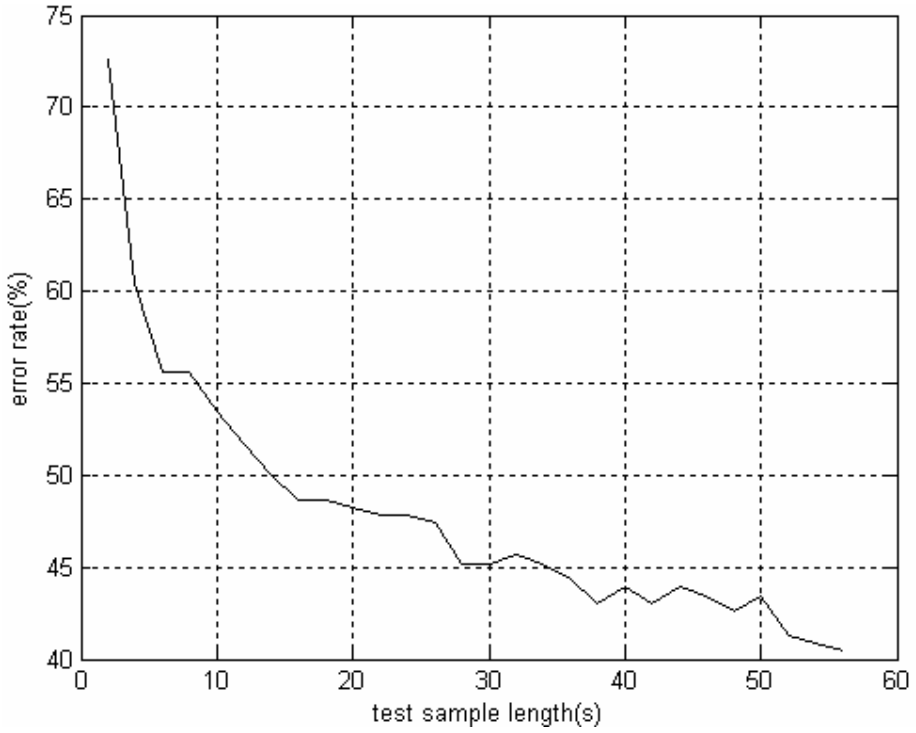

Figure 5.4. Identification performance change with the growth of testing sample size.

By this experiment we can examine that the theoretical discussion in the beginning of this subchapter about the correlation between sample length and identification performance was right. Actually, the error rate is reducing with growth of the sample length. Additionally, we have to notice that under the 30s testing sample length the mapping is close to exponential while above this value it is almost linear.

5.4 Scores Fusion of LTAS and MFCC

After analyzing the LTAS behavior for ASI problem as a separate feature we have studied the scores fusion of the LTAS and MFCC. The parameters for MFCC tuning were set as it was discussed in the works of Kinnunen *et al.* [KHF03, KHF 04, KKF04]. This settings list can be seen in the Table 5.3.

Table 5.3. MFCC parameters for fusion.

Window size	30ms
Window shift	20ms
Window function	Hamming
Number of MFCC's	12
Mel filters	27
Model	Vector quantization
Codebook size	64

The identification based on scores fusion consists of the following steps:

1. Calculate separately distances from unknown speaker to speaker database models for both features d_{LTAS} and d_{MFCC} .
2. Normalize distances that will lead to the following: $0 \leq d_{LTAS} \leq 1$ and $0 \leq d_{MFCC} \leq 1$. This procedure consists of two steps:
    A. Select the speaker and find the maximum distance from him to the database.
    B. Divide all the other distances from this speaker to the database to maximum distance value.
3. Calculate combined distance as:

$$d_{FUS} = a \cdot d_{MFCC} + (1-a) \cdot d_{LTAS} \tag{5.2}$$

4. The model with the smallest fused features distance d_{FUS} is used to identify the test speaker.

For LTAS feature calculation we have used the parameters shown in Table 5.4.

Table 5.4. LTAS parameters for fusion.

Window function	Hamming
Window size	300ms
Window shift	200ms
Feature vector size	1200

In the first experiment we have studied how scores fusion of LTAS and MFCC is performing for original NIST99 data with no additional noise. In the rest of this chapter we will reference to original NIST99 samples as “noiseless” and data with additional noise as “noisy”. The parameters for these features were set as described in Table 5.3 and Table 5.4. The weight of MFCC a from (5.2) was varied from 0 to 1 with the step of 0.01. The results of this experiment are presented in the following graph while the whole list of values can be found in Table B.4.

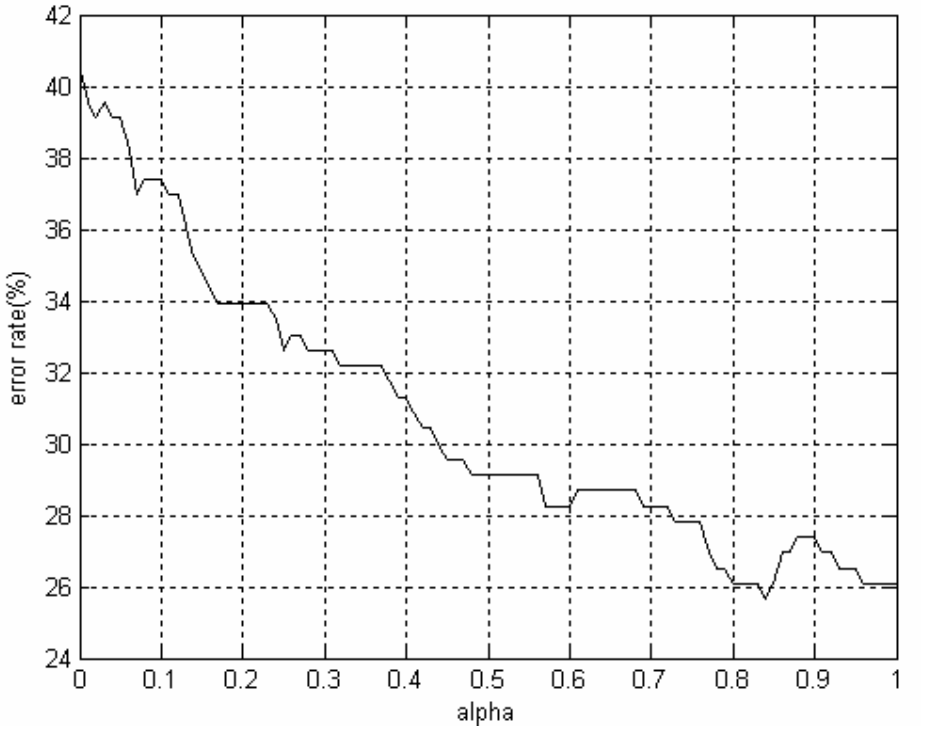

Figure 5.5. Fusion of LTAS and MFCC performance in noiseless case.

For noiseless data of 230 male speakers the identification error rates for pure features LTAS and MFCC are respectively 40,43% and 26,08%. The best result was obtained with $a = 0{,}84$ when the identification error rate had a value of 25,65%. In spite of that fact the general performance of these two features fusion in case of noiseless data can be described as

follows: for $a > 0{,}8$ the fusion does not seriously affect the system performance, for $a \leq 0{,}8$ the with the increase of LTAS weight which is equal to $(1-a)$ the identification error rate is growing up to maximum value of 40,43%.

In this way, there is no advantage in fusion of LTAS and MFCC for noiseless data since that does not reduce the identification error. However, the next experiment shows that the picture can change in case of noisy data.

The original samples were recorded from telephone calls and there are only small occasional background noises. We have changed the situation by applying to all samples the monotonous background noise that was recorded on the factory at working time [NOI06]. The additive combination of NIST99 speech samples and this factory noise was done at signal-to-noise ratio (SNR) of 10dB. The results of the features fusion performance in case of noisy data are as follows.

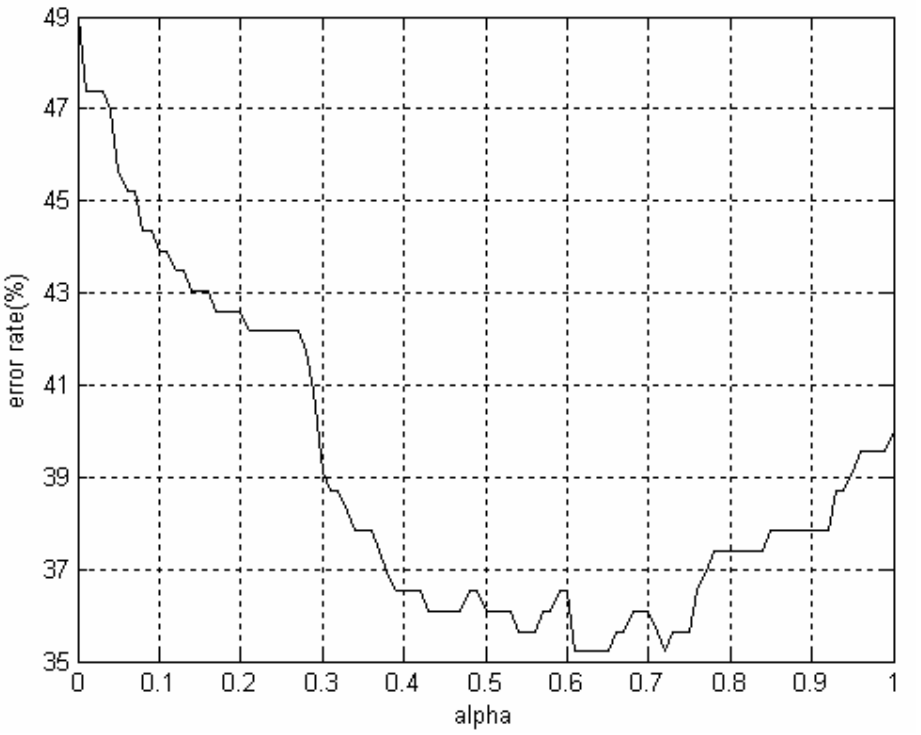

Figure 5.6. Fusion of LTAS and MFCC performance for noisy data.

The separate performance of features in noisy case degrades from 40,43% to 49,13% for LTAS and from 26,08% to 40,00% for MFCC. However, the scores fusion in this case allows to improve the identification performance when with $a = 0{,}61$ the error rate has a value of 35,21%. The whole list of values can be found in Table B.5.

As it is shown in Figure 5.6 the identification error rate in noisy case has a trend to be lower when MFCC and LTAS are mixed. The best results are obtained when the weight coefficient takes values from the following range: $0{,}6 \le a \le 0{,}75$. Thus, the scores fusion of MFCC and LTAS for the task of speech-independent ASI can improve the system performance for noisy speech samples if compared to the pure MFCC feature.

Chapter 6

Conclusions

Long-term average spectrum is one of the basic characteristics of a digital signal and it shows the energy distribution over the frequency band. In this work we have studied the LTAS as a feature for the text-independent automatic speaker identification. In the first part of the work we have studied ASR basics and human speech production. After that we have discussed necessary DSP basics and concept of the speech features extraction for the ASI.

The experimental part was divided into two parts. Firstly we have studied the LTAS performance as a separate feature for the speakers' identification and have found out the influence of such parameters as feature vector size, window size and testing sample length on the identification accuracy. After that LTAS was studied as an additional to MFCC feature based on the concept of scores fusion.

From the first experimental part it was concluded that LTAS has the best results for ASI with Hamming window of 200-360ms length and 2/3 of this window shift and no frequency band quantization. In this case identification error rate is about 40% for the original data that was taken from the NIST99 speaker recognition corpus. The study of influence of the test sample length on the feature behavior shown that when taken from 30 to 60s the error rate does not exceed 45%, however, for test samples smaller than 30s length the LTAS performance is reducing at a high speed.

The scores fusion of LTAS and MFCC has shown that the use additional LTAS feature for identification in case of signals with low noise rate does not provide any benefits. From the other side, for noisy data when separate features performance reduces to error rates of 49,13% for LTAS and 40,00% for MFCC, the scores fusion has better results improving identification error rate down to 35,21%.

In this way, in case of noisy speech the use of LTAS as an additional feature can improve the total ASI system work. The LTAS performance can be extended further. Now in LTAS vector all the frequencies are treated the same, however, some frequencies in general can contain more speaker-specific information than others. Thus, the LTAS with weighted components could be studied as a speaker recognition feature.

References

[ATA76] B.S.Atal, “Automatic Recognition of Speakers from Their Voices”. Proceedings of the IEEE, Vol.64, No.4, April 1976, pp.460-475.

[BCF73] A.C.Bajpai, I.M.Calus, and J.A.Fairley, “Mathematics for Engineers and Scientists”, Volume 2. Wiley, New York, 1973.

[BOG63] B.Bogert, M.Healy and J.Tukey, “The Quefrency Analysis of Time Series for Echoes: Cepstrum, Pseudo-autocovariance, Cross-cepstrum a Shape Cracking”. Proceeding of the Symposium on Time Series Analysis, NY, 1963, pp.209-243.

[CAM97] J.Campbell, “Speaker Recognition: A Tutorial”. Proceedings of the IEEE, Vol.85, No.9, 1997, pp.1437-1462.

[CT65] J.W.Cooley and J.W.Tukey, "An algorithm for the machine calculation of complex Fourier series". Math. Comput. 19, 1965, pp.297–301.

[DHP00] J.R.Deller, J.H.L. Hansen and J.G.Proakis, “Discrete-Time Processing of Speech Signals”, IEEE Press, 2nd edition, NY, 2000.

[DV90] P. Duhamel and M. Vetterli, "Fast Fourier transforms: a tutorial review and a state of the art," Signal Processing 19, 1990, pp.259–299.

[FLA72] J.L.Flanagan, “Speech Analysis, Synthesis and Perception”. Springer-Verlag, New York, 2nd edition, 1972.

[FOU03] J.Fourier, translated by A.Freeman, “The Analytical Theory of Heat”. Dover Publications, 2003 (published 1822, translated 1878, re-released 2003).

[FUR01] S.Furui, “Digital Speech Processing, Synthesis, and Recognition”. Marcel Dekker, 2nd edition, NY, 2001.

[GS94] H.Gish, M.Schmidt, “Text-Independent Speaker Identification”. IEEE Signal Processing Magazine, Vol.11, No.4, 1994, pp.18-32.

[HAH01] X.Huang, A.Acero, H.-W.Hon, “Spoken Language Processing”. Prentice Hall, Upper Saddle River, NJ, USA, 2001.

[HAR78] F.J.Harris, “On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform”, Proceedings IEEE, Vol.66, 1978, pp.51-84.

[HC99] J.Harrington, S. Cassidy, “Techniques in Speech Acoustics”. Kluer Academic Publishers, Dordrecht, 1999.

[IJ02] E.C.Ifeachor, B.W.Jervis, “Digital Signal Processing: A Practical Approach”, Prentice Hall, 2nd edition, 2002.

[KM81] S.Kay, S.L.Marple, “Spectrum Analysis - A Modern Perspective”. Proceeding of the IEEE, vol.69 (11), 1981, pp.1380-1419.

[KHF03] T.Kinnunen, V.Hautamäki and P.Fränti, “On the Fusion of Dissimilarity-Based Classifiers for Speaker Recognition”. In Proc. 8th European Conference on Speech Communication and Technology, 2003, pp. 2641-2644.

[KHF04] T.Kinnunen, V.Hautamäki and P.Fränti, “Fusion of Spectral Feature Sets for Accurate Speaker Identification”. In 9th International Conference Speech and Computer (SPECOM'2004), 2004, pp. 361-365.

[KIN05] T.Kinnunen, “Optimizing Spectral Feature Based Text-Independent Speaker Recognition”, Ph.D. thesis, University of Joensuu, 2005.

[KKF04] T.Kinnunen, E.Karpov and P.Frдnti, "Real-time speaker identification and verification", IEEE Trans. on Audio, Speech and Language Processing, 14 (1), 277-288, January 2006.

[LDC06] Linguistic Data Consortium, University of Pennsylvania, Philadelphia. WWW page, September 2006.
http://www.ldc.upenn.edu/

[LN72] P.Lindsay, D.Norman, “Human Information Processing: an Introduction to Psychology”. New York Academic Press Inc, 1972.

[MP00] A.Martin, M.Przybocki, “The NIST 1999 speaker recognition evaluation - An Overview”. Digital Signal Processing, 10, 2000, pp.1-18.

[NAI90] J.M.Naik, “Speaker Verification: A Tutorial”. IEEE Communications Magazine, 1990, pp.42-48.

[NIST99] National Institute of standards and technology. WWW page, September 2006. http://www.nist.gov/speech/tests/spk/1999/euro99_v2/

[NOI06] Noise data. WWW page, November 2006. http://spib.ece.rice.edu/spib/data/signals/noise/factory1.html

[NRC79] National Research Council, “On the Theory and Practice of Voice Identification”. Washington DC, 1979.

[PM92] J.G.Proakis, D.G.Manolakis, “Digital Signal Processing, Principles, Algorithms, and Applications”. Macmillan Publishing Company, New York, 1992.

[PRA06] Praat: doing phonetics by computer. WWW page, October 2006. http://www.praat.org/

[PTVF92] W.Press, S.Teukolsky, W.Vetterling, B.Flannery, “Numerical Recipes in C: the Art of Scientific Computing, Second Edition”. Cambridge University Press, NY, USA, 1992.

[REY02] D.Reynolds, “An Overview of Automatic Speaker Recognition Technology”. ICASSP 2002, pp.4072-4075.

[RJ93] L.R.Rabiner, B.W.Juang, “Fundamentals of Speech Recognition”. Prentice-Hall, Englewood Cliffs, NJ, USA, 1993.

[RS75] L.R.Rabiner, R.W.Shafer, “Digital Signal Processing of Speech Signals”. Prentice Hall, New Jersey, 1975.

[SHA86] D.O’Shaughnessy, “Speaker Recognition”. IEEE ASSP Magazine, October 1986, pp.3-17.

[SKHF] J.Saastamoinen, E.Karpov, V.Hautamäki and P.Fränti, “Automatic Speaker Recognition for series 60 mobile devices”. In 9th International Conference Speech and Computer (SPECOM), St.Petersburg, Russia, 2004, pp.353-360.

[SMI99] S.W.Smith, “The scientist and Engineer’s Guide to Digital Signal Processing”, California Technical Publishing, 1999.

[SMI03] Julius O. Smith III , ``Mathematics of the Discrete Fourier Transform (DFT), with Music and Audio Applications". W3K Publishing, 2003.

[SV40] S.Stevens and J.Volkman, “The Relation of Pitch to Frequency”. American Journal of Psychology, vol.53, 1940, p.329.

[SW49] C.E.Shannon, W.Weaver, “The Mathematical Theory of Communication”. University of Illinois Press, 1949.

[WEL67] P.D.Welch, “The Use of Fast Fourier Transforms for the Estimation of Power Spectra: A Method based on Time Averaging Over Short Modified Periodograms”. IEEE Transactions on Audio and Electroacoustics, vol.15, 1967, pp.70-73.

[WOL72] J.J.Wolf, “Efficient acoustic parameters for speaker recognition”. Journal of the Acoustical Society of America, 1972, pp.2044-2055.

Appendix A

Table A.1. Training data from NIST99 corpus.

1106a.wav	4160a.wav	4334a.wav	4516a.wav	4697a.wav	4884a.wav
1231a.wav	4162a.wav	4338a.wav	4529a.wav	4703a.wav	4889a.wav
1831a.wav	4173a.wav	4344a.wav	4531a.wav	4707a.wav	4893a.wav
3241a.wav	4179a.wav	4359a.wav	4533a.wav	4717a.wav	4897a.wav
3297a.wav	4184a.wav	4363a.wav	4535a.wav	4721a.wav	4902a.wav
3764a.wav	4187a.wav	4371a.wav	4536a.wav	4726a.wav	4903a.wav
4011a.wav	4192a.wav	4374a.wav	4539a.wav	4730a.wav	4910a.wav
4016a.wav	4194a.wav	4377a.wav	4543a.wav	4740a.wav	4914a.wav
4018a.wav	4206a.wav	4386a.wav	4550a.wav	4741a.wav	4923a.wav
4030a.wav	4213a.wav	4388a.wav	4554a.wav	4747a.wav	4927a.wav
4036a.wav	4217a.wav	4391a.wav	4564a.wav	4748a.wav	4928a.wav
4041a.wav	4218a.wav	4393a.wav	4567a.wav	4749a.wav	4931a.wav
4045a.wav	4237a.wav	4399a.wav	4569a.wav	4750a.wav	4946a.wav
4047a.wav	4239a.wav	4401a.wav	4570a.wav	4757a.wav	4949a.wav
4049a.wav	4241a.wav	4402a.wav	4572a.wav	4766a.wav	4951a.wav
4059a.wav	4242a.wav	4416a.wav	4583a.wav	4767a.wav	4952a.wav
4060a.wav	4250a.wav	4418a.wav	4586a.wav	4773a.wav	4954a.wav
4063a.wav	4252a.wav	4421a.wav	4589a.wav	4778a.wav	4960a.wav
4072a.wav	4259a.wav	4422a.wav	4591a.wav	4782a.wav	4961a.wav
4074a.wav	4261a.wav	4424a.wav	4593a.wav	4785a.wav	4966a.wav
4081a.wav	4263a.wav	4428a.wav	4595a.wav	4792a.wav	4969a.wav
4101a.wav	4270a.wav	4437a.wav	4596a.wav	4793a.wav	4971a.wav
4104a.wav	4276a.wav	4438a.wav	4600a.wav	4795a.wav	4972a.wav
4105a.wav	4286a.wav	4447a.wav	4610a.wav	4801a.wav	4977a.wav
4107a.wav	4289a.wav	4451a.wav	4613a.wav	4807a.wav	4987a.wav
4108a.wav	4292a.wav	4454a.wav	4621a.wav	4809a.wav	4990a.wav
4110a.wav	4295a.wav	4457a.wav	4633a.wav	4815a.wav	4991a.wav
4113a.wav	4298a.wav	4460a.wav	4638a.wav	4822a.wav	4996a.wav
4119a.wav	4302a.wav	4462a.wav	4640a.wav	4823a.wav	4998a.wav
4124a.wav	4306a.wav	4467a.wav	4643a.wav	4829a.wav	4999a.wav
4129a.wav	4308a.wav	4474a.wav	4648a.wav	4838a.wav	
4134a.wav	4309a.wav	4482a.wav	4653a.wav	4841a.wav	
4137a.wav	4313a.wav	4487a.wav	4656a.wav	4851a.wav	
4143a.wav	4314a.wav	4495a.wav	4658a.wav	4853a.wav	
4145a.wav	4316a.wav	4496a.wav	4662a.wav	4854a.wav	
4148a.wav	4322a.wav	4501a.wav	4669a.wav	4858a.wav	
4149a.wav	4325a.wav	4503a.wav	4675a.wav	4866a.wav	
4150a.wav	4328a.wav	4504a.wav	4686a.wav	4867a.wav	
4154a.wav	4330a.wav	4506a.wav	4691a.wav	4882a.wav	
4156a.wav	4332a.wav	4514a.wav	4694a.wav	4883a.wav	

Table A.2. Test data from NIST99 corpus.

1106b.wav	4160b.wav	4334b.wav	4516b.wav	4697b.wav	4884b.wav
1231b.wav	4162b.wav	4338b.wav	4529b.wav	4703b.wav	4889b.wav
1831b.wav	4173b.wav	4344b.wav	4531b.wav	4707b.wav	4893b.wav
3241b.wav	4179b.wav	4359b.wav	4533b.wav	4717b.wav	4897b.wav
3297b.wav	4184b.wav	4363b.wav	4535b.wav	4721b.wav	4902b.wav
3764b.wav	4187b.wav	4371b.wav	4536b.wav	4726b.wav	4903b.wav
4011b.wav	4192b.wav	4374b.wav	4539b.wav	4730b.wav	4910b.wav
4016b.wav	4194b.wav	4377b.wav	4543b.wav	4740b.wav	4914b.wav
4018b.wav	4206b.wav	4386b.wav	4550b.wav	4741b.wav	4923b.wav
4030b.wav	4213b.wav	4388b.wav	4554b.wav	4747b.wav	4927b.wav
4036b.wav	4217b.wav	4391b.wav	4564b.wav	4748b.wav	4928b.wav
4041b.wav	4218b.wav	4393b.wav	4567b.wav	4749b.wav	4931b.wav
4045b.wav	4237b.wav	4399b.wav	4569b.wav	4750b.wav	4946b.wav
4047b.wav	4239b.wav	4401b.wav	4570b.wav	4757b.wav	4949b.wav
4049b.wav	4241b.wav	4402b.wav	4572b.wav	4766b.wav	4951b.wav
4059b.wav	4242b.wav	4416b.wav	4583b.wav	4767b.wav	4952b.wav
4060b.wav	4250b.wav	4418b.wav	4586b.wav	4773b.wav	4954b.wav
4063b.wav	4252b.wav	4421b.wav	4589b.wav	4778b.wav	4960b.wav
4072b.wav	4259b.wav	4422b.wav	4591b.wav	4782b.wav	4961b.wav
4074b.wav	4261b.wav	4424b.wav	4593b.wav	4785b.wav	4966b.wav
4081b.wav	4263b.wav	4428b.wav	4595b.wav	4792b.wav	4969b.wav
4101b.wav	4270b.wav	4437b.wav	4596b.wav	4793b.wav	4971b.wav
4104b.wav	4276b.wav	4438b.wav	4600b.wav	4795b.wav	4972b.wav
4105b.wav	4286b.wav	4447b.wav	4610b.wav	4801b.wav	4977b.wav
4107b.wav	4289b.wav	4451b.wav	4613b.wav	4807b.wav	4987b.wav
4108b.wav	4292b.wav	4454b.wav	4621b.wav	4809b.wav	4990b.wav
4110b.wav	4295b.wav	4457b.wav	4633b.wav	4815b.wav	4991b.wav
4113b.wav	4298b.wav	4460b.wav	4638b.wav	4822b.wav	4996b.wav
4119b.wav	4302b.wav	4462b.wav	4640b.wav	4823b.wav	4998b.wav
4124b.wav	4306b.wav	4467b.wav	4643b.wav	4829b.wav	4999b.wav
4129b.wav	4308b.wav	4474b.wav	4648b.wav	4838b.wav	
4134b.wav	4309b.wav	4482b.wav	4653b.wav	4841b.wav	
4137b.wav	4313b.wav	4487b.wav	4656b.wav	4851b.wav	
4143b.wav	4314b.wav	4495b.wav	4658b.wav	4853b.wav	
4145b.wav	4316b.wav	4496b.wav	4662b.wav	4854b.wav	
4148b.wav	4322b.wav	4501b.wav	4669b.wav	4858b.wav	
4149b.wav	4325b.wav	4503b.wav	4675b.wav	4866b.wav	
4150b.wav	4328b.wav	4504b.wav	4686b.wav	4867b.wav	
4154b.wav	4330b.wav	4506b.wav	4691b.wav	4882b.wav	
4156b.wav	4332b.wav	4514b.wav	4694b.wav	4883b.wav	

Appendix B

Table B.1. Error rates for LTAS with Hanning window function, window size of 30ms, shift size of 20ms and varying vector size.

Feat.vector size	Error rate(%)	Feat.vector size	Error rate(%)	Feat.vector size	Error rate(%)
1	100	45	53.91	90	48.60
5	80.87	50	50.87	95	45.87
10	63.04	55	48.69	100	49.13
15	56.08	60	46.65	105	47.39
20	54.34	65	50.00	110	48.26
25	56.52	70	50.82	115	46.08
30	52.60	75	50.26	120	44.34
35	53.47	80	48.08		
40	50.82	85	46.04		

Table B.2. Error rates for LTAS with Hanning window function, no frequency band quantization, varying window size and shift size that is 2/3 of window size.

Window size	Error rate(%)	Window size	Error rate(%)	Window size	Error rate(%)	Window size	Error rate(%)
30	44,34	810	41,73	1590	41,73	2370	43,47
60	43,91	840	40,87	1620	42,60	2400	43,47
90	43,04	870	40,87	1650	40,00	2430	41,73
120	43,47	900	41,30	1680	42,60	2460	40,43
150	42,17	930	40,43	1710	43,04	2490	40,87
180	41,30	960	40,00	1740	43,04	2520	41,30
210	40,00	990	41,30	1770	40,43	2550	40,87
240	41,30	1020	42,60	1800	42,60	2580	40,87
270	39,56	1050	40,87	1830	43,04	2610	43,47
300	40,43	1080	41,73	1860	40,00	2640	42,60
330	40,87	1110	41,30	1890	42,60	2670	41,30
360	40,43	1140	40,87	1920	43,47	2700	43,04
390	41,73	1170	42,60	1950	42,60	2730	42,60
420	40,87	1200	41,73	1980	43,47	2760	41,30
450	40,00	1230	42,60	2010	41,30	2790	40,87
480	41,73	1260	41,30	2040	41,73	2820	44,34
510	40,87	1290	42,60	2070	43,47	2850	43,91
540	42,17	1320	43,47	2100	42,17	2880	44,34
570	40,00	1350	42,60	2130	42,17	2910	41,73
600	42,60	1380	42,60	2160	42,60	2940	42,17
630	41,73	1410	41,73	2190	42,60	2970	41,73
660	40,87	1440	42,60	2220	45,21	3000	43,47
690	40,00	1470	42,17	2250	42,60	3030	42,17
720	43,04	1500	39,56	2280	40,43	3060	42,60
750	42,17	1530	43,04	2310	42,60	3090	43,91
780	40,87	1560	41,30	2340	41,73	3120	42,17

Table B.3. Error rates for LTAS with Hanning window function, window size 300ms, shift size 200ms, modeling samples of full length 56-65s and varying testing sample length.

Testing sample length(s)	Error rate(%)	Testing sample length(s)	Error rate(%)	Testing sample length(s)	Error rate(%)	Testing sample length(s)	Error rate(%)
2	72.60	16	48.69	30	45.21	44	43.91
4	60.43	18	48.69	32	45.65	46	43.47
6	55.65	20	48.26	34	45.21	48	42.60
8	55.65	22	47.82	36	44.34	50	43.47
10	53.47	24	47.82	38	43.04	52	41.30
12	51.73	26	47.39	40	43.91	54	40.87
14	50.00	28	45.21	42	43.04	56	40.43

Table B.4. Fusion of LTAS and MFCC performance for noiseless data.

alpha	Error Rate (%)	alpha	Error Rate (%)	alpha	Error Rate (%)	alpha	Error Rate (%)	alpha	Error Rate (%)
0,00	40,43	0,20	33,91	0,40	31,30	0,60	28,26	0,80	26,09
0,01	39,57	0,21	33,91	0,41	30,87	0,61	28,70	0,81	26,09
0,02	39,13	0,22	33,91	0,42	30,43	0,62	28,70	0,82	26,09
0,03	39,57	0,23	33,91	0,43	30,43	0,63	28,70	0,83	26,09
0,04	39,13	0,24	33,48	0,44	30,00	0,64	28,70	0,84	25,65
0,05	39,13	0,25	32,61	0,45	29,57	0,65	28,70	0,85	26,09
0,06	38,26	0,26	33,04	0,46	29,57	0,66	28,70	0,86	26,96
0,07	36,96	0,27	33,04	0,47	29,57	0,67	28,70	0,87	26,96
0,08	37,39	0,28	32,61	0,48	29,13	0,68	28,70	0,88	27,39
0,09	37,39	0,29	32,61	0,49	29,13	0,69	28,26	0,89	27,39
0,10	37,39	0,30	32,61	0,50	29,13	0,70	28,26	0,90	27,39
0,11	36,96	0,31	32,61	0,51	29,13	0,71	28,26	0,91	26,96
0,12	36,96	0,32	32,17	0,52	29,13	0,72	28,26	0,92	26,96
0,13	36,09	0,33	32,17	0,53	29,13	0,73	27,83	0,93	26,52
0,14	35,22	0,34	32,17	0,54	29,13	0,74	27,83	0,94	26,52
0,15	34,78	0,35	32,17	0,55	29,13	0,75	27,83	0,95	26,52
0,16	34,35	0,36	32,17	0,56	29,13	0,76	27,83	0,96	26,09
0,17	33,91	0,37	32,17	0,57	28,26	0,77	26,96	0,97	26,09
0,18	33,91	0,38	31,74	0,58	28,26	0,78	26,52	0,98	26,09
0,19	33,91	0,39	31,30	0,59	28,26	0,79	26,52	0,99	26,09
								1,00	26,09

Table B.5. Fusion of LTAS and MFCC performance for noisy data.

alpha	Error Rate (%)	alpha	Error Rate (%)	alpha	Error Rate (%)	alpha	Error Rate (%)	alpha	Error Rate (%)
0,00	49,13	0,20	42,61	0,40	36,52	0,60	36,52	0,80	37,39
0,01	47,39	0,21	42,17	0,41	36,52	0,61	35,22	0,81	37,39
0,02	47,39	0,22	42,17	0,42	36,52	0,62	35,22	0,82	37,39
0,03	47,39	0,23	42,17	0,43	36,09	0,63	35,22	0,83	37,39
0,04	46,96	0,24	42,17	0,44	36,09	0,64	35,22	0,84	37,39
0,05	45,65	0,25	42,17	0,45	36,09	0,65	35,22	0,85	37,83
0,06	45,22	0,26	42,17	0,46	36,09	0,66	35,65	0,86	37,83
0,07	45,22	0,27	42,17	0,47	36,09	0,67	35,65	0,87	37,83
0,08	44,35	0,28	41,74	0,48	36,52	0,68	36,09	0,88	37,83
0,09	44,35	0,29	40,87	0,49	36,52	0,69	36,09	0,89	37,83
0,10	43,91	0,30	39,13	0,50	36,09	0,70	36,09	0,90	37,83
0,11	43,91	0,31	38,70	0,51	36,09	0,71	35,65	0,91	37,83
0,12	43,48	0,32	38,70	0,52	36,09	0,72	35,22	0,92	37,83
0,13	43,48	0,33	38,26	0,53	36,09	0,73	35,65	0,93	38,70
0,14	43,04	0,34	37,83	0,54	35,65	0,74	35,65	0,94	38,70
0,15	43,04	0,35	37,83	0,55	35,65	0,75	35,65	0,95	39,13
0,16	43,04	0,36	37,83	0,56	35,65	0,76	36,52	0,96	39,57
0,17	42,61	0,37	37,39	0,57	36,09	0,77	36,96	0,97	39,57
0,18	42,61	0,38	36,96	0,58	36,09	0,78	37,39	0,98	39,57
0,19	42,61	0,39	36,52	0,59	36,52	0,79	37,39	0,99	39,57
								1,00	40,00