# Dynamic Automatic Noisy Speech Recognition System (DANSR)

# Dynamische Automatische Verrauschte Spracherkennung

Vom Fachbereich Elektrotechnik und Informationstechnik
der Technischen Universität Kaiserslautern
zur Verleihung des akademischen Grades
Doktorin der Ingenieurwissenschaften (Dr.-Ing.)
genehmigte Dissertation

von
M. Sc. Sheuli Paul

D 386

|  |  |
|---|---|
| Tag der mündlichen Prüfung: | 26.02.2014 |
| Dekan des Fachbereichs: | Prof. Dr.-Ing. Hans D. Schotten |
| Vorsitzender der Prüfungskommission: | Prof. Dr.-Ing. habil. Norbert Wehn |
| Berichterstatter: | Prof. Dr. Michael M. Richter |
| Berichterstatter: | Prof. Dr.-Ing. Steven Liu |

I humbly dedicate this thesis to Professor MMR. Without his firm supports, continual inspirations and vivid guidance, this thesis would have not been initiated and would have not been to this stage.

# Acknowledgements

I like to acknowledge Professor Willi Freeden's active participations to help me to initiate this research studies and I am very thankful for this. I like to express my sincere appreciations and gratefulness to Professor Norbert Wehn for his active support for ICA conference and being very supportive to bring this studies to an end. This has an enormous value to me.

I am very thankful to Professor Steven Liu. He accepted me as a doctoral student and gave me the opportunity and supports to work in his group. It has a special significance to me. He gave me advice to apply state space models for the noise reduction. He also provided me the connection to the Zöller-Kipper company in Mainz where I could perform experiments.

I am very thankful to Professor Maurice Charbit for his instant and continuous valuable advice in my work. This has been a huge encouragement to me. I am thankful to Professor Alexander Potchinkov for allowing me use his audio recorder for my data collection. I like to thank Professor Volker Michel for his encouragement and support. I am thankful to all my colleagues at LRS for their cooperation. I like to thank Dr. Heiko Hengen for his advice. I also thank to Ullrich Stadt to help me to collect data from his comparny called MM packaging.

Here I would like to express my gratitude and respect to my elder brother Bijoy Paul for being always supportive. This has been a great help to work for my thesis in a recreational environment without feeling much pressure. I would like to express my sincere respect and thanks to my very loving and affectionate parents. Their ethical sense and moral values are strengths of mine which inspire me to do my tasks enthusiastically applying my best efforts. I am grateful to my all family members for their continuous supports and affections. All these supports and affections help me to do my tasks according to my wish and will.

All these have been possible, I am here at this stage and I am able to make this much progress throughout the studies and on the thesis because of my respected Professor Michael Richter's continuous guidance, encouragements and suggestions. These have been a firm support to do my thesis work in a positive and confident manner. I like to acknowledge and to express my humble thanks to Professor Richter's enormous supports and his vital inspiration. I have developed a dynamic work ethic for my thesis because of his continuous guidance. I have also overcome all the barriers during my studies because of his supports and help.

# Abstract

In this thesis we studied and investigated a very common but a long existing noise problem and we provided a solution to this problem. The task is to deal with different types of noise that occur simultaneously and which we call hybrid. Although there are individual solutions for specific types one cannot simply combine them because each solution affects the whole speech. We developed an automatic speech recognition system DANSR ( Dynamic Automatic Noisy Speech Recognition System) for hybrid noisy environmental noise. For this we had to study all of speech starting from the production of sounds until their recognition. Central elements are the feature vectors on which pay much attention. As an additional effect we worked on the production of quantities for psychoacoustic speech elements.

The thesis has four parts: 1) The first part we give an introduction. The chapter 2 and 3 give an overview over speech generation and recognition when machines are used. Also noise is considered. 2) In the second part we describe our general system for speech recognition in a noisy environment. This is contained in the chapters 4-10. In chapter 4 we deal with data preparation. Chapter 5 is concerned with very strong noise and its modeling using Poisson distribution. In the chapters 5-8 we deal with parameter based modeling. Chapter 7 is concerned with autoregressive methods in relation to the vocal tract. In the chapters 8 and 9 we discuss linear prediction and its parameters. Chapter 9 is also concerned with quadratic errors, the decomposition into sub-bands and the use of Kalman filters for non-stationary colored noise in chapter 10. There one finds classical approaches as long we have used and modified them. This includes covariance mehods, the method of Burg and others. 3) The third part deals firstly with psychoacoustic questions. We look at quantitative magnitudes that describe them. This has serious consequences for the perception models. For hearing we use different scales and filters. In the center of

the chapters 12 and 13 one finds the features and their extraction. The fearures are the only elements that contain information for further use. We consider here Cepstrum features and Mel frequency cepstral coefficients(MFCC), shift invariant local trigonometric transformed (SILTT), linear predictive coefficients (LPC), linear predictive cepstral coefficients (LPCC), perceptual linear predictive (PLP) cepstral coefficients. In chapter 13 we present our extraction methods in DANSR and how they use window techniques And discrete cosine transform (DCT-IV) as well as their inverses. 4) The fourth part considers classification and the ultimate speech recognition. Here we use the hidden Markov model (HMM) for describing the speech process and the Gaussian mixture model (GMM) for the acoustic modelling. For the recognition we use forward algorithm, the Viterbi search and the Baum-Welch algorithm. We also draw the connection to dynamic time warping (DTW). In the rest we show experimental results and conclusions.

# Contents

# List of Figures

# Chapter 1

# Introduction

This chapter contains a general discussion of the whole thesis. It deals with the speech which is a natural communication form. A substantial amount of different views and a huge variety of aspects are inherent in the communication while using the speech. Obviously at a technical level, this makes the speech analysis an interesting and a difficult task.

The speech recognition is a technology that receives and also reconstructs speech on the machine. For this the human speech recognition approach is closely replicated. The main goal of this thesis is in short an automatic speech recognition (ASR) in a difficult environment. By difficult we mean simply that there are various kinds of noise.

This occurs in many practical situations and leads to several technical problems. Our environment is a technical factory where people give commands to a machine that are executed automatically. The state of the art of this investigation is probabilistic. Particularly a pattern recognition method namely the Hidden Markov Model (HMM) is used in order to find the most likely answer to the pattern recognition problem. We deal with a very large dimensional space. For instance, the analog speech waveform is first captured by some transducers. A common type of transducer is a microphone to capture the speech waveform. The analog speech waveform is digitized for its processing in the computer. Suppose, the digitized signal has 90000 samples at 48 KHz sampling rate. These samples are then processed into short blocks which has a length for example 10 to 30 milli seconds (msec), these are then used to extract features by feature extraction technique for dimensionality reduction. These features are classified and modeled by a Gaussian mixture model. In each class, the features contain information for the corresponding class, these are then recognized by the techniques such as forward algorithm, Viterbi algoirhm and Baum-Welch algorithm used in the HMM in order to obtain the most likely result. The probabilistic speech recognition approach is most commonly used practical and commercial applications.

An additional topic is to understand psychoacoustic elements. Such elements contain information that cannot be easily expressed in a written form or in words. Examples are pauses or intonation; they can change the meaning of the spoken words significantly. We are interested in extracting quantitative magnitudes that are used in the speech. This is closely related to the techniques we developed for dealing with the noise.

The speech signal contains information at many different levels such as informational aspects, for example semantic, perceptive and syntactic information and also an information about the speaker. All these information influences recognition and understanding of speech. There are many other external factors that impact the speech recognition. One such dominant factor is environmental noise. A rough distinction between the noises is that they can be extreme, soft or steady and unsteady time varying. Such a scenario can be obtained e.g. when machines, radios, and human speeches interact. This study focuses on recognizing speech in the presence of the environmental noise. We consider a very general kind of noise that, however, occurs in many practical situations. It has been studied rarely in a general way with very little or no success at all.

There exists plenty of studies in speech research on the noise problem. Even each of these approaches is a unique. The aim is usually the same. Here we stress on the research studies that considered the noisy speech recognition only. Most approaches solve the noise problem by enhancing the noisy speech features. Combinations of different solution techniques in order to enhance the noisy speech features or mapping the features prior to recognition; this has been investigated over the decades. There the most common solution approaches are support vector machine (SVM), blind source separation (BSS) in combination with Kalman filters, independent component analysis (ICA) in combination with Wiener filter, neural network, code book mapping, model adaptation, cepstral mean subtraction (CMC), warped filter-bank, Gaussian mixture model and hidden Markov model [46], [45], [117],[73],[86],[148],[146],[77].

## 1.1 The DANSR Approach

Our results are contained in a system called DANSR (Dynamic Automatic Noisy Speech Recognition System). This gave the title to the whole thesis.

In the thesis one sees contributions from a combination of two views:

- In the users view one sees more increased possibilities for recognizing speech, in particular in the presence of environmental complex noise.
- From the structural and methodological view one observes that the system provides an integrated approach of several and partially innovative methods in a complete system. For this purpose we had to discuss the whole recognition system. It can be a starting point for future research too.

The speech signal analysis is based on the discrete time. We have used the discrete time speech samples of the real world continuous time speech sounds. The purpose of analyzing the speech signal for its machine recognition is to reconstruct the speech signal in the machine. Moreover, if the information of the signal can be restricted to a certain limit, then the signal is band limited. According to Nyquist theorem, a band limited signal can be reconstructed from its discrete time samples if the sampling rate of the signal is higher than twice their highest frequency [20]. The bandwidth of the speech signal is 200 Hz to 3500 Hz and most speech energy lies at 7 kilo Hertz (kHz).

The vocabulary used in this study is not arbitrary. We have a list of some predefined small commands that are used by the speaker. In the terminology of artificial intelligence this establishes a closed world because the situation is precisely defined (although very complex). We assume that we have a single microphone for reception only. This is termed as a single-channel reception.

The state of the art of our ASR problem solution approach is probabilistic: In principle we take a Hidden Markov model (HMM). For explaining our work we shortly touch prior achievements. The noisy speech recognition is considered in [17]. The focus is on the feature enhancement in order to recognize speech. The main difference of our approach and the literature in [17] is: We focus on very different noise types taking place simultaneously in a hybrid industrial noisy speech and classify the noises for their treatments. Actually we are being specific about the noise types and provide a solution accordingly for the hybrid industrial noisy speech. Because of such differences we cannot restrict ourselves to one method only. Instead we have to use several approaches and in addition the order of using them is relevant. The Vector Taylor Series (VTS) compensation in combination with Mel frequency cepstral coefficients (MFCC) feature extraction and HTK for noisy speech recognition is used. The noise is additive and it is considered as white Gaussian noise. This has been applied to noisy speech databases in a car and in a room[107]. The hidden Markov model toolkit (HTK) is a speech recognition development toolkit which uses the probabilistic approach namely Hidden Markov Model (HMM) for the speech recognition [128]. This also focuses on the speech feature enhancement first. First order cepstral normalization (FOCN) and minimax normalization are used to enhance the speech features in order to recognize the speech which is assumed to be corrupted by an additive noise using the Baum-Welch algorithm which is used in the HMM based speech recognition for learning [123]. For the recognition of noisy speech, linear prediction coefficients (LPC) cepstral features are used for the multilayer perceptron (MLP) classification and recognition that are investigated in [71]. The noisy speech is used for suppressing the noises using minimum mean square error (MMSE) optimization criterion and multi layer perceptron neural network for recognition in [93]. At this stage, we have not investigated the performance of the MLP or Neural Network (NN) for the recognition. The voice commands in thai speech are recognized in a quiet room, in an office room and in a noisy room for a Radio controlled (RC) car in [104]. This transforms the voice commands to digital signals and then this is converted to a radio active wave commands which are later recognized by HMM based recognition system using HTK tool. We focus on the speech enhancement by reducing the noise and speech feature enhancement by our extended perceptual feature extraction technique called perceptual adaptive local trigonometric transformation (APLTT). We have applied there the perceptual entropy (PE) instead the best basis spectral entropy that exists in the SILTT. The perceptual entropy is useful for de-noising speech [57].

The term ”dynamic” for our dynamic automatic noisy speech recognition (DANSR) has in our studies a number of particular properties:

- Firstly the speech has a relation to its past occurrence, it is not memoryless and it is dynamic in this sense. For example, if we would like to say ”Open the door”. Relating to this expression in this example, saying only ”door” makes no sense considering the semantics of the original intension or only saying ”d” for the ”door” also makes no sense.

- Secondly the research study is based on small but varying spoken commands and this is reconstructed as well as recognized on the computer. This is an online lively approach. Thus we say the system is dynamic.

- Thirdly we use a dynamic programming approach to attain to our solution of the problem. The dynamic programming approach includes [112]:
  - Recursive approaches for the optimal result.
  - The main solution requires a solution to the sub problems i.e. the problem is divided into sub problems in order to find the problem solution.
  - The solutions of the sub problems are based on the solutions of previous problems. The solution of the problem is inter-dependent. This means each output of each approach is used as input to the next approach.

An essential element for speech recognition is provided by the features. Short feature vectors are easier to handle by the actual recognition algorithms than long signal sequences. However, they should still contain the whole information contained in the speech what makes extraction very difficult. With respect to the feature extraction stage in the speech recognition studies, mostly Mel frequency cepstral coefficients (MFCC), linear predictive cepstral coefficients (LPCC), perceptual linear predictive (PLP) cepstral coefficients are considered. In MFCC and PLP the signal decomposition and spectral analysis are followed by the process of the lapped transformation where the FFT is applied. The problem of abrupt discontinuity is present although it is reduced because of the lapped transformation. There also exists the non-standard shift invariant local trigonometric transformed (SILTT) features based on the local trigonometric transformation (LTT) approach. But the feature extractions in SILTT do not make use of all available information provided by the speech. Another problem of the SILTT transformation is that the perceptual feature extraction is not possible and it does not provide perceptual speech features. The SILTT has been used for speech processing and speech recognition in [95], [32], [22]. There a perceptual mapping is not used while it is used in our speech processing and speech recognition.

We handled the discontinuity problem in MFCC, LPCC, PLP by applying a local trigonometric transformation followed by a lapped transformation and took extra care to the application of a folding operation. The discontinuity is smoothened here better than using the traditional MFCC and PLP. In earlier research, the adapted local trigonometric transformation is used in the vector quantization (VQ) based HMM speech recognition [22]. There the signal is decomposed signal into M uniform-subbands to each subinterval. The energy of each sub-band is used as speech features. These features are applied to VQ and HMM. Here we used a continuous classification model, i.e. the GMM for speech recognition tool HMM and we integrated this with APLTT.

Occurrence of psychoacoustics elements in speech is very basic and it is common. They express information that one cannot directly express in words. It is not clear in the first place how these elements occur in speech in a quantitative way. We explain how these elements can be captured by a specific feature extraction. We adopted such quantities into the existing LTT approach. These are in particular the psychoacoustic quantities that describe the speech properties that are important for human hearing. In the normal SILTT they are not included. For this inclusion the different quantities and their commputational properties have to be studied and combined.

## 1.2 The Chapters

The thesis has four main parts. The backgrounds, analysis of standard techniques and the techniques used in the DANSR are discussed in each chapter.

- Part A: This part introduces into our topic. This includes chapter 2 and chapter 3.
- Part B: This part explains our noise solutions to hybrid noise problems. The related chapters for this are : Chapter 4, chapter 5, chapter 6, chapter 7, chapter 8, 9, 10.
- Part C: This part introduces basic psychoacoustics quantities for speech perceptions and their adaptation into our approach, the part also describes feature extraction. The included chapters in this section are: Chapter 11, chapter 12, chapter 13.
- Part D: This part talks about classification and recognition. The included chapter in this section is chapter 14.

Now we list the outlines of the chapters individually.

Chapter 2 outlines the speech generation and recognition with respect to a human being and it outlines the speech recognition with respect to a machine and chapter 3 introduces into the methodology we used to enhance the noisy speech for our noisy speech recognition. The general background of our noise treatment is the subject of chapter 3. In chapter 4 we discuss the pre-emphasizing methods. The data preparation is the content of chapter 4. Chapter 5 is devoted to strong noise. Chapter 6 introduces the standard source excitation model. The chapters 7,8, 9 and 10 focus on the parametric speech production model. The autogressive process in the vocal tract model is discussed in chapter 7. Linear prediction and its parameters are handled in chapter 8 and 9. Sub-band coding, spectral minimization, Kalman filtering is the subject of chapeter 10. The psychoacoustics for the thesis is in chapter 11. Feature extraction is handled in chapter 12,13. Chapter 14 is concerned with classification. Chapter 15 shows experimental results including evaluations and chapter 16 gives conclusions.

## 1.3 Thesis Contributions

The innovations of the thesis are two fold, applications and structural contributions. The combined methodological approach developed in the thesis and the sub components of this approach are tested independently and as a whole in a real industrial environment. The system is in applications very practical and serving the purpose and meeting the aim as we intended this. We have provided in thesis our experiments, analysis, evaluations and results that we have done using the real world hybrid noisy industrial data.

Here we list the main contributions of this thesis:

- An integrated hybrid solution approach to an existing environmental hybrid noisy ASR problem.

- A new noisy speech pre-emphasizing approach. Here we modified and extended an existing approach but the existing approach is used for speech silence detection. Our application for this here is for noisy speech pre-emphasizing.

- Strong noise modeled by a Poisson distribution and its treatment by matched filter.

- A new perceptual feature extraction approach. Here an existent adaptive local trigonometric transformation (LTT) mathematical tool is extended. This is already applied to speech processing and recognition. We have extended this adaptive LTT approach to perceptional adaptive LTT (APLTT ) approach and extended this for model based speech recognition system.

- We have applied the Gaussian mixture model (GMM) to model APLTT features for classification and the HMM for recognition. The HMM based speech recognition system is continuous when we apply the GMM.
- Applying the techniques to model the psychoacoustic quantities.

For this purpose we have studied the existing approaches in details and made many experiments with the data collected from the real world on our own and evaluated these with other existing approaches.

# Chapter 2

# Excursion: Human Speech and Machine

**Outline of the chapter** In this chapter we describe the speech from the views of speech generation, perception and its recognition in the real world. Here we discuss how it is done by the human body which we model from an engineering point of view. For this we followed the relevant literature and modified several aspects for our purposes and for simplification.

## 2.1 Excursion:Human and Machine Interaction

The speech is an acoustic signal which is produced by a human speaker as a sound pressure wave and comes out of a speaker's mouth and goes to a listener's ears. The speech is a dynamic and an information bearing signal. The speech is composed of a sequence of sounds that serve as a symbolic representation of a thought that the speaker transmits to the listener. The arrangement of these sounds is governed by some linguistic rules associated with a language. The scientific study of the language and the rules are discussed in linguistic and phonetic studies. The problem is to automate the whole process, i.e., the sound production as well as the sound reception. In this thesis we concentrate on sound reception. We approach the problem by looking into the information content of the speech following engineering types of a technical approach. That is, we are building machines that simulate speech production and speech reconstruction for its recognition in such a way that engineering methods can be applied.

The task of the speech recognition is to find the most likely information given by the speech. The speech is in general viewed as a probabilistic process. For describing a given signal we need a model. The model can be an acoustic, a phonetic or a lexicon or a language based. The language model provides the composition and combination of the words of the speech. The lexicon or phonetic model discusses the fundamental sound formations of the word. The acoustic models can be based on the sound units as e.g. words or phonemes.

Our objective is to make speech interpretable to a machine what the speaker has originally said. This leads to a model based speech processing and then to a dynamic speech recognition system. By dynamic we mean that the system approach is dynamic programming based i.e. we have recursion, tracing, bottom up solution approach and we search for the optimal result.

Next we introduce how the speech is generated and recognized by a human being. We give a general impression and are concerned with technical or medical elements.

## 2.2 Human Speech Generation and Recognition

We show an overview of the speech production and recognition in figure 2.1. In this figure the connection between sending and receiving is described by the right and the left vertical arrows. Figure 2.1 is our simple modification of the speech chain given in [80]. The figure has two parts:

- The upper part : Speech Generation
- The lower part : Speech Recognition

### 2.2.1 Human Speech Generation

We describe the speech generation in brief only because it is not our central task. However, for certain aspects it is necessary for us to know something about the speech generation. Later on in the chapter we describe the vocal tract from an engineering point of view using speech acoustical information. Here we explain the influence of the vocal tract in the speech production and generation. This will clarify the motivation of the use of the vocal tract system in the speech research as a main organ in the practical speech production model for speech processing[30]. The process is described in steps:

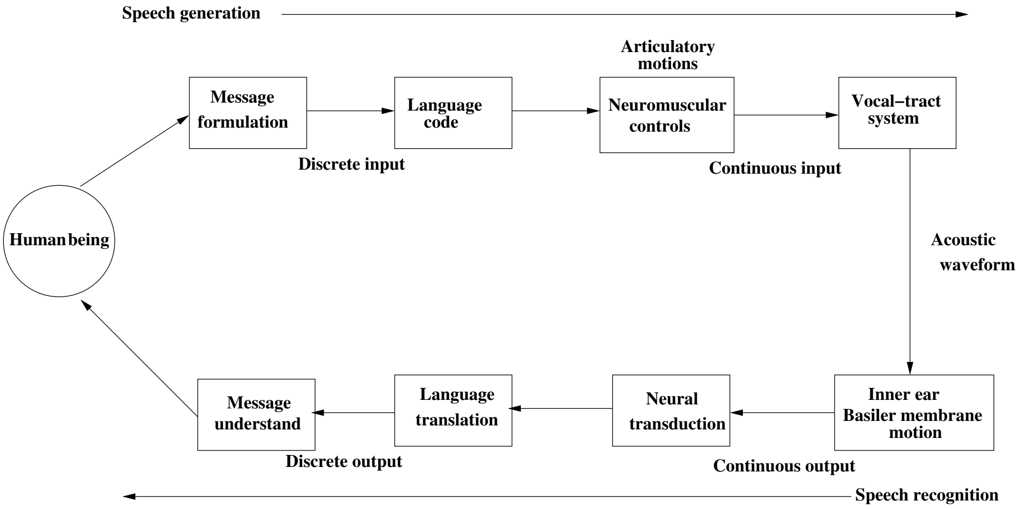


Figure 2.1: Speech Generation and Speech Recognition [80]

- The speech production process begins when the speaker formulates a message in the speaker's mind and in the brain. That is what the speaker wants to say to the listener.

- The next step is the language code. This converts the message into some text and phonetic symbols. These are the elements of a certain language governed by linguistics and phonetic rules. When the phonemes i.e. the acoustics units are in correct order, the speaker can pronounce an understandable word. For example, if the speaker wants to greet someone saying "Good Morning" the speaker first needs to decide the language, e.g., if it is in German or in English. The result of the message formulation or the conversion of this message into a syntactic form is then sent to the neuromuscular controls. The smallest element in the speech is called a phoneme. The phonemes are given as sounds of a language produced by an individual speaker

- Next the neuromuscular movement takes place to control the vocal apparatus, for example the vocal folds, or the nasal part or the lips that are needed to be moved to generate the message for instance here it is the greeting : "Good Morning".

- When the loudness of the pitch is established, the vocal-tract system, for instance the vocal-tract vibration, acts, the speaker can say ”Good Morning”.

- The result of the whole process is a continuous time analog waveform at the lips, jaw, velum etc. In this way the speech waveform is produced.

Here the generation process ends. We will not be concerned with the human speech generation and its automation. However, the generation process has to be understood to use it in the model that represent the spoken speech. The source excitation model discussed in chapter 6 is the most commonly used model to represent the speech generation process. The purpose of using this model is given in chapter 10 but some physical explanations of this model relating speech generation by the human being is given in section 2.4.

The vibration rates of the vocal folds during the speech production while transmitting the process through the vocal-tract are different. Similarly, the speech waveform of formulating the message may change depending on the speaker.

### 2.2.2 Human Speech Recognition

For recognition we look at figure 2.1 to see what is intended.

The message interpretation shown at the bottom right corner in figure 2.1 is the speech recognition or the speech perception.

- The first step is an effective conversion of the acoustic waveform into a spectral representation. This takes place in the inner ear by the basilar membrane. This membrane acts as a non-uniform spectrum analyzer by spatially separating the spectral components of the incoming speech signals and analyzing them, for example using a non-uniform filter bank.

- The next step is the neural transduction of the spectral features into a set of distinctive sound features.

- These features are decoded and processed by the brain. This is described by linguistic rules.

- Finally these features are converted to a set of phonemes, word sequences and sentences in order to understand or recognize the intended message (that was originally generated). This takes place in the brain and requires much human knowledge.

## 2.3 Speech Recognition by Machine

Here we give a pictorial motivation for the approach. Figure 2.2 shows how the speech recognition can be replaced by a machine. If we compare figure 2.1 and figure 2.2, we see in figure 2.2 that the tasks in the human ear and the brain is replaced by a machine. A basic understanding of the processes taking place in the human ear is useful for the speech recognition tasks. We see that these tasks are rather complex and require interdisciplinary knowledge, e.g. from the physics of sound transmissions, the physiology of the human auditory system, the human speech perception to begin with. We have given in chapter 11 a brief superficial introduction and outline of the study of the psychoacoustics that studies the human speech perception in order to capture an essence of human speech perception and perceptual speech recognition. These are introductions and outlines of the study of the psychoacoustics. We show how the psychoacoustics elements are quantized. Our main aim is to recognize noisy speech and this is discussed in next chapter 3. The ASR studies belong to an area of pattern recognition which is to some degree a sub-area of machine learning. In the overview of an automatic speech recognition (ASR) system given in figure 2.3, we see the speech data as input are transformed into some trained set in order to apply some learning tool in the training phase. Then some test data are used by applying some search tool in the testing phase. Therefore the ASR system has two main phases:

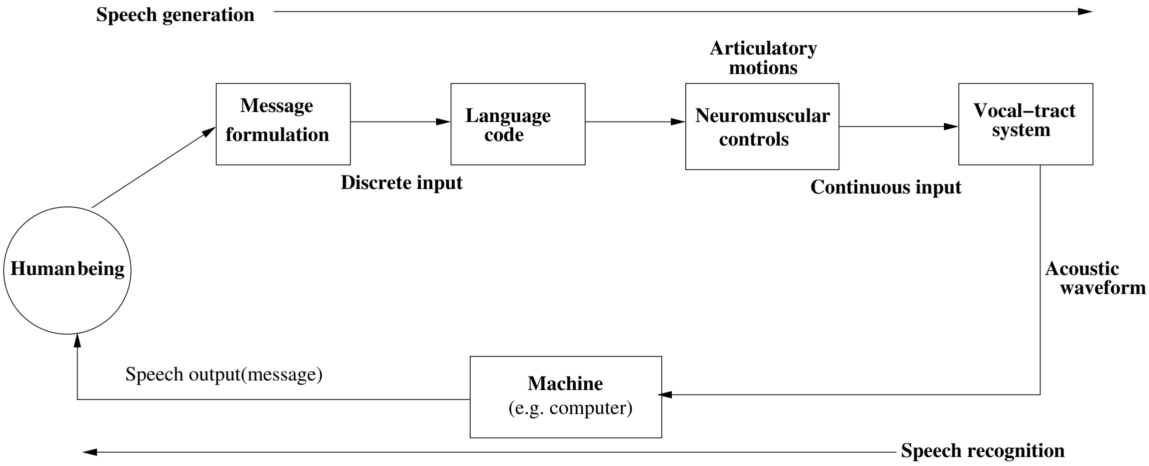


Figure 2.2: Human Speech Generation and Machine for the Speech Recognition

- Training Phase: Here the examples are given to machine learning.

- Testing Phase: Here some learning tool is employed and then classification of the examples in order to recognize the test data to find the overall the outcome as a result of the learning process takes place.

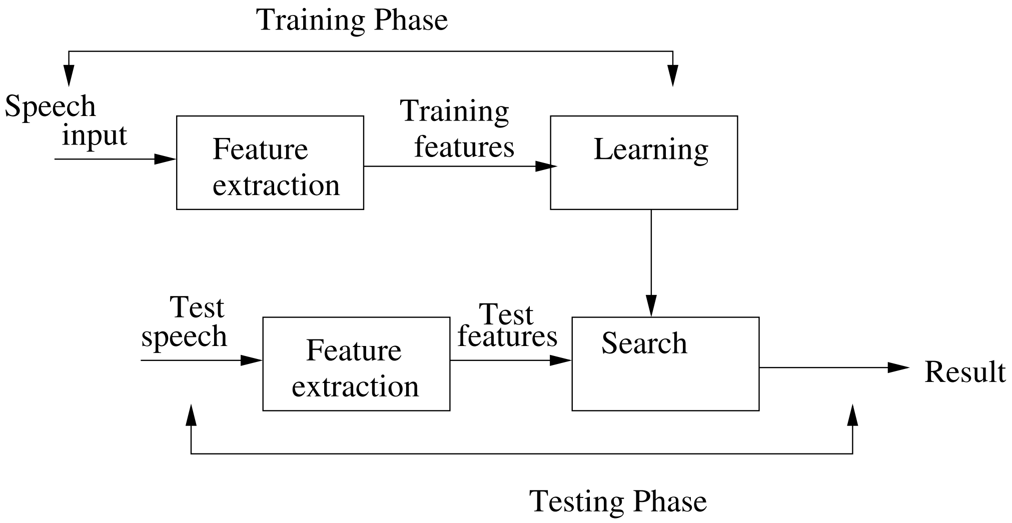


Figure 2.3: Overview of ASR Process

There are different ways the generated speech can be represented in the machine. Two most common approaches are:

- Parametric approach: Here some signal models are used to extract speech parameters. An example of the parametric approach is linear prediction analysis (LPC). They are individual for each speech act, unkown and have to be estimated. These parameters are the starting point for the speech recognition task. We followed this approach. This is discussed in chapters 6,7,8,9.

- Non-parametric approach: FFT based analysis is an example of this approach. This is a commonly used tool to begin the speech recognition tasks. Examples of the non-parametric approach is MFCC discussed in chapter 12.

The ASR architecture and structures are now briefly mentioned. An overview is shown in figure ??.

### 2.3.1 ASR Types

The speech recognition can be of different types. Thus the architecture and structure of the ASR can be varied. Below we provided a list of possible ASR types and their architecture [69], [80].

**System Architecture** This discusses acoustic and linguistic elements as e.g. phonemes, words, phrases and sentences. The structure of the ASR can be:

- Continuous: Speech that is naturally spoken in a sentence.
- Discrete: Discrete speech systems use one word at a time and it is useful for people having difficulties in forming complete phrases in one utterance.
- Isolated: In isolated speech, single words are used and it is easier to recognize the speech.

The type of the ASR can be :

- Speaker Dependent : A speaker dependent system is intended for a use by a single speaker. In a speaker dependent system, necessary training data are : 100 different people saying the speech for instance 10 times separately and necessary testing data: 25 different individuals that are not in the list of the speakers in the training data collection saying the corresponding speech.
- Speaker Independent: A speaker independent system is intended for use by any speaker; it is more difficult in the sense that it has more variations to be considered than the speaker dependent one. The speaker independent system involves a collection of thousands of data.

The vocabulary size of the ASR can be:

- Small Vocabulary: Tens of words for example a list of 10 to 100 vocabulary model.
- Medium Vocabulary: Hundreds of words for example a list of 100-300 vocabulary model.
- Large Vocabulary: Thousands of words for example a list of 1000- 10000 or more vocabulary.

Some ASR applications and possible environment are :

- Examples are speech in a hospital or in a nursing home to monitor the patients, in an industry to command a machine, for dictating in law enforcement, in robotics to perform some intended tasks using some voice commands etc.

- Environment: This can be noisy, moderate, mixed of noise and normal environment or quiet.

Our DANSR specification is mentioned in chapter 3.

The human speech generation process is captured for speech processing by the source excitation model.

## 2.4 Acoustics of Speech Production Model

The acoustic phonetics studies the acoustic properties of the speech and how these are related to the human speech production. A standard computational speech production model is discussed in chapter 6 which makes use the study of the acoustics, phonetics, psychoacoustics and digital signal processing in order to model the speech process.

The purpose of the computational speech production model is to manipulate the reality computationally and to estimate the constraints and the constants in the body. This correlates the physical process to a computational model for the processing.The constraints in this context are the natural regulations in generating the human speech and the constants are the weights or the speech parameters and the outputs.

Next we present computational aspects about some basic components used in the model in an overview. They are concerned with both, the human body and the machine. The model is described in chapter 6.

The vocal-tract is playing a vital role in the speech production discussed in chapter 6. In the next description we use partially some qualitative terms.

The vocal-tract shown in figure 2.4 is considered a lossless acoustical tube. We see in the figure that the vocal-tract has different cross sectional areas denoted by $A_1, A_2, \cdots, A_5$.

### 2.4.1 Resonant Frequency, Formant and Sampling Rate

The resonant frequency of the vocal tract tube is the peak frequency of the vocal tract tube. It happens when the particular frequency and the vocal tract

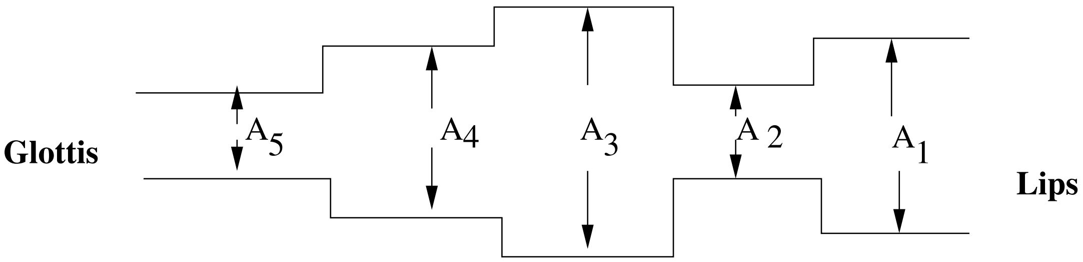


Figure 2.4: Sketch of the vocal-tract: Non-uniform cross-sectional area [118]

frequency coincide. The standing wave is the current wave in the vocal tract tube. These are intuitive simple definitions of the resonant frequency and standing or ongoing wave of the vocal tract. The details of this can be found in the area of the acoustics phonetics study which is not investigated further.

**Resonant Frequency and Formant** A connection between the formant and vocal tract tube is shown in equation (2.2).

The length $l$ of the vocal tract tube is an odd multiple of $\lambda$ which is the wavelength of the standing wave. $\lambda_n$ indicates the wave length at $n$ which is a positive integer number.

$$l = \frac{(2n-1)\lambda_n}{4} \tag{2.1}$$

The length of an adult male vocal tract is approximately 17.7 cm long and the length of an adult female vocal tract is approximately 14.75 cm long [87].

The resonant frequencies $f_n$ for $n = 1, 2, 3 \cdots$ are shown in equation (2.2). The speed of the sound in the air denoted by $c$ is assumed to be 35400cm/sec. The formants are defined by the spectral peak in the speech sound spectrum. They are determined by the resonance frequency. This means if the resonant frequency is 1500 Hz, then a formant is generated at 1500 Hz.

$$f_n = \frac{c}{\lambda_n} = \frac{(2n-1)c}{4l} \tag{2.2}$$

**Selection of Sampling Rate** The relation between the sampling rate selection and the vocal tract architecture is shown in equation (2.4). The wave propagates in vocal tract cross-sections. In figure 2.4, the cross-sections of the vocal tract are denoted by $A_1, A_2, \cdots, A_n$ and $n$ denotes a positive integer number. Suppose, the length of each tube is $j$, then the wave propagates in each section is $\tau$ is computed by equation (2.3).

$$\tau = \frac{j}{c} \tag{2.3}$$

The discrete time system, the sampling rate of the vocal tract is $2\tau$ seconds. The sampling rate $f_s$ is then can be expressed by equation (2.4) [30].

$$f_s = \frac{1}{2\tau} = \frac{c}{2j} \tag{2.4}$$

The order of the model relating this to the physical vocal tract tube is discussed in chapter 8 in section 8.1.1.

### 2.4.2 Reflection Coefficients

As we mentioned earlier the vocal-tract is an acoustical tube which has several non-uniform sections. The waves that propagate from the tube are partially reflected and partially interrupted by the discontinuities of the junctions of the tubes. This is described by the reflection coefficients. The reflection coefficients reflects the vocal-tract structure, the shape of the vocal-tract and the speech transmission that is taking place in the acoustical vocal tube. The 0 value of the reflection coefficient means that all transmission in the vocal-tract tube are passed and 1 value of this reflection coefficients indicate that the transmissions are reflected [81], [87].

The reflection coefficients between two sections of the vocal tract can be shown by equation (2.5). The reflection coefficients are denoted by $\kappa$. $\kappa_i$ is denotes the reflection coefficients for $i = 1, 2, \cdots, p$. $A_i$ and $A_{i+1}$ are the cross sections of the vocal tract tube where $1 \leq i \leq p$. There are $p$ many tube sections. $A_0 = \infty$ is the area of the space beyond the lips and therefore it is a lossless transmission.

$$\kappa_i = \frac{A_i - A_{i+1}}{A_i + A_{i+1}} \quad \text{where} \quad |\kappa_i| \leq 1 \tag{2.5}$$

The length of the vocal tract tube is determined by the sampling period and the speed of the sound as discussed in section 2.4.1. The reflections cause spectral shaping of the excitation which acts as a digital filter with the order of the system equal to the number of tube boundaries. The digital filter can be realized by a lattice structure. In this structure, reflection coefficients are used as weights. This is the background of the reflection coefficients and its use in the lattice structured filter. This is briefly discussed in chapter 9 in section 9.3.1. The details of this can be found in [24].

## 2.5 Categories of Speech Excitation

From the speech acoustics point of view, an excitation type can be categorized by the following kinds of speech sounds [69].

- Voiced (Example: The letter /I/ sound in the utterance of ”six”)
- Unvoiced (Example: The letter /s/ sound in ”six”)
- Mixed (Example: The sound corresponding to the letter ”z” in the phrase ”three Zebras”)
- Plosive (Example: A short region or silence, followed by a region of the voiced speech, the unvoiced speech, or both. A plosive example (silence + unvoiced) is the sound corresponding to /t/ in ”pat”. Another (silence + unvoiced) in the /b/ in ”boot” )
- Whisper is the pressure in the glottal area to utter any excitation types.

We will see in chapter 6 how the above mentioned excitations types are simplified to the voiced and the unvoiced types and how these two types are modeled by only a single simple computational model to reflect the speech production process in reality.

# Chapter 3

# Noisy Speech Recognition

**Outline** In this chapter we talk about noisy speech and its definition and handling this in our studies. We explain our aim, problems, challenges and difficulties relating these studies to the real world. We introduce the hybrid noise and their treatments. This incorporates different kinds of noises. Our solution approach takes care of this. This approach is mixed with a preview of literature about noisy speech evaluation and our own methodological approach. We introduce both active and passive noise solutions to this problem.

For our approach an industrial environment is selected as an application area. What is new here to our perspective is that we provide a hybrid solution to our problem and the actions we take in order to arrive at the solution.

In chapter 2, we provided a simple realization of speech generation, speech recognition by the human being and also a scenario of speech recognition by a machine. Here we talk about noisy speech recognition by a machine. The speech is in the first place not noisy by itself and it is noisy generally only after its generation by environmental factors.

Next we first introduce to our noisy speech, hybrid noise, their impacts in section 3.1.

## 3.1 General Aspects

In general, an industrial environment is noisy. Here we are talking about a noisy industrial environment which is equipped with different types of machines, machinery handling and their operations such as manufacturing, assembling. The next figure 3.1 is not intended to be a definition. The corresponding definitions are complex and come later when we discuss the technical aspects. 3.1 is rather thought of an illustration so that one can see in principle what we want to do. In figure, 3.1, we see that spoken commands are generated by a human being in a noisy industrial environment and given to a machine i.e. a computer for its recognition in the same environment. If we compare figure 2.2 given in chapter 2 and figure 3.1, we see the difference between the two figures. In figure 2.2, speech is generated in a clean environment but in figure 3.1 speech is generated in a noisy environment. In figure 3.1, the speech generated by a human being is delivered to a noisy environment. The speech is corrupted by the environmental noise. The industrial noises are not all the same type. They have different intensity and extremity and we call this combination hybrid. Typically, we categorize them as strong, steady and mild. A reality is that we cannot process our data that we have collected from the noisy environment for its required enhancement and neither do we have an option to enhance the noisy observations by some standard noise reduction techniques. A main problem is that a "common" approach or a "standard" approach is not an appropriate solution approach to this hybrid noise. There is not yet any such solutions to the hybrid noise that could enhance the hybrid noisy speech for its recognition. Nevertheless, our scenario occurs quite often, see section 3.2. Though there is a huge amount of literature about noisy speech enhancement or noise reduction or removal [117], a majority of this [148] solves this problem by applying some standard noise solution approaches or standard digital filtering or some adaptive filtering such as Wiener filter, Kalman

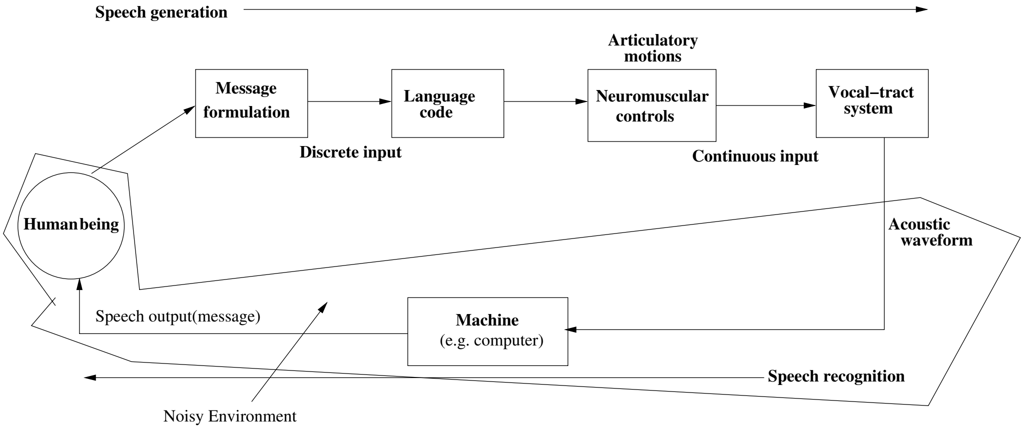


Figure 3.1: Noisy industrial environment: Speech Generation by a Human being and a Machine for the Speech Recognition

filtering for white noise, or spectral subtraction, or sometimes a combination of one or more of them [86]. In fact, a hybrid noise solution has rarely been considered. The problem seems to be that it is not trivial to combine different techniques [46]. But the situation is that because of the different types of noise we need a hybrid treatment for them. For each type we apply a method based on the existing noise source. Each method will, however, not just remove or add something but will effect the whole signal. Here the main tasks for our noisy speech recognition problem are:

- Removal or reduction of the noise that corrupts the speech. We use the removal or reduce the noise because both of these are done. The removal or reduce term is dependent on the type of the noise.
- Recognition of the enhanced speech.

## 3.2 Scenario

In an industrial environment, a smooth communication is not possible and the necessity of removal or reduction of the noise in the desired speech becomes significant for an effective communication.

There the noise is mixed and originating from different sources. These come for example from lifter systems and related machines or different types of conversations among people. We term **strong noise** as a sudden burst, press or dropping sound originating from various heavy material handling and falling down. The duration of this type of noise is very short. The time-varying **steady-unsteady noise** in our description is originating from varying electromechanical machines. We consider the remaining background noise as **mild noise**. A precise duration and formulation of the mixed noise from various sources is not possible in this hybrid noisy environment and precise mathematical definitions cannot be given. Hence we use qualitative arguments. Here we consider the noises that affect the commands at its duration which is in our case no longer than three seconds. It is not always possible to maintain an exact timing.

The scenario of this studies is shown in figure 3.2 in an overview over the whole situation. Figure 3.2 is again only of an illustrative character, as common in artificial intelligence. There are elements of the speech, the noise and the system shown in a combined way to inform the user. We have a predefined command list. The environment is a closed world because the situation is precisely defined.

The task is to recognize the delivered speech in spite of the existing environment. Figure 3.2 shows the different inputs to the recognition system. The inputs can be desired spoken command, undesired different types of signals such as noise, the different types of environmental impacts. But the aim is that the recognition system recognizes the desired spoken command and omits the other undesired environmental influences. To fulfill the aim of the tasks, we have used different techniques and integrated them. However, the integration is somewhat different than in ordinary software systems. We have no modules where we just have to take care of input-output relations. Each technique concerns more or less with almost the whole system. We have to take care that certain properties of the system still hold and the system is interactive. Therefore the integration of the different techniques have to be embedded in such a way that an immediate interaction between the techniques applied to perform the tasks are possible.

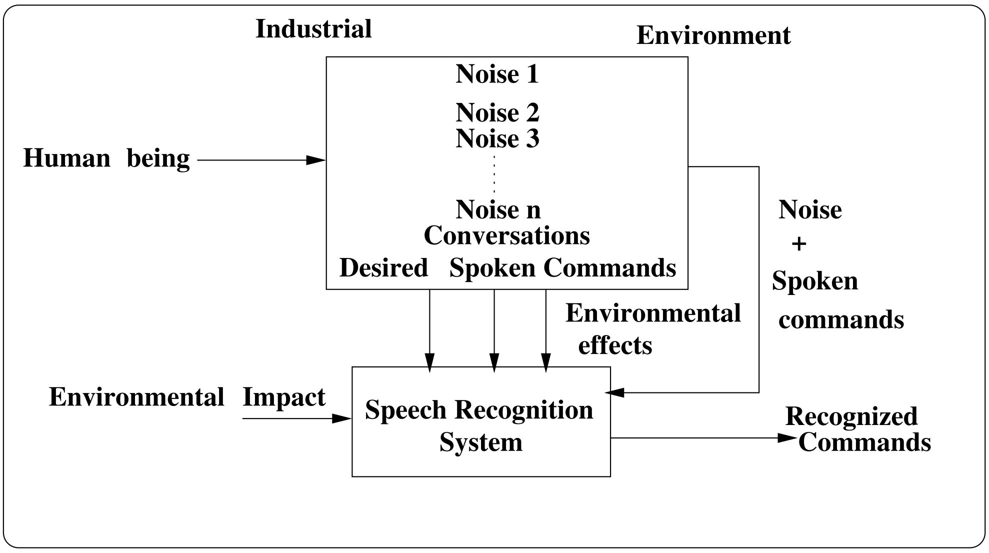


Figure 3.2: Hybrid Noise and Industrial Environment

### 3.2.1 Goals of DANSR

We focus on developing a small vocabulary speech recognition system. The small vocabulary speech is a set of small spoken words which we interpret as commands. This set of small words is spoken to a single microphone. The speech sound is a mixed tonal sound and it has a variety of variable patterns. The variability we want to preserve is the speech acoustic information on the word level. For this we have a followed mainly the parametric modeling. We look first at the vocal tract configuration by a parametric model and use this model for noise reduction in order to obtain an enhanced speech, then we use the enhanced speech for a non-parametric spectral analysis and a perceptual speech feature extraction technique in order to obtain the features and finally we apply a model based pattern recognition technique for classifying the features and recognizing them. Thus the goal of the DANSR is :

- An integrated approach to deal with the different types of noise simultaneously by the followings:
    1. A suitable combination of mixed noise reduction approaches.
    2. Extraction of perceptual speech features of the enhanced speech.
    3. Pattern recognition techniques applying the Hidden Markov Model (HMM) which model is based on the Gaussian mixture model (GMM).

## 3.3 Noisy Speech and Difficulties

In our day to day life, we can not interpret or if we do not understand a speech of a speaker in case of an extreme strong noisy situation, we ask the speaker to repeat. The question is what not understanding means; there is no general definition. If the listener is a human then this is personal. A machine however needs a formal definition. We circumvent this problem by deleting the speech depending on the noise definition, see chapter 5. In an acoustic sense the sound or speech or noise is an atmospheric pressure waveform where its variation as it progresses and its differentiation is received subjectively. This means some sound or speech or noise may be perceived in different meanings from a person or may vary from theme to theme. For example loud music may be noisy to an individual or some conversation may be noisy to an individual but for others this may sound useful or not so influential. Regardless of an environmental influence, the aim is to recognize some predefined spoken command.

We need to record as much variability as needed. We mention the necessary amount in chapter 4 in section 4.1. The speech sound is a stochastic process and variability is one of the major difficulties of this process.

The success of an ASR system requires knowledge from multi-discipline areas such as electrical engineering that discusses signal processing, communication and transmission, physics related to psychoacoustics, linguistics for example phonology, computer science as for instance pattern recognition, searching, logic etc. An individual can hardly attain all the required knowledge. Therefore, one has sometimes in practice a group research where the tasks can be sub-divided based on an individual's expert knowledge. Sometimes the expense to continue this research is not available at the spot. Therefore, in many cases the success of the research may not be fully achievable. These are certainly some common challenges in this research. Below we talk about our ASR research problems and challenges.

### 3.3.1 Challenges

The speech signal has a complex pattern. It is mixed with different tones and varies with time. The speech signal has different frequencies and different intensities. The complex tonal sound has more processing complexities than a pure simple sinusoidal tonal sound. The variability of the speech signal makes the speech research complicated and challenging. A spoken command generated by a particular speaker several times is not the same. The utterances all have different frequencies and different intensities. Many factors are involved such as time, speaker, speaking style. Moreover, in our study, we have noisy speech to process for its recognition. The noise is originated from the background, from the environment. The environment is mixed with different kinds of noise. We have listed some common challenges of the speech recognition problem:

- Variability of the speech due to a variety of speeches spoken by the same speaker.
- Variability of the speech due to a variety of speakers speaking style.
- Variability of the speech due to the speech linguistics formation.
- A variety of environmental noise.

### 3.3.2 Difficulties

So far we mentioned the challenges; but what are the difficulties we encounter in the research problem? The human being itself has a difficulty to understand others in a noisy environment. Some additional difficulties we encounter are listed below:

- Identifying the noise in the noisy environment.

- Modeling the hybrid noise.
- Finding a proper solution to remove or reduce the noise that corrupts the spoken word.
- Managing the collected noisy speech data for noise reduction and training. This means we cannot use the data directly for the noise reduction without preparing and pre-emphasizing the collected data. Also, only standard pre-emphasizing which is generally a first order high pass filter is not sufficient to prepare the noisy data for their further processing.

## 3.4 Noise Measurement and Distinction

The measurement of the noisy signal characteristics and the solution to the noisy signal problem can be passive or active or a combination of both. Here we will introduce the active and passive approaches and their application to our studies.

- By ”active” we mean actions such as removal, filtering, Poisson modeling and matched filtering operations, and Kalman filtering.
- By ”passive” we mean some standard measurements of the noisy speech or the noisy sounds.

To distinguish noise, we first use the passive measurements. Then we apply the active approach. The active approaches are introduced here but the details of these are in chapters 4, 5,6,7,8,9,10. For the passive measurements, we first measure the loudness of the noisy speech using a standard A-weighting filter. We also calculate the energy of the noisy signals, compute the probability density function, use a box plot evaluation, and evaluate the noisy signals by computing the signal to noise ratio (SNR). We call these passive measurements because they do not make any changes and they do not improve the noisy speech by removing noises. Rather they give some information about our collected noisy speech.

### 3.4.1 Noise Measuring Filters and Evaluation

Since the situation is taking place in the bounded space i.e. in a factory which is a very spacious room in a building and we consider this as a ambient noise measurement. In such situation, sound levels are measured by sound level measurement devices such as A, B, C, D, E-weighted filters. We choose the A-weighting filter among those device because the A-weighting filter accounts for the human hearing perception that is the human ear sensitivity [91]. This is a useful property for the speech recognition system. The A-weighting filter is also commonly used for the ambient environmental sound level measurement. The A-weighting filter measures the loudness of the average sound level over a period as a root mean squared power in dB. According to the statistical information and the sound level measurement, we have continuous, varying, intermittent, and impulse types of sound. We have a random noise level variation such as mixed noise levels varying from 60 dB to 115 dB. At 90 to 115 dB a communication is not even possible among the human beings. Such noisy environments damage our hearing capability. In such measurements, the range of 70 to 80 dB is in the mild-steady noise level, 80 to 89 dB is in the varying steady-unsteady and above 90 dB is strong noise level [23]. In figure 3.4, we see the sound levels of the noisy signals collected from hybrid noisy industrial environment. We have given the locations in chapter 4. The sound level is measured by A-weighting filter.

#### 3.4.1.1 A-weighting Filter

The A-weighting filter has been used in the frequency domain of the signal. Here we explain how we have used this filter to evaluate the signal level of the noisy signal: First the spectrum of the noisy signal is computed using the discrete Fourier transform (DFT) and the A weighting filter is then applied to measure the signal power in dBA. The power spectrum of the $N$ sampled signal $s[n]$ where $n = 0, 1, 2, \cdots, N-1$ is computed by equation (3.1).

$$S(k) = \sum_{n=0}^{N-1} |s[n]| e^{-\frac{2\pi kn}{N}} \tag{3.1}$$

Equation (3.2) below shows a relation between the frequency response of the A-weighting filter denoted by $\alpha_A(f)$ and the linear frequency of the signal denoted by $f$. The measurements of A-weighting filter give us some intuitive idea about the signal and the sound level. From this we get an approximate idea about the enhanced signal level and use the enhanced signal for the feature extraction. Equation (3.2) is mainly collected from [26] and more information on this is in [135], [23], [91], [51], [49]. We use this formulation to generate the figures 3.4, 3.5, 3.6 using the implementation given in [49] and in [26]. The derivations and formulations of the A-weighting filter is discussed in acoustic noise measurements, monitoring and control and some information about this studies can be found in [23], [91], [51]. The mathematical derivations and modeling of the weight filters are the results of many experiments to measure the loudness of the sound in sound pressure level (SPL) or in dBA [135]. We will not discuss the derivations of the filter here. We only use the filter to get an information about our noisy signals, their sound levels. We have modified the implementations given in [51] and [26] according to our own our measurements; the experimental results are shown in figures 3.4, 3.5 and 3.6. In figure 3.3, we see the noisy signal in time and frequency domain. This is included to get an outlook of the mixed noisy signal.

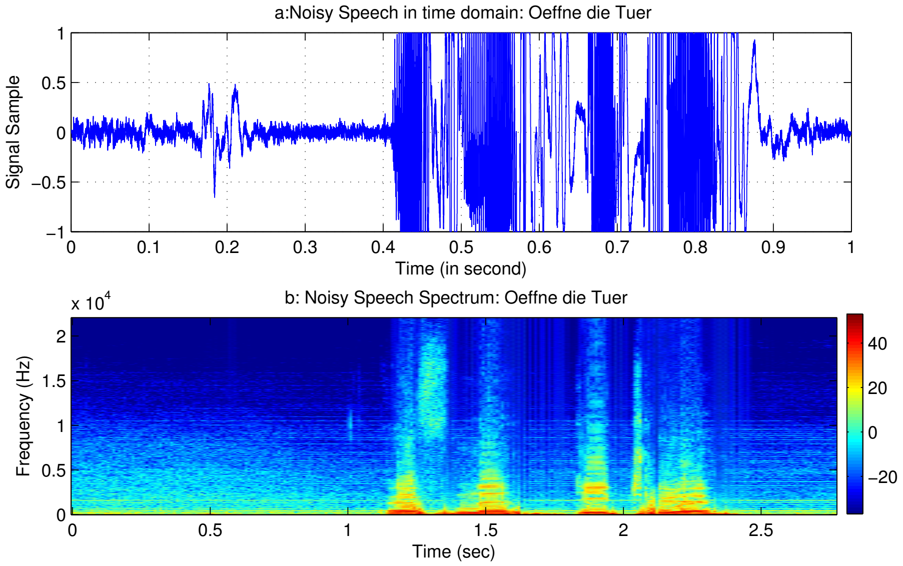


Figure 3.3: Hybrid noisy signal : Noisy speech signal in the time and in the frequency domain

Figure 3.4 shows sound levels of our measurements using A-weighting filter. These are collected at different times from the mixed noisy environment. This figure implies the varying sound levels of the signals.

$$\alpha_A(f) = \frac{(12200^2)f^4}{(f^2 + 20.598997^2)^2(f^2 + 12200^2)((f^2 + 107.7^2)^{0.5})((f^2 + 737.9^2)^{0.5})} \tag{3.2}$$

If the frequency of signal spectrum $S(k)$ measured at $f_k$ where $k$ is the index of the frequency component then the A-weighting filter measurement for this

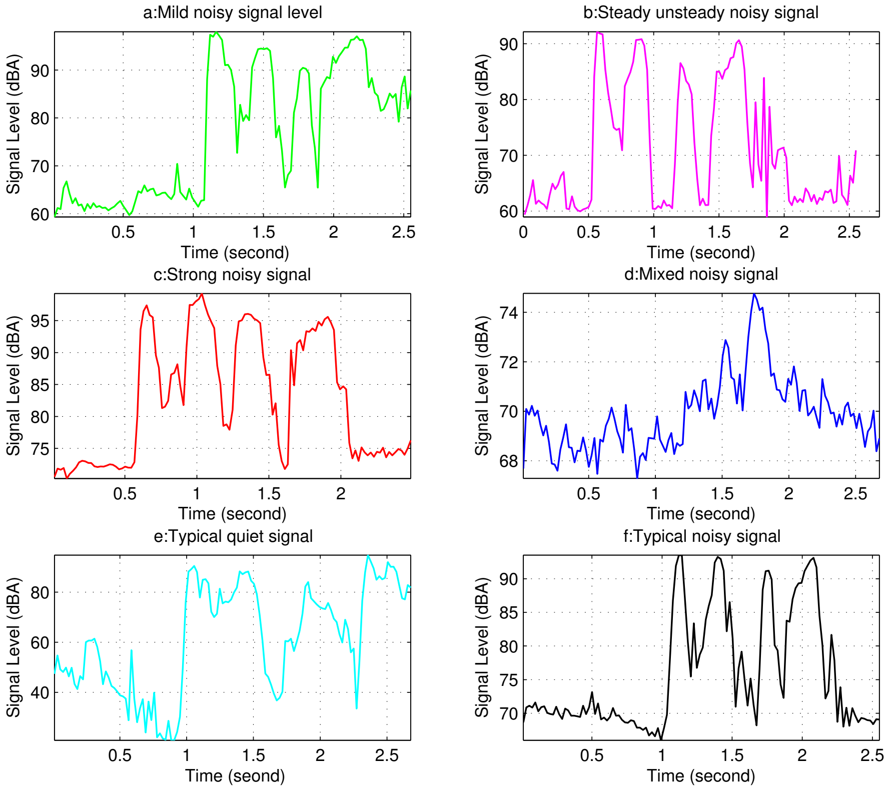


Figure 3.4: Hybrid noisy sound level measurements by an A-weighting filter

frequency at specific index $k$ is obtained by equation (3.3) for $f_k = k\Delta f = \frac{f_s}{N} = \frac{1}{NT}$. $f_s$ is the sampling frequency, $N$ is the length of the interval. Here we have chosen $N = 256$. The DFT is computed at each $N$ interval. $T$ is the sampling period $T = \frac{1}{f_s}$. The frequency resolution denoted by $\Delta f$ is $\frac{1}{NT}$. The A-weighted signal level measurement denoted by $S_A(k)$ is computed by the multiplication of the A-weighting filter frequency response and the noisy signal spectrum $S(k)$ in equation (3.3).

$$S_A(k) = \alpha_A(f_k)S(k) \tag{3.3}$$

Then the signal energy $\zeta$ shown in equation (3.4) is computed by squaring the spectrum of $S_A(k)$ for 0 to $\frac{N}{2}-1$ because of the symmetry property of the Fourier transform.

$$\zeta = |\sum_{k=0}^{\frac{N}{2}-1} S_A[k]|^2 \tag{3.4}$$

Next the signal level in dBA is computed by equation (3.5) where $\zeta_{ref}$ is reference pressure and its value is 0.000204 dynes/cm$^2$.

$$\text{Signal level in dBA} = 10\log_{10}\frac{\zeta}{\zeta_{ref}} = 10\log_{10}\zeta - 10\log_{10}\zeta_{ref} \tag{3.5}$$

In equation (3.5), $\zeta_{ref}$ is a constant which is replaced by a calibration constant in its real application.

$$\text{Signal level in dBA} = 10\log_{10}\zeta + C \tag{3.6}$$

In our measurements, we use a calibration constant of 55 dBA because in a noisy environment a human being can perceive sounds in the range of 50 to 90 dBA [91].

In figure 3.5, the signal level is computed on the frame (defined in chapter 12) of the signal using an A-weighting filter. In this 3-D figure, we see the sound level, time and frequency information of the hybrid noisy signal.

### 3.4.2 Box Plot Evaluation

The box plot shows the distribution of data in general. This represents the data using their lowest value, highest value, median value and the size of the first and the third quartile. The median is counted by equation (3.7) and is denoted by $d(m)$ where $m$ denotes the median and $n$ is the number of data points in the

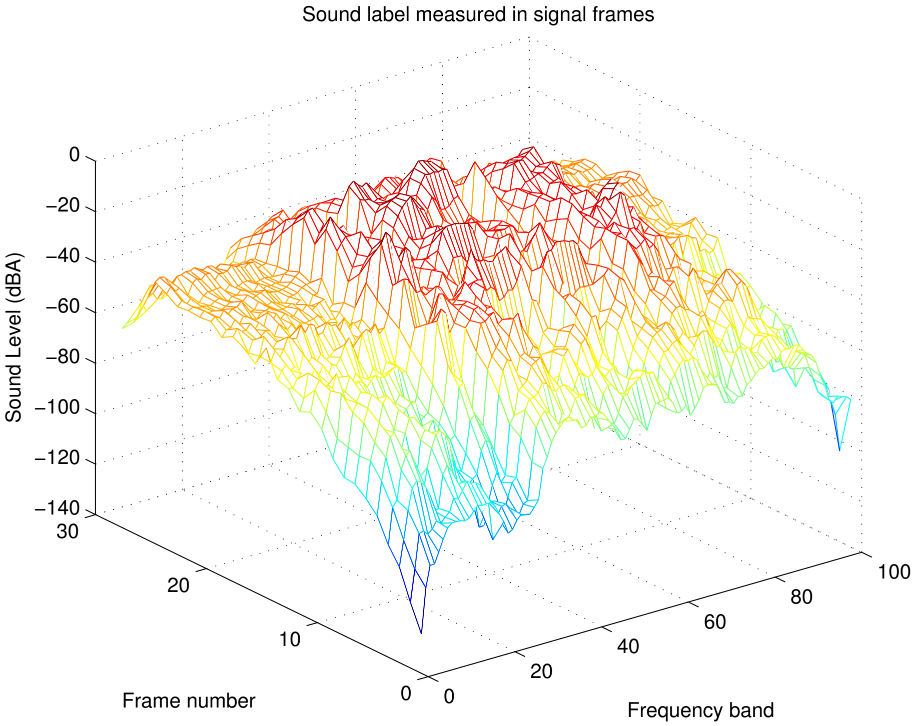


Figure 3.5: Noise level and noisy signal energy in signal frames

observation.

$$d(m) = \frac{n+1}{2} \tag{3.7}$$

The depth $dq_l$ of the quartile is shown in equation (3.8). There $n$ is the number of data points in the observation, $i$ takes the value based on the quartile depth i.e. $i$ takes the value of $l$. In equation (3.8), $dq_1$ denotes the first quartile. $i$ can be 2, 3, or 4 which indicates the corresponding quartile.

$$dq_l = \frac{in+2}{4} \tag{3.8}$$

In figure 3.6, we see a number of noisy signals denoted by dat1, dat2, dat3, $\cdots$ , dat10 on the boxplot. We see there the noisy signals and their samples values and signal level in a box plot.

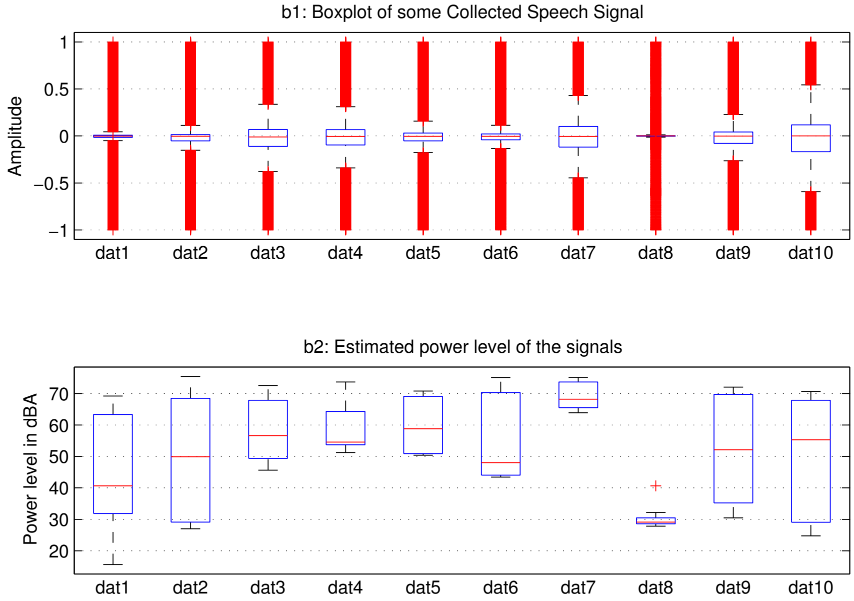


Figure 3.6: Hybrid Noisy Signals : Sound level in Box plot

The part between the lowest adjacent limit and the bottom of the box represents one-fourth of the data. One-fourth of the data falls between the bottom of the box and the median and another one-fourth between the median and the top of the box. The part between the top of the box and the upper adjacent limit represents the final one-fourth of the data observations. This opens up the pattern for the data and their variations [138].

### 3.4.3 Signal Energy and Kernel Density Estimation

In figure 3.7 we see the energy information of the noisy signals in the signal frame (defined in chapters 8, 12) and in b, we see the probability distribution of the samples values computed by probability density function (pdf) of the noisy signals. The energy of the signal is computed in figure 3.7 by $\sum_{n=0}^{N-1} s[n]^2$ where $n$ is an integer i.e. $\mathcal{Z}$ and $s_n$ is a signal of finite length $N$. The probability density function of the random variable $X$ is calculated using the kernel probability density estimation (k-pdf). We assume $n$ independent measurements such as $x_1, x - 2, \cdots, x_n$ from the random variable $X$ is considered. The kernel density function approximates the probability density function of $p(\centerdot)$ of the random variable $X$. The computation is available in matlab. For this we have used the matlab "kdf" function. The calculation is followed by the formulation given in equation (3.9). In this equation, $p_h(x)$ is the pdf as a function of $x$. $h$ is a normalization factor; $h > 0$ but less than 1 and dependent on the available information of $x$. Here $x$ is independent and identically distributed (iid). Here $K(\centerdot)$ denotes kernel function, $x$ is any value and $x_i$ is a sample of the $x$ [18], [137].

$$p_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - x_i}{h}) \tag{3.9}$$

### 3.4.4 Signal to Noise Ratio (SNR)

We judge the speech intelligibility by the signal to noise ratio (SNR). The speech intelligibility says if the speech is audible or not. The signal strength and noise strength are primarily measured by the signal to noise ratio (SNR) in dB. This gives a relative performance of the signal with respect to noise. If the signal strength is higher than the noise strength, the ratio is positive. If the noise strength is higher than the signal strength, then the SNR is negative.

The SNR is computed by equation (3.10). There the SNR is denoted by $\varkappa_m$, $P_s$ is the power of the signal and $P_n$ is the power of the noise. The whole signal is divided into M segments such that $m = 1, 2, \cdots, M$. There $N$ is the number of samples in each $m$. $y_m[n]$ is noisy speech, $b_m[n]$ is the noise collected from the

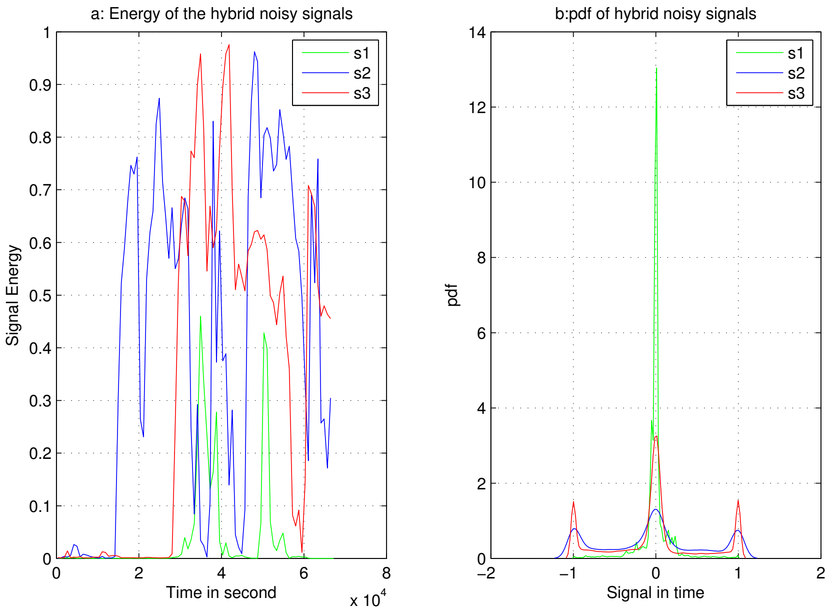


Figure 3.7: Hybrid noisy signal : Signal energy in signal frames and pdf of noisy signal

environment without speaking commands to the microphone.

$$\varkappa_m = 10 \log_{10} \frac{P_s}{P_n} = 10 \log_{10} \frac{\frac{1}{N} \sum_{n=0}^{N-1} y_m^2[n]}{\frac{1}{N} \sum_{n=0}^{N-1} (b_m[n])^2} \tag{3.10}$$

## 3.5 Overview of DANSR

The different kinds of noises characterize the situation as hybrid. As said, we need a hybrid solution consisting of different elements. A major problem is the integration because these elements influence each other and cannot be combined in an arbitrary order.

Figure 3.8 shows the DANSR system. This gives a rough overview. Details of the approaches and their motivations are discussed in the subsequent chapters in the thesis. In the figure we see, first we focus on reducing the noise, then we apply the perceptual feature extraction consisting of spectral shaping, spectral analysis and perceptional feature transformation, and finally the GMM model, evaluation, searching and learning in the classification and recognition stage to obtain the most likely result. Details of the approaches represented in figure 3.8

are in the subsequent chapters.

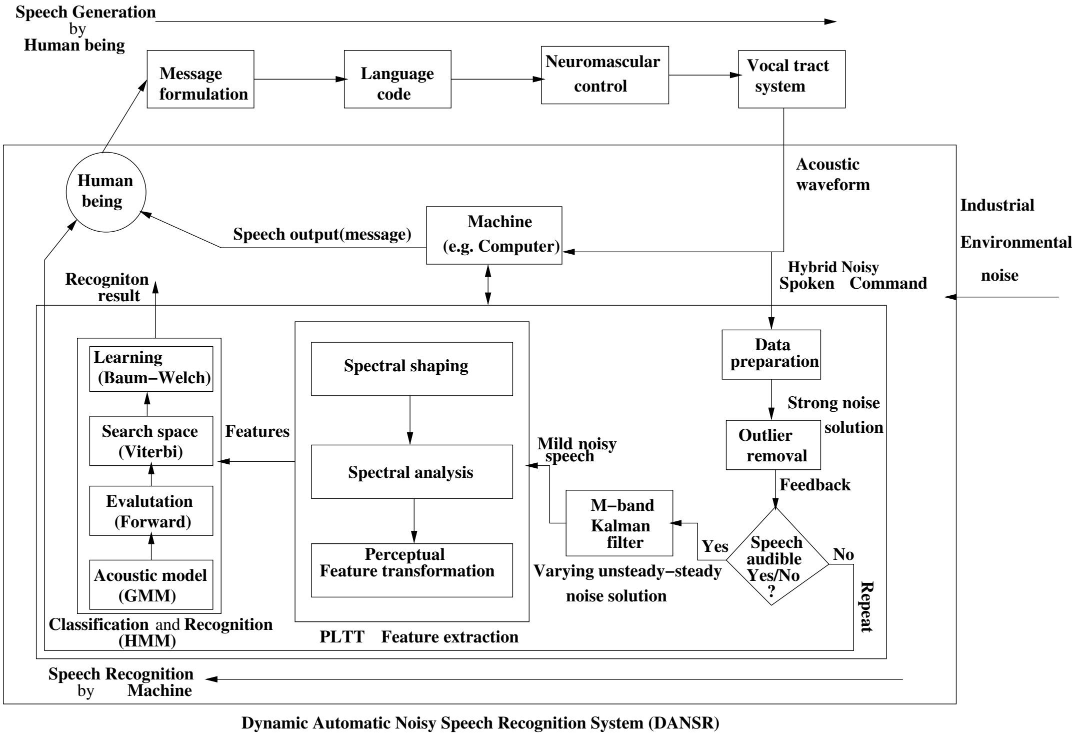


Figure 3.8: Hybrid Noisy Speech Recognition: Framework of DANSR

Above we have introduced active and passive processes. Now we extend these notions to tasks and they will be discussed now.

**Passive Tasks** Data collection and their proper management, noisy signal evaluation.

**Active Tasks** Pre-emphasizing the collected data, active noise reduction approaches, perceptual feature extraction and classification and recognition of the features.

### 3.5.1 DANSR's Hybrid Noise Treatments

The noise treatments of the DANSR are introduced next.

#### 3.5.1.1 Noisy Speech Pre-emphasizing

According to our plan to maintain a precise 3 second speaking time for the data collection was sometimes difficult. Despite the noise, the speaker's speaking style, accent and non-speech or silence in between added more redundancy in the data. Therefore we smoothened the data prior to its processing. We termed this pre-emphasizing. This is discussed in chapter 4. We collected spoken commands. These collected spoken commands are our data. We present a short overview over the considered noise types that will be detailed in the following chapters.

#### 3.5.1.2 Strong Noise

As mentioned, this noise occurs rarely, randomly for a short time period. We call a very short pulse with very high amplitude strong noise. We are not accurate about its occurrence but it happen stochastically and we are not certain about its effect on the spoken commands. For the strong noise there are no absolute numbers given. Our sizes have a qualitative character. They are defined in relation to the average noise. For this purpose we choose a threshold. A noise signal is considered as strong if its amplitude exceeds the threshold. The strong noise is handled first by detecting it as an outlier. This is an abnormal quantity in an observation. The identification of outliers is a standard difficult problem in data analysis. In chapter 5 we will discuss strong noise in detail. Here we just mention that it is modeled by a Poisson statistics. We see in figure 3.9 a signal that is modeled by Poisson process. This says the Poisson process $x_t$ generates the strong noise 5 times in a time period of duration $t = 2$. $\lambda$ is the occurrence of the event which is the strong noise and $t$ is the time interval.

#### 3.5.1.3 Mild Noise

The mild noise is modeled as white noise. This noise is characterized by a Gaussian process. This is commonly known as white Gaussian noise (WGN). There the mean denoted by $\mu$ is zero and variance denoted by $\sigma$ is 1. For instance, take a Gaussian process $x$ for time instants $n$ and $n = 0, 1, 2, \cdots, N-1$. Then $x[n]$ and $x[n+1]$ are independent and uncorrelated. This indicates $E\{x[n+l]x[n+m]\}$ is zero for $l \neq m$ and $l, m \in \mathcal{Z}$. The mild noise is shown in figure 3.10 where we see the time domain plot of WGN, autocorrelation of two WGN sequence is 1. The variance of white noise is 1 and mean 0. The frequency information of WGN is also shown. These plots are generated using matlab functions rand, xcorr and

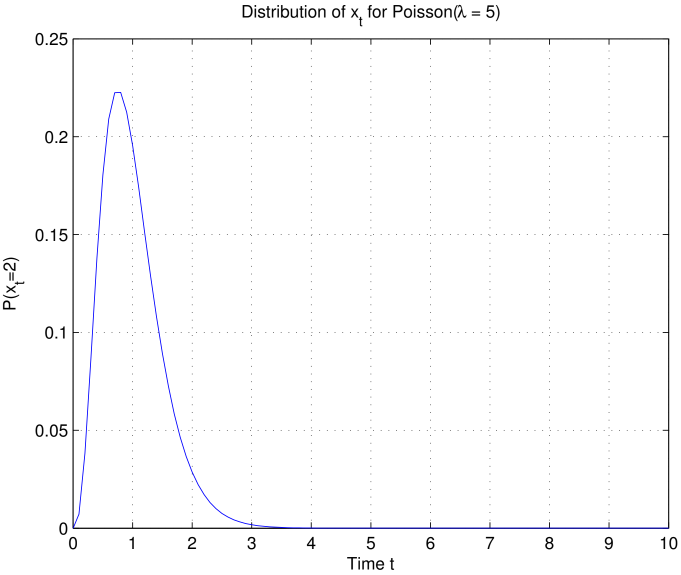


Figure 3.9: Strong noise modeled by Poisson process

fft. The rand generates uniformly distributed random sequences, fft gives the frequency information of of the WGN and xcorr computes the auto-correlation between the two random squence at time $n + l$ and $n + m$.

This Gaussian process is also known as normal standard distribution process. The probability density function (pdf) of this noise is shown in equation (3.11).

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x-\mu)^2}{2\sigma^2}) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2}) \tag{3.11}$$

The model of mild noise is characterized by white Gaussian process is used in eq (3.12).

$$\sigma^2 = E\{[x^2[n]]\} \tag{3.12}$$

#### 3.5.1.4 Steady-unsteady Time Varying Noise

This noise comes from a running machine. This is characterized as a Gaussian process but its mean $\mu$ is not zero and the variance $\sigma$ may not be always 1 as it is the case for the white Gaussian noise. If $x$ is a Gaussian process for time instants $n$ and $n = 0, 1, 2, \cdots, N-1$, and its mean is $\mu$ and variance $\sigma$, then the pdf of the noise is characterized by equation (3.13) where $p(x)$ is the pdf of $x$. The colored noise modeled by the Gaussian process is pictured here in figure 3.11

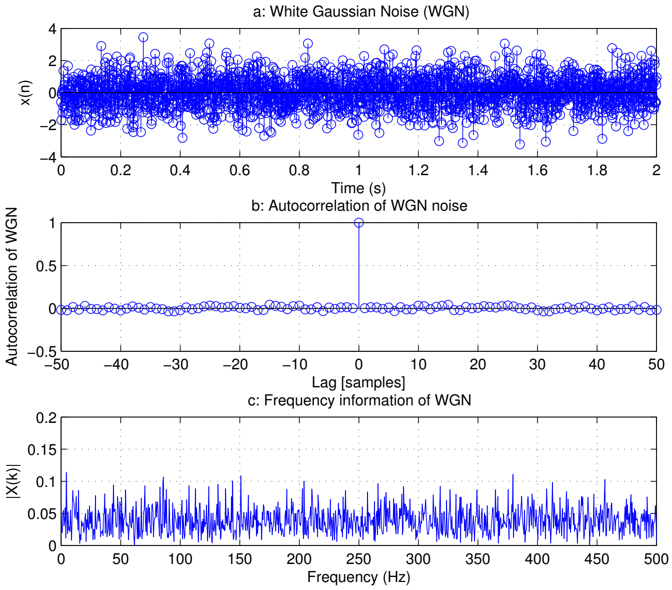


Figure 3.10: Mild noise modeled by white Gaussian noise (WGN)

where we see the random Gaussian noise, its auto-correlation and the frequency information computed by matlab fft and corr functions.

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x-\mu)^2}{2\sigma^2}) \tag{3.13}$$

The noise model shown in equation (3.14) is an AR model but its parameters are obtained by the linear prediction namely the Yule-Walker approach. In equation (3.14), the noise $d[n]$ is a linear combination of past $i$ many $\beta$ coefficients and a disturbance $w[n]$. This is assumed to be a white noise and it is weighted by $g_b$.

$$d[n] = \sum_{i=1}^{q} \beta_i d[n-i] + g_b w[n] \tag{3.14}$$

For treating this noise, the signal is first divided into sub-bands using a cosine modulated quadrature mirror filter bank (QMF) and then the noise is minimized from each sub-band by a spectral minimization technique. Afterwards the signal is enhanced in each band by Kalman filter. In this noise reduction, noise is varying in each sub-band. The solution of this is discussed in details in chapter

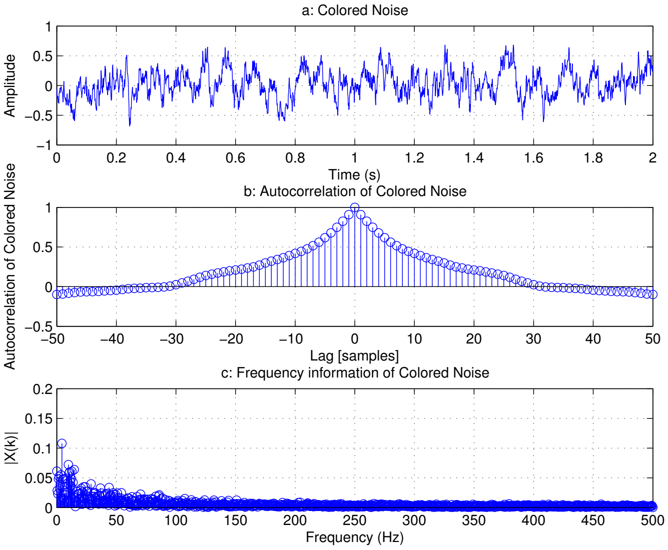


Figure 3.11: Time varying steady-unsteady noise modeled by Gaussian process

10.

Next we present the framework of our DANSR approach.

### 3.5.2 Framework of DANSR

The architecture of a typical ASR was introduced first in chapter 2 in section 2.3.1. The DANSR is a small vocabulary speech recognition system. The DANSR uses basically the HMM. The frame work has the equation written below as a central element. This is repeated here but it is discussed in detail in chapter 14. In the equation $p(\mathbf{o}|\mathbf{q}, \lambda)$ is called an acoustic model where the likelihood of the features given the model $\lambda$ has to be obtained. $\mathbf{o}$ is the feature vectors, $\mathbf{q}$ is the sequence of states. The equation is discussed in chapter 14 in section 14.1.

$$p(\mathbf{q}|\mathbf{o}, \lambda) = \arg\max_{\mathbf{q}} \frac{p(\mathbf{o}|\mathbf{q}, \lambda)p(\mathbf{q}, \lambda)}{p(\mathbf{o})} = \arg\max_{\mathbf{q}} p(\mathbf{o}|\mathbf{q}, \lambda)p(\mathbf{q}, \lambda)$$

If we look at figure 3.8, then we see that the main key tasks of the DANSR are:

- Data collection
- Noise reduction or removal

- Feature extraction
- Classification and recognition

These tasks give a guide for the dissertation.

# Chapter 4

# Pre-emphasizing of DANSR

**Outline of the Chapter** In this chapter we describe our data collection, their preparation and their pre-emphasizing. We realize our data preparation and data pre-emphasizing by pre-filtering. This reduces the redundancy and smoothens the data in two steps: i) Redundancy removal, and ii) Pre-emphasizing by pre-emphasis filter. In the reduction step, it removes silence and then it uses the pre-filtering to smoothen the data. The reduction step makes the data size smaller. How this is performed is explained in this chapter.

## 4.1 Data Collection

The speech is a random process. The statistical properties such as mean, variance, correlation, probability density function (pdf), power spectral density are used primarily to describe the signals. For the processing and recognition we need a huge number of data in order to capture certain patterns and the statistics of the data.

We have a predefined list with small spoken commands. The list has 20 small German spoken commands. We collected data according to our predefined list (see Appendix).

We have used the digital recorder Zoom 4 Samsung Handy Recorder to collect our data and noise taking place in the industrial environment. Our selected data type is in 'Wave' format and we have used single channel only. We collected data using 48000 Hz sampling rate at 16 bits per sample. We positioned the recorder about 3 to 5 meters from the speaker for our data collection. We did this to avoid flappy sound or any clicking sounds generated in the mouth. The length of the data is 3 sec for each command. Therefore sometimes the speaking time limit was exceeded. Furthermore in the noisy stage, the speaker's speaking style, accent and non-speech or silence in between added additional redundancy in the data.

### 4.1.1 Location and Data Collection

The data are collected from the mechanical assembling and manufacturing laboratory of the university of Kaiserslautern, the environmental maintenance company Zoeller-Kipper Gmbh located in Mainz and the assembling and manufacturing company MM Packaging located in Kaiserslautern. In figure 4.1, we see the data we collected from the hybrid noisy environment. We collected the data from German speaking people. Each command is collected 100 times for training and 25 times for testing. The speaker is selected randomly from the environment. In

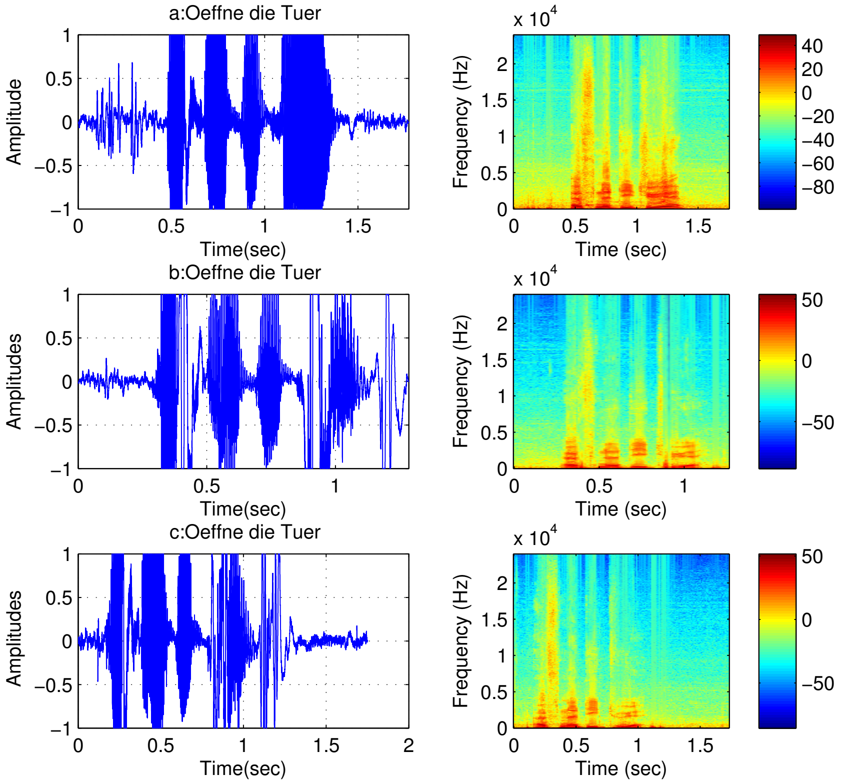


Figure 4.1: Data at first look at 48 kHz sampling rate

some cases the same speaker repeated the commands several times for the training. On the other hand, for the testing the speaker is not necessarily always the same. For our data processing, we are mainly concerned about the noise in the range of 200 to 5000 Hz for certainty at a speech bandwidth 200 Hz to 3500 Hz. In an industrial environment, a communication can take place if the background sound level is 70 dBA [23]. The noise we encounter has different extremity. This ranges from 50 dB to 110 dB.

Reducing the redundancy and then using a pre-emphasis filter to smooth and emphasize the signal is so far new in this area. We applied this at the very first step to prepare our data for their further processing. Our focus for the speech enhancement is 70 dBA sound level. We then use this speech for the speech pattern recognition processing.

We are concerned about building an acoustic model. To build such a model related to a speaker dependent system, a recognizer needs several hundreds of data. On the contrary, for a speaker independent system, a recognizer needs several thousands of data [120]. A collection of several thousands of data is not possible at the current scope of our research studies, therefore we consider a system that will be speaker dependent.

In figure 4.2, one sees how the same speech is different for the same speaker. This variability is one of the main reasons among others that makes this research very challenging.

We reduced some redundancies by applying an application dependent threshold. In order to prepare the data for the processing, we first decimated the data to a 16 kHz sampling rate, then we reduced non-speech pause, silence in the redundancy removal stage. How these are done is discussed below.

## 4.2 Data Preparation

The redundancy of the collected data is handled in removing and pre-emphasizing by the following steps:

- Decimation: This is a process that reduces the sampling rate by a factor. This process has two steps: i) Antialiasing filtering and ii) Down-sampling. The anti-aliasing filtering is used to avoid aliasing. For the anti-alising filter, we have used a low pass filter. This is designed using the windowing method. For the windowing, we used the Kaiser window function because of its controlling parameters. The Kaiser window has a shape controlling

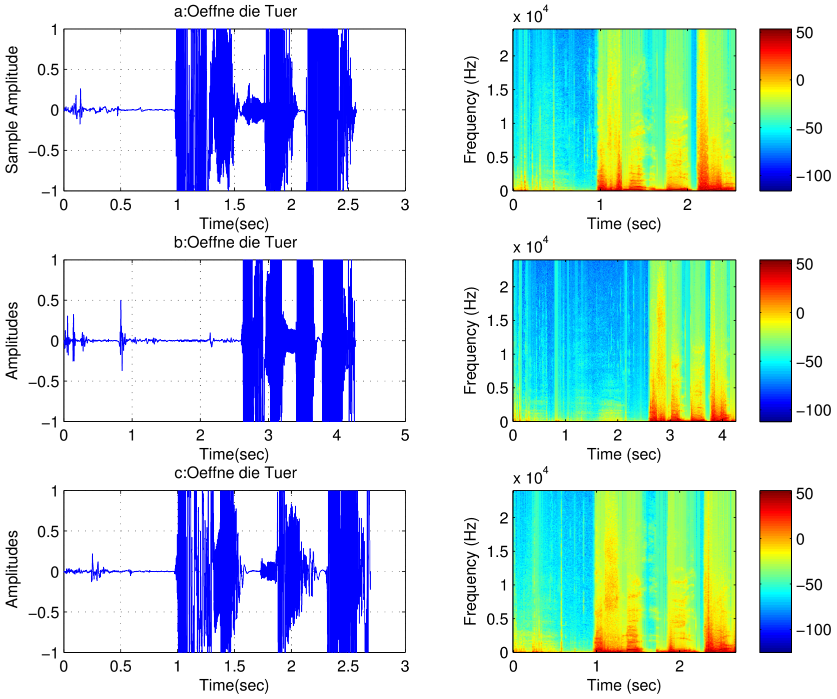


Figure 4.2: Variability of the same word spoken by the same speaker in the time and frequency domain at 48 kHz sampling rate in a relatively quiet residential environment

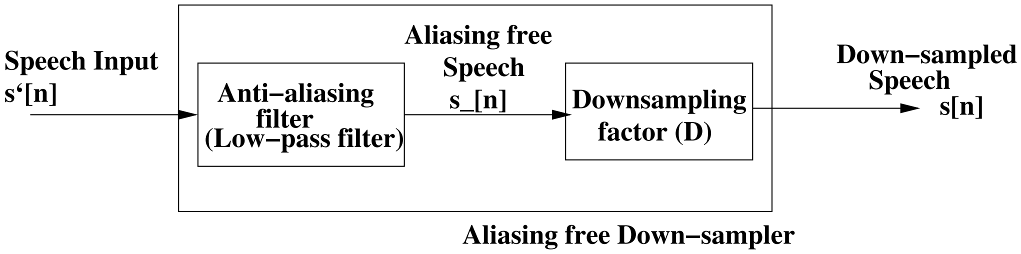

Speech Input
s'[n]
Anti–aliasing filter (Low–pass filter)
Aliasing free Speech s_[n]
Downsampling factor (D)
Down–sampled Speech s[n]
Aliasing free Down–sampler

Figure 4.3: Decimator: Data is downsampled from 48 kHz sampling rate to 16kHz sampling rate

parameter such as width of the main lobe and side lobe. Then we have used down-sampling by the factor 3. Here the sampling rate is reduced from 48 kHz to 16 kHz.

- Redundancy removal: This has several sub steps in order to remove the pauses in the speech or silence or non-speech sound of the speech signal. The sub-steps are: i) Compute the signal envelop by using the Hilbert transform and ii) Select the threshold. This reduces the redundancy and shortens the signal. It has a computational benefit due to less samples in the signal processing.

Next the decimation process is described.

### 4.2.1 Decimation

For the decimation, we first down sample from 48,000 to 16,000 Hz. This has a computational benefit and is explained below. For this we do not loose information because the speech signal is band-limited 200 to 3500 Hz. The human audible signal frequency lies between 20 Hz to 20 kHz. Yet most of the speech energy lies under 7 kHz [19]. Figure 4.3 shows the decimation process which has an antialiasing filter before down-sampling the signal. The antialiasing filter is a 64 length finite impulse response (FIR) low pass filter based on the Kaiser window, the cut-off frequency is 5 kHz,the Nyquist sampling rate is 8 kHz. The down-sampling factor is 3. In matlab, the resample function also does this down-sampling but we used the Kaiser window based low pass FIR filter for the decimation.

**Computational Benefit** Here we explain how the decimation is computationally beneficiary. If the speech recording time has a maximum of 3 sec, and if the speech signal is sampled at 48 kHz at 16 bits per sample (bps) i.e. 16 bps, then the storage space for the sampled speech is $3 \,\mathbf{.}\, 16 \,\mathbf{.}\, 48 = 2304$ kilo bits per sample (kbps) or $\frac{3.16.48}{8} = 288.5$ kilo byte per sample or $\frac{3.16.48}{8.1024} = 0.282$ mega byte per sample (mbps). **.** denotes the muliplication. This is doubled for the stereo typed two channels recorded samples. Therefore we selected the single channel stereo typed data where both channels record the similar information. If the sampling rate is 16 kHz at 16 bits per sample, then the required space for the sample is $3 \,\mathbf{.}\, 16 \,\mathbf{.}\, 16 = 768$ kbps. So the down sampling in our scenario will save a huge computational cost and a lot of processing time.

### 4.2.2 Envelope Detection

The envelop of a signal is a boundary within which the signal is contained. The envelop of a signal is also an estimate of the signal level. The pause or any clicking distortion or the silence in the speech signal is detected by the envelop of the speech segment [108]. One way of doing this is computing the envelop of the signal by the Hilbert transform of the signal. We have not investigated the Hilbert transform in details. We reviewed this only in order to compute the signal envelop.

The basic goal of the Hilbert transform in the time domain signal is to get another time domain signal. The Hilbert transform shifts the frequency components of the signal by $-90$ degree but it does not change the amplitude. The Hilbert transform acts as a differentiator to a constant signal. This means if the signal has any constant component, the Hilbert transformation of the signal cancels this. This is equivalent to getting the zero mean of the signal. The speech signal is processed under the assumption that it is an ergodic process. In this process, the time average of the signal is equivalent to the ensemble average of the signal. The importance of this is that the time average of the signal can be computed easily but the ensemble average can not. This time average processing and more about the speech ergodic process we will see in chapter 7 and 8.

In order to compute the envelope of the signal, we first take the Hilbert transform of the signal. The envelope signal has a frequency that is much lower than the measured signal. The problem is that the envelope makes the signal rough [106]. On the other hand the pre-emphasis filter increases the frequencies from low to high smoothly.

The computation of the envelop using the Hilbert transform also maintains a representation of the signal. The envelop of the signal using the Hilbert transform sets a qualitative boundary around the silence. We have used this computation to obtain the envelop of the signal. Then we have selected the threshold. We exclude all the data that fall below the threshold and remove pause, silence typed redundant data. The advantage of the smoothing is that we get less computational costs due to reduced amount of data because of the non-speech signal removal.

#### 4.2.2.1 Formulations

Below we mention the relevant formulations for these computations. For this we follow the formulations mentioned in [106].

The Hilbert transform of the signal $s[n]$ is denoted by $s_H[n]$. How we computed this is given in equation (4.1). This says the convolution of the signal $s[n]$ with $\frac{1}{\pi n}$ gives the Hilbert transform of $s[n]$ denoted by $s_H[n]$. $\otimes$ denotes the convolution operation.

$$s_H[n] = \frac{1}{\pi n} \otimes s[n] \tag{4.1}$$

In equation (4.2), we see how the envelop is computed using the real valued signal $s[n]$ and the Hilbert transformed signal $s_H[n]$.

$$\mid s_A[n] \mid = \sqrt{\{s[n]^2 + s_H[n]^2\}} \tag{4.2}$$

In figure 4.4, in a we see the spectrum of oeffne die Tuer computed by FFT where frequency along the x-axis and amplitude of the frequency information of the signal along the y-axis. In the same figure in b, we see the spectral envelop of the oeffne die Tür. The spectral envelop shows the signal amplitudes versus frequency in the plot. In b, we see the FFT spectrum of the envelop computed by Hilbert transform. For implementation, we used the existing matlab function hilbert to compute the Hilbert transformation.

### 4.2.3 Adaptive Threshold Selection

The silence intervals from the speech are removed using a threshold. One commonly takes for the threshold one fourth of a median of the envelop for removing speech silence or pause or clicking sounds from spoken speech. There is again no precise reason for doing so. Then the speech samples with amplitudes below the threshold are detected similar to literature [4]. This literature detected these samples and deleted the non-speech sound from the speech. Thus the removal of the pauses shortened the time and thus the length of the signal.

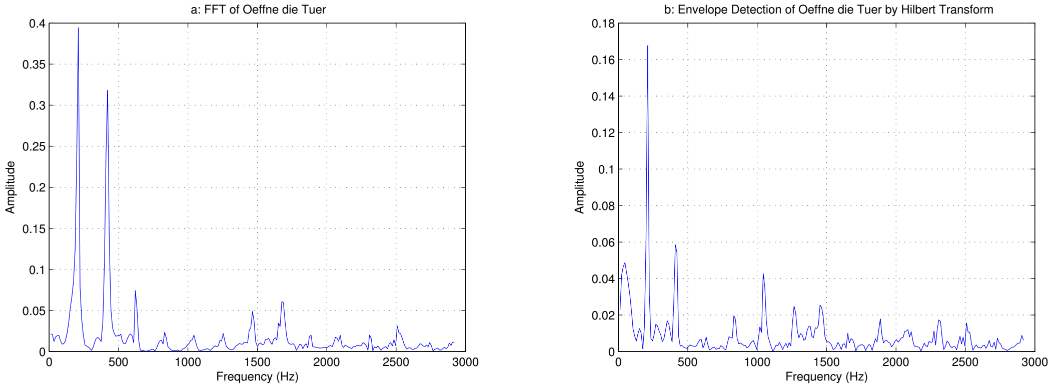


Figure 4.4: Spectrum of hybrid noisy speech and spectrum of envelop computed by Hilbert transform

In figure 4.5 we can see how the redundancy of the data is removed by applying decimation, envelop detection and then the threshold selection. The recorded samples in a are reduced from 85056 times 2 to 26287 times 1, in b 61697 times 2 to 17908, in c 84096 times 2 to 25142 times 1. These data are easier to process than the originally recorded data. 2 means two channels. We have used only one channel but the information on the two channels are about the same. The two channels record data in two directions such as left and right. One channel can be selected only for the right or for the left direction.

**Alternative Recommendation** An alternative to this redundancy removal approach is to apply Savitzky Golay filters to the raw signal envelope before computing the phase and its derivative [108]. But we have not yet investigated this approach.

## 4.3 Pre-emphasizing and Pre-emphasis Filter

The purpose of the pre-emphasis filtering and its effect is discussed here. In the speech processing literature equation (4.3) is seen as pre-emphasizing the signal and the filter in equation (4.3) is known as pre-emphasis filter [80]. A common way of seeing the application of a pre-emphasis filter is to emphasize the frequency component by considering both the low and high frequency components of the signal. The formants lie in the frequency range from 200 to 3500 Hz. The speech signal is a relatively low frequency signal.

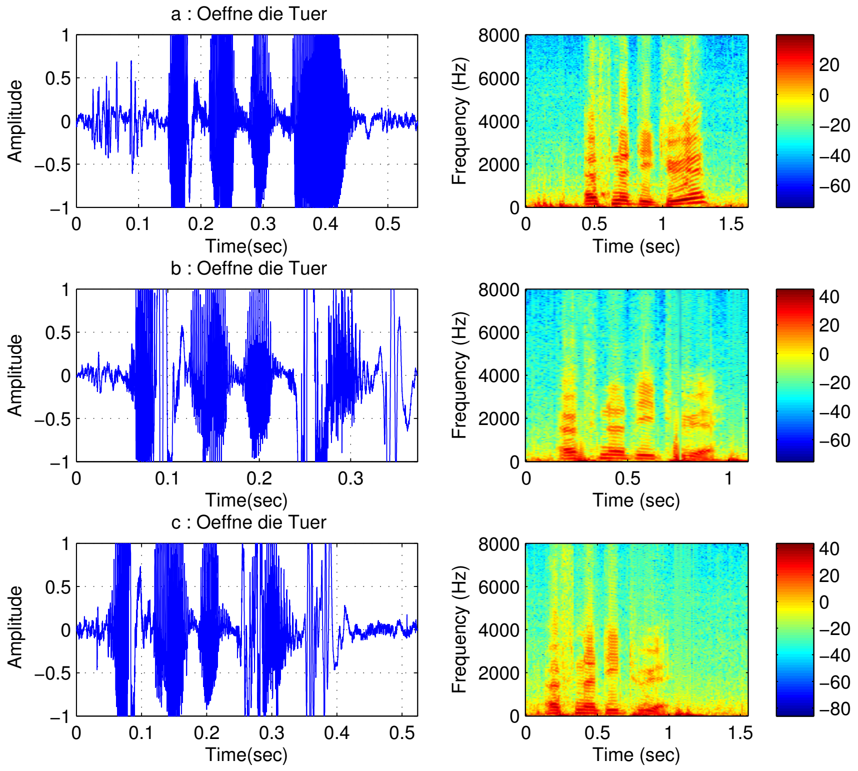


Figure 4.5: Redundancy removed signal and sampling rate is 16 kHz: Time domain plot and spectrogram

The redundancy removed signal denoted by $s^{'}[n]$ is then used to pre-emphasize the high frequency of the speech signal prior to its analysis. The pre-emphasis follows 6 dB per octave rate. This means if the frequency is doubled, then the amplitude increases by 6 dB. The speech sound has normally higher amplitudes in the low frequencies than in the high frequencies.

The pre-emphasize filter is identical to the filter that is used to model the lip radiation filter discussed in chapter 6. The pre-emphasis filter cancels the effect of the glottis. The system difference equation is presented in equation (4.3). The system is shown in figure 4.6. We see the result of the pre-emphasis filter is the emphasized signal $s[n]$ if the input signal is $s'[n]$. This pre-emphasis filter reduces the effect of 6dB/octave loss occurring by the glottal source and lip-radiation.

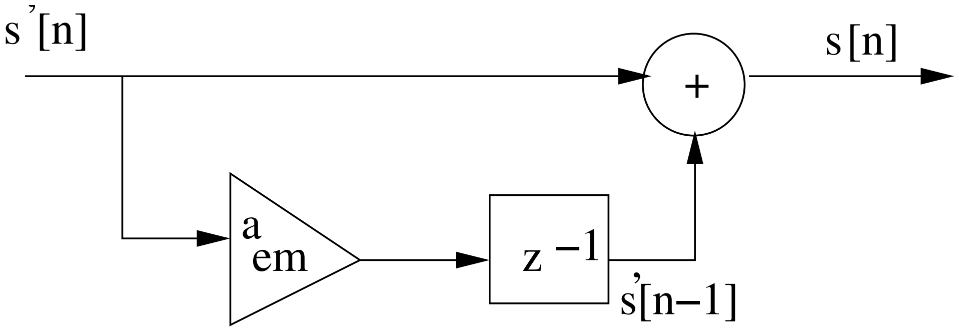


Figure 4.6: Pre-emphasis filter

$$s[n] = s'[n] - a_{em}s'[n-1] \tag{4.3}$$

Equation (4.3) can be rewritten in equation (4.4) in the z-domain by replacing $s'[n-1] = S'(z)z^{-1}$.

$$S(z) = S'(z)(1 - az^{-1}) = S'(z)H_{em}(z) \tag{4.4}$$

The transfer function of the pre-emphasize filter shown in equation (4.5) is just a high pass filter where we have approximately $a_{em}$ close to 1, for example 0.97. The determination of the coefficient $a_{em}$ is an empirical adjustment. This is again not precisely defined.

The z-transform of equation (4.3) results in equation (4.5) .

$$H_{em}(z) = 1 - a_{em}z^{-1} \tag{4.5}$$

In this equation, if $a_{em} > 0$, the pre-filter acts as a low-pass filter and if $a_{em} > 0$, the filter is a high-pass filter. The frequency response of this filter increases slowly from low to high, therefore it sets up a balance between the high and low pass frequencies [114]. The parameter $a_{em}$ controls the slope of the curve [147]. Therefore, this pre-filter may be called a pre-emphasis filter. In figure 4.7, we see the amplitude and phase response of the pre-filter for $a_{em} = 0.97$. This response also shows that it is a high pass filter.

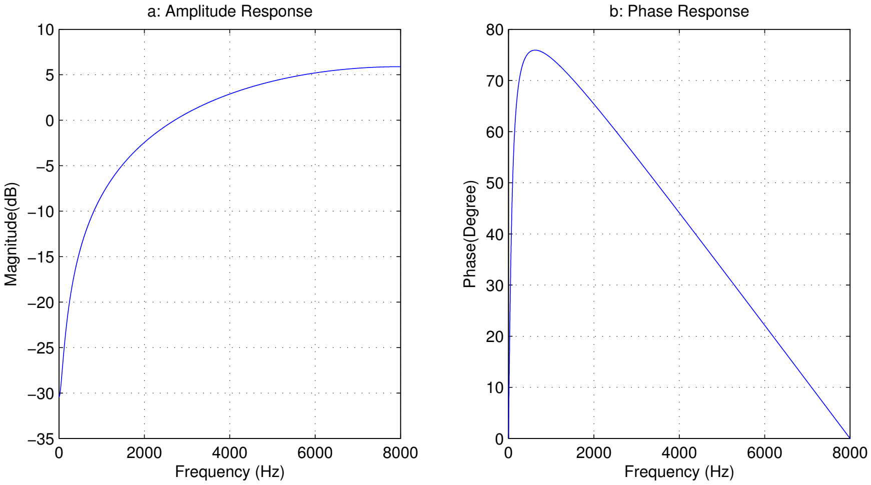


Figure 4.7: Amplitude and phase response of the pre-emphasis filter

In figure 4.8, $a$ shows the noisy signal which is pre-filtered as it is shown in $b$. In the same figure $c$ shows the spectrum of the noisy speech and $d$ is the spectrum of the pre-filtered speech. In figure 4.9, $a_{em}$ is the redundancy removed speech signal which is pre-filtered in $b$. In the same figure $c$ shows the spectrum of the redundancy removed signal and $d$ shows the spectrum of the redundancy removed pre-filtered speech. In both figures 4.8 and 4.9, we see the frequency is flattened by the employment of the pre-filter. This amplifies the high frequency components and attenuates the low frequency components.

In figure **??**, the industrial noisy speech and its pre-emphasized signal is shown. We can see in the figure a substantial amount of non-speech typed samples are removed. For this visualization , we have used Praat software [102].

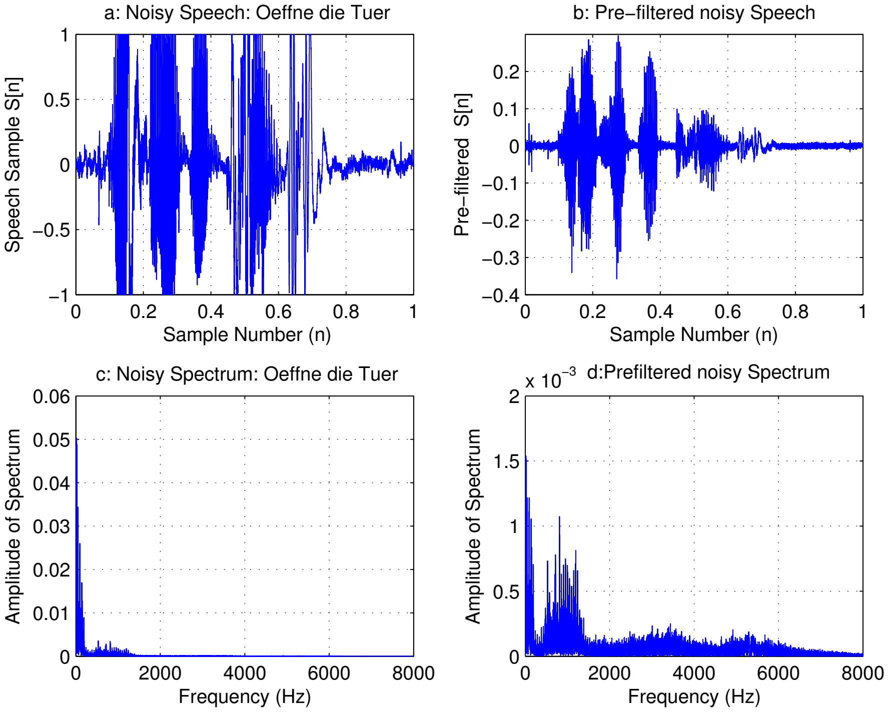


Figure 4.8: The effect of pre-emphasis filter on the speech signal: Noisy signal

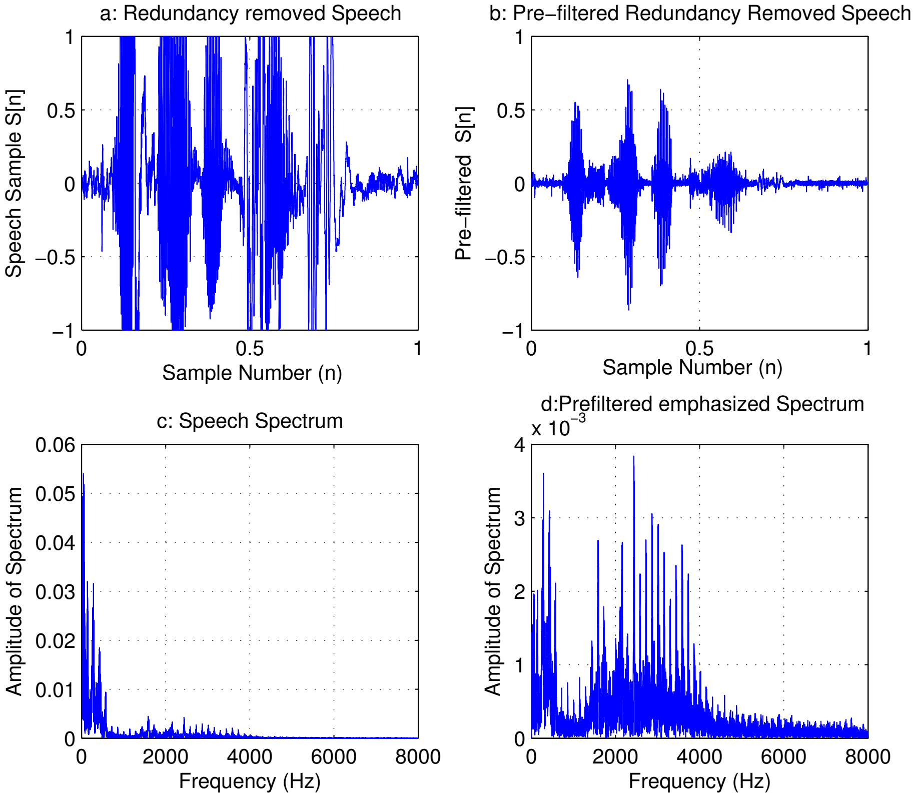
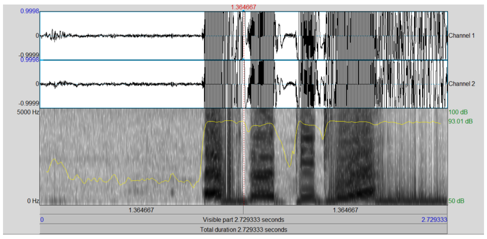
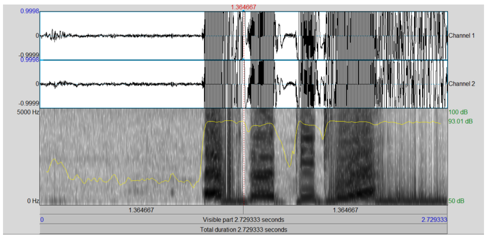


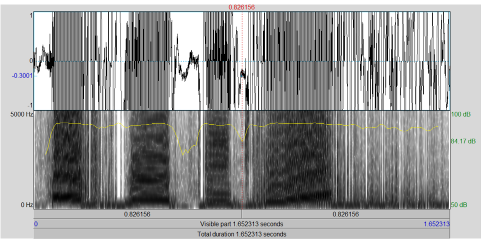
Figure 4.9: The effect of pre-emphasis filter on the speech signal: Redundancy removed signal



- Industrial hybrid noisy signal: Öffne die Tür.
- Sampling rate: 48 kHz.
- Time: 2.729 second and total samples: 131008 × 2.



- Industrial hybrid noisy signal: Öffne die Tür.
- Sampling rate: 48 kHz.
- Time: 2.729 second and total samples: 131008 × 2.



- Pre-emphasized signal: Öffne die Tür.
- Sampling rate: 16 kHz.
- Time: 1.652 second and total samples: 26437.

# Chapter 5

# Strong Noise Solution

**Outline of the Chapter** We have provided our solution to the strong noise problem in this chapter. For this first we describe the strong noise as an outlier. To detect the outliers we select an adaptive threshold. This will then be used to remove them. First we describe this approach in general. Because our overall approach is based on a probabilistic basis we need to introduce such a basis here too. For dealing with strong noise we selected the Poisson distribution for our purpose and we describe its basic properties first. Then we draw the connections to our problem and design a matched filter for the treatment.

## 5.1 Basic Steps

We have described strong noise as something where human beings cannot understand the speech due to its loudness. For us, this happens for a very short time only and occurs randomly distributed. Using our own intuition and our own real life experience, we see a strong noise interrupts understanding a speech clearly. Therefore our approach considers the strong noise as outliers occurring in a larger speech process. First we describe the outlier definition, its detection and then its removal.

An outlier is an abnormal quantity in an observation that is often marked as an perturbation in the observation. Detection and removal of this outliers need a careful analysis so that the remedy of the outliers does not affect the signal. The remedy most often begins with the statistical information of the data such as the probability distribution of the data, a histogram or a boxplot of the data.

Next we say what strong noise is for us. As said, intuitively, strong noise makes it even for humans almost impossible to undertand the speech.

First we introduce a basic assumption about strong noise. It says that strong noise occurs at small intervals only. However, these intervals are not regular but are randomly distributed. Below we will give a closer description saying what strong is. Basically, a noise signal is considered as strong if its amplitude exceeds some threshold and it lasts for a very short time.

The handling of outliers now proceeds in two steps:

- Identifying outliers.
- Removing outliers.

The identification of the outliers has an essential step to identify the intervals in which they occur. Removal means to remove these intervals from the speech signals. After the outlier removal there are two possibilities:

- The speech is not affected by the removal. In particular, the understanding is not disturbed because in the removed intervals no speech took place.
- Some part of the speech is removed too. As a consequence, the speech is incompletely delivered what affects the understanding.

In the first case one can proceed in the ordinary way. The second case is more difficult. For handling it the system needs an additional possibility for giving a feedback message from the receiver (machine) to the user (a human). This message is:

"Repeat the speech"

Such message is necessary because the given speech could anyway not be understood because of the strong noise. This also does not help if the statement is not understood properly because of the noise. The realization of the feedback provides no problems. It can be done in many ways that we will not discuss here.

## 5.2 Outlier Detection

The identification of outliers is a common difficult problem in data analysis. In addition, the definition is user and context dependent.

For the outlier identification we follow our previously given arguments. Therefore we need to determine :

- Underlying probability.

- The average loudness.
- The threshold above which noise is considered as strong. This is application dependent.

The threshold is adaptive as it depends on the expectation of the segment of the speech signal. This operation is carried out in the time domain. For our purpose we consider the following steps where we assume a noisy observation $z[n]$. We assume for the moment that a probabilistic environment has been defined.

- Determine a threshold $\theta$ where $0 < \theta < 1$.
- Compute the expectation $z_{ex} = E\{|z[n]|\}$.
- Identify set of the "outlier" samples $z[n]$ as $|z_{ex} - z[n]| \geq \theta$.
- Remove these identified "outliers" obtained in the previous stage from the noisy signal.

In the probabilistic description this will be made precise.

## 5.3 Stochastic Process

Before we make the steps precise we need to represent the noise as a stochastic process. The process representation will be somewhat different than before, in particular with respect to the underlying probability. For this we define a homogeneous Poisson model (defined in section 5.3.1) for the strong noise and its detection by matched filtering. Here we look at the notion of a shot. A strong noise event will be handled as a shot.

### 5.3.1 Poisson Distributions

First we consider Poisson distributions of events in general. A stochastic process of events with a random variable $x$ is a Poisson process with a parameter $\lambda$ , $\lambda > 0$ and with functions indexed by $k$ as the density functions:

$$f(k, \lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \quad \text{for } k = 0, 1, 2, \cdots \tag{5.1}$$

We assume for non-overlapping intervals $(t_1, t_2)$ and $(t_3, t_4)$, the random variables $n(t_1, t_2)$ and $n(t_3, t_4)$ counting the occurrences of events in the intervals are independent. The parameter $\lambda$ represents the average number of events in a unit length interval.

For any fixed $t \geq 0$, we consider a Poisson random variable $x(t)$ with the parameter $\lambda$. Here the expectation is $E[x(t)] = \lambda$ for $t \geq 0$. The autocorrelation function is $R(t_1, t_2) = E[x(t_1)x(t_2)]$ for $t_1 \geq 0$ and $t_2 \geq 0$.

If the expected number of occurrences of events in an interval is $k$, then the probability that there are exactly $k$ occurrences is equal to equation (5.1).

For a time interval of length $t$ we have the probability that $k$ events take place in this interval is given by equation (5.2).

$$P_\lambda(t,k) = \frac{(\lambda t)^k}{k!} e^{(-\lambda t)} \quad \text{for } k = 0, 1, 2, \cdots \tag{5.2}$$

#### 5.3.1.1 Homogeneous Poisson Model

Given an arbitrary impulse waveform $\delta(t)$ and a set of Poisson points $t_i$ the homogeneous Poisson model is shown in equation (5.3) [141]. In equation (5.3), $\delta$ appears at random times $t_i$ governed by a Poisson distribution and it has an amplitude $a_i$.

$$X(t) = \sum_{i=1} a_i \delta(t - t_i) \tag{5.3}$$

Each time an arriving of strong noise is detected, it causes a small impulse shaped noise as a shot in the signal. This means the arrival rate denoted by $\lambda$ of strong noise is described by equation (5.2).

Next we define the shots for the strong noise.

## 5.4 Shots

An event will now be a shot and the process model describes the shots. Shots are randomly, rarely, large valued events that occur at Poisson points. It is defined in equation (5.4) where $d[n]$ is the shot noise at the time instants $n$, $a_i$ is amplitude, $n_i$ is Poisson points. At a short time interval, the shots may occur or may not occur. The probability of the occurrence of the shots at short time interval is 0 or 1.

$$d[n] = \sum_i a_i \delta[n - n_i] \tag{5.4}$$

The response to $\delta[n]$ is called an impulse response and it is denoted by $h[n]$. If the arrival rate of the shorts is constant, then equation (5.4) can be rewritten by equation (5.5):

$$d[n] = \sum_i a_i h[n - n_i] \tag{5.5}$$

If the time interval denoted by $\Delta n$ and $\Delta n << 1$ and the shot may not occur in each interval, then its probable events can be confined by equation (5.6).

$$V_n = \begin{cases} 0, & \text{if no impulse occurs in the time interval } \ n\Delta n < n < (n+1)\Delta n \\ 1, & \text{if impulse occurs in the time interval} \, n\Delta n < n < (n+1)\Delta n \end{cases} \tag{5.6}$$

Similar to the Poisson distribution assumption, if we assume there is only one impulse occur in each time interval $\Delta n$ and the events are independent, then the probability of the event occurrence can be desribed by equation (5.7).

$$\begin{aligned} p(V_n = 0) &= \exp(-\lambda \Delta n) \approx 1 - \lambda \Delta n \\ p(V_n = 1) &= \lambda \Delta n \exp(-\lambda \Delta n) \approx \lambda \Delta n \end{aligned} \tag{5.7}$$

## 5.5 Matched Filter

If the input signal is in a finite time and mixed with noise, there is a filter which can be designed to maximize the signal to noise ratio (SNR). This type of filter is generally called the matched filter. The matched filter can be used to detect the shot noise. First we define the matched filter. The definition and derivation is mainly based on [5].

First we consider a more general setting. If a signal $x$ is mixed with noise $v$, then we can formulate the resulting signal as

$$x(t) = f(t) + v(t) \tag{5.8}$$

In equation (5.8), the signal $v(t)$ is a signal with known power spectrum $S(\omega)$. We assume that $f(t)$ is known and we wish to establish its present location. To do so, we apply the process $x(t)$ to a linear filter with a response $h(t)$ and Fourier transform $H(\omega)$. The resulting output $y(t)$ is

$$y(t) = x(t) \otimes h(t) \tag{5.9}$$

Now equation (5.9) can be rewritten by equation (5.10).

We see $y(t)$ which is expressed as $y_f(t)$ and $y_v(t)$.

$$y(t) = \int_{-\infty}^{\infty} x(t-\alpha)h(\alpha)d\alpha = y_f(t) + y_v(t) \tag{5.10}$$

By taking FFT, equation (5.10) can be rewritten by (5.11). Here $F(\omega), S(\omega), H(\omega)$ are the spectrum of the $f(t)$, $v(t)$ and the filter $h(t)$.

$$y(t) = \int_{-\infty}^{\infty} x(t-\alpha)h(\alpha)d\alpha = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)H(\omega)e^{j\omega t}d\omega \tag{5.11}$$

Now we describe the Fourier transformed of $y_v(t)$ by equation (5.12).

$$E\{y_v^2(t)\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} S(\omega)|H(\omega)|^2 d\omega \tag{5.12}$$

Since $y_v(t)$ is due to $v(t)$ and $E\{v(t)\} = 0$, then $E\{y_v(t)\} = 0$ and $E\{y_f(t)\} = y_f(t)$. The objective is to find $H(\omega)$ so as to maximize the signal to noise ratio $\varkappa$ such that at a specific time $t_0$ is written in equation (5.13).

$$\varkappa = \frac{|y_f(t_0)|}{\sqrt{E\{y_v^2(t_0)\}}} \tag{5.13}$$

Now if $S(\omega) = S_0$, by applying Schwarz's inequality we find equation (5.14). In the equation $E_f = (\frac{1}{2\pi})\int |F(\omega)|^2 d\omega$ is the energy of $f(t)$.

$$\varkappa^2 \leq \frac{\int |F(\omega)e^{j\omega t_0}|^2 d\omega \int |H(\omega)|^2 | d\omega}{2\pi S_0 \int |H(\omega)|^2 d\omega} = \frac{E_f}{S_0} \tag{5.14}$$

Equation (5.14) is an equality if equation (5.15) is taking place.

$$H(\omega) = kF^*(\omega)e^{-j\omega t_0} \tag{5.15}$$

Now the time domain of $H(\omega)$ in equation (5.15) is written in equation (5.16).

$$h(t) = kf(t_0 - t) \tag{5.16}$$

This determines the optimum $H(\omega)$ within a constant factor $k$. The whole system when these elements are combined is called the matched filter. The resulting signal to noise ratio is maximum and it equals $\sqrt{\frac{E_f}{S_0}}$.

## 5.6 Strong Noise and Matched Filter

Here we apply the results of the last section to our shot problem. Now in order to detect the presence of shots, we apply the detection formalisms that is discussed in [36]. One common approach is to insert a filter between the input and the matched filter so that the transfer function of the inserted filter is chosen such that the input is transformed into white noise and this is known as the whitening the process [141].

In order to apply the matched filter:

- We assume the signal is mixed with some noise which is shots.
- We use the linear prediction analysis for the signal model.
- We apply the whitening approach.

### 5.6.1 Analysis

Now the signal $s$ is mixed with a shot noise $d$ such that equation (5.17) holds. In this equation, we assume that $a_i$ can be only 0 or 1 for a signal interval and $n_i$ is unknown since shots are occurring randomly and rarely. The derivation of the analysis is discussed in details in [36].

$$o[n] = s[n] + d[n] \tag{5.17}$$

Here we assume that $s$ includes the true speech and all other noises in our modelling. We would like to detect the presence and location of shots $d$ in the signal $o$. Now the signal $o$ is a $p^{th}$ order AR process modeled as linear predictor shown in equation (5.18) using least squares approach. Here $\alpha_i$ is unknown and solved by using the Burg or Yule-Walker orthe unconstrained least squares (ULS) approach. The order of the model is $p = \frac{2lf_s}{c}$ where $l$ is length of the vocal tract, $c$ is the speed of the light and $f_s$ is the sampling rate. It is discussed in chapter 8. Since shots occur rarely, thus we can say $\hat{o}[n] \equiv \hat{s}[n]$. This is written in equation (5.18).

$$\hat{s}[n] = \alpha_1 s[n-1] + \alpha_2 s[n-2] + \cdots + \alpha_p s[n-p] = \sum_{i=1}^{p} \alpha_i s[n-i] \tag{5.18}$$

This says that the influence of the shots to the estimates is neglegible despite the fact that the size of the shots are individually quite large. As a consequence, the difference $o - \hat{s}[n]$ consists essentially of the shots only. That means we have to analyze this difference for identifying the shots. First we look at the estimation error in equation (5.19). $b[n]$ is a white noise and it has variance $\sigma^2$.

$$b[n] = s[n] - \hat{s}[n] = s[n] - \sum_{i=1}^{p} \alpha_i s[n-i] \tag{5.19}$$

Now we have the parameters $\alpha_i$ and we can design a FIR filter $F(z)$ such that equation (5.20) holds.

$$F(z) = 1 + a_1 z^{-1} + \cdots + a_p z^{-p} \tag{5.20}$$

- Now $b$ and $d$ are the input to the filter $F(z)$ and we are trying to detect $d[n]$.

The next idea is to design a filter that suppresses the signals with size less that the threshold. For this we define a filter that gives a maximum ratio of $d$ and $b$.

This will be a linear filter $g[n]$ that generates $z[n]$. Now suppose,

$$y[n] = b[n] + d[n] = b[n] + \sum_{i} a_i \delta[n - n_i] \tag{5.21}$$

Now following the hypothesis of the statistical detection of the matched filter, we have only two cases:

- $H_0$: $b[n]$.
- $H_1$: $b[n] + d[n]$

In equation (5.21), if $a_i$ is 0, then input to filter $F$ is $b$ only which supports the hypothesis i.e. $H_0$ noise is present only. If $a_i \neq 0$, then the shots denoted as $\delta$ is located in $n_i$. This supports hypothesis $H_1$. If we are under $H_0$, then the output of filter $F$ is $(b \otimes h)[n]$ where $h$ is the impulse response. Thus, for the white noise with zero-mean i.e. $\mu = 0$ and standard deviation $\sigma$, the threshold $\theta$ can be written by equation (5.22).

$$\theta = \sigma \sqrt{(1 + \alpha_1^2 + \alpha_2^2 + \cdots + \alpha_p^2)} \tag{5.22}$$

Since the filter is linear, the output of the filter $F(z)$ can be written as equation (5.23). There $z_b[n]$ is white noise and $z_d[n] = a_i h[n - n_i]$

$$z[n] = z_d[n] + z_b[n] \tag{5.23}$$

This is visualized in figure 5.1. There, the impulse response of the matched filter $G(z)$ is $g[n]$ Now, a signal greater than $\theta$ is shot. This is an outlier.

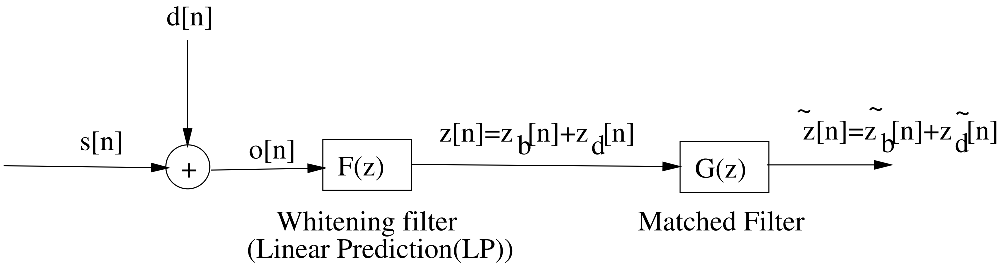


Figure 5.1: Signal whitening and matched filtering for shot noise

Applying the Schwarz inequality that is applied in matched filter we receive the signal to noise ratio $r$ in equation (5.24). We consider the signal has a finite length that is $n = 1, 2, \cdots, N$.

$$\varkappa = \frac{|z_d[n]|^2}{E\{|z_b[n]|^2\}} = \frac{1}{\sigma^2} \frac{\sum_{k=1}^{N} |g[k]d[n-k]|^2}{\sum_{k=1}^{N} |g[k]|^2} \tag{5.24}$$

Now applying the Schwarz inequality we get

$$|\sum_{k=1}^{N} g[k]d[n-k]|^2 \leq \sum_{k=1}^{N} g^2[m] \sum_{k=1}^{N} d^2[k] \tag{5.25}$$

Now in equation (5.25), $\varkappa \leq \frac{E_d}{\sigma^2}$ where $E_d = \sum_{k=1}^{N} d^2[k]$. The resulting upper bound is reached when we assume $g[k] = d[n-k]$. It is therefore the maximum for $g[k]$. Now the optimum solution is the reversed copy of $g[n]$ for the hypothesis $H_0$. In this case $d[n]$ has a finite duration $N$ that is $g[k] = d[n-k]$ in order for the filter $g[n]$ to be causal. Hence, the matched filter for our shots is in equation (5.26). This says $G(z)$ is the reverse version of $F(z)$.

$$G(z) = F(-z) = \alpha_p z^{-1} + \alpha_{p-1} z^{-2} + \cdots + \alpha_1 z^{-p} \tag{5.26}$$

In figure 5.2, we see the noisy observation $o$ in the first row, whitened transformation of the signal $s$ in the second row and third row shows the success of the matched filter in detecting the shots. In the signal, we see a detection of shots, one is close to sample 100 and another one is close to sample 500.

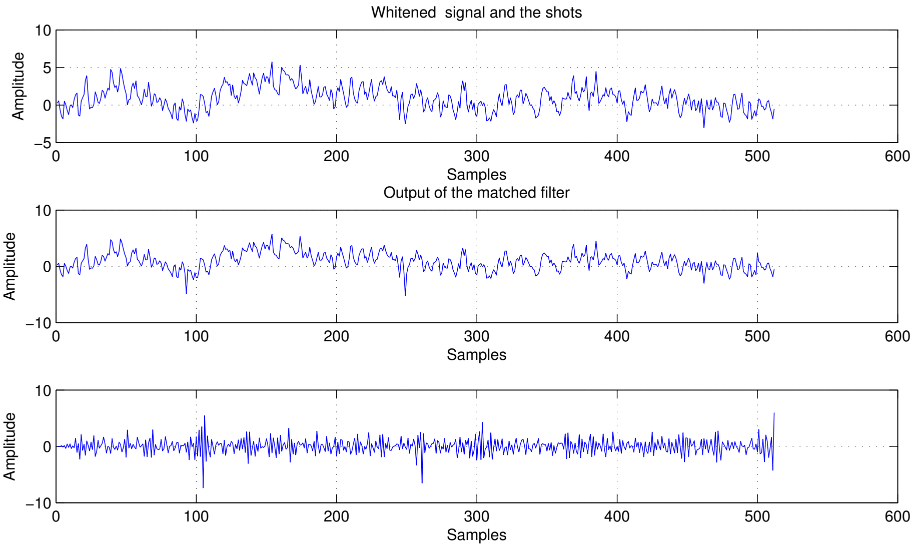


Figure 5.2: Strong noisy signal and matched filtered output

## 5.7 Actions

Now we find the treatment of the strong noise. If the speech is affected by the omission of the shots, this means that also some speech occurred in the interval may have been omitted. In this case, the system gives a feedback as indicated in section 5.1 to the speaker "Repeat".

# Chapter 6

# Source Excitation Model

**Outline of the Chapter** Here we see a relation between the physical and the numerical interpretation of the speech production model. This regards the human body as a machine and provides a computational model. We describe the model from an acoustics point of view. This is also known as a source excitation model. We find how the acoustic filter mainly consists of some cavities, namely the vocal tract, the nasal tract, the mouth and the lips but finally it is simplified to a vocal tract model only. This can represent three different excitation types, namely voiced, unvoiced and plosive. It is a linear model which is excited by a white noise. In order to make this complicated speech process simpler by reducing many variables, this model has some approximations and assumptions. These are discussed here. The model is a standard speech production model. The model analysis is based on the discrete time.

## 6.1 General Aspects

The speech is produced by an excitation source which is later transformed into different shapes by the actions of the vocal and articulatory organs. The vocal organs are vocal tract i.e. the glottis, pharyngeal tract, the vocal folds and the articulatory organs are palete, nasal tract, tongue, mouth, lips. These are mainly some cavities which generate resonances for the human speech sound production [69]. A pictorial representation of these organs can be seen in figure 6.1. We have included this figure in order to make an impression of the position and the participation of these organs. The number of parameters such as the poles and zeros are used in general for an efficient tractability. The all pole model appears to be simplest in the parametric typed vocal tract modeling and the details of

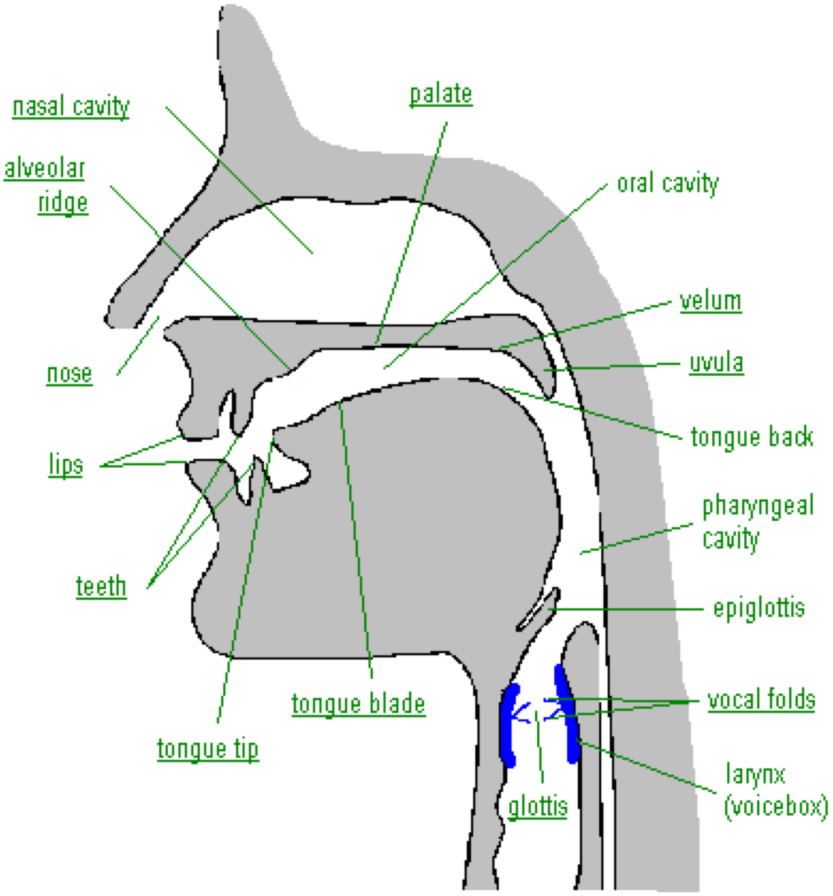


Figure 6.1: Human vocal and articulation organs [52]

this is explained in this chapter. Therefore the purpose is here to use the all pole model in order to capture the human speech production for its modeling. This chapter explains the background of the speech production modeling. Further discussions of this model and the use of this model will be seen in chapters 7, 8, 9, 10.

The analysis of the model is discussed using two approaches:

- Difference equation: To emphasize and manipulate the system using the input and output.
- Z-transform: To analyze and provide the transfer function of the vocal tract model. This is used to represent the speech production system.

More details about the difference equation and the z-transformation for the discrete time system analysis can be found in [60].

## 6.2 Analysis Speech Production Model

As mentioned initially the speech is produced by the excitation source supplied by the lungs. This then goes to the larynx. It shows preliminary acoustical shapes, namely voiced or unvoiced shapes of the source. This then goes to the vocal tract. This in combination with articulatory organs transforms the acoustical source into a speech waveform [131]. The main cavities, the vocal tract, the nasal tract, the mouth and the lips generate the speech waveform.

In the analysis the source excitation model is a discrete time speech production model. How the continuous time speech waveform $s(t)$ is converted to discrete time signals $s[n]$ is discussed in Appendix. This discrete time speech representation $s[n]$ is now our starting speech processing point. We assume that if the continuous time to discrete time (C/D) conversion is processed properly (see Appendix). The information we lose in the C/D conversion is negligible. Properly means if we follow the Nyquist theorem for sampling the signal, then the signal can be reconstructed to its original form.

The major aspects of the speech production model are listed below. All these concepts in the model are intended to describe the model in reality.

- The speech is first excited by some source and this source is a white noise.
- The final element of the source excitation model is the vocal-tract model.

- In the vocal-tract system, the speech sound is produced by opening and closing of the vocal folds. This introduces a vibration in the system. The opening and closing rate of the vocal folds varies from person to person.

- The articulatory features and events associated with the production of the sequence facilitates the continuous-time acoustic speech waveform in the discrete-time source excitation model.

There is one difference between the model provided in figure 2.1 and the source excitation model provided next. In figure 2.1, the vocal-tract has a continuous input and output while the excitation model has discrete input and output. The purpose of this is to reconstruct the speech and to make the manipulations and computations easier. The discrete speech production model is efficient to represent the physical speech production process [80]. We have used it in our study.

The vocal tract system takes a continuous input which comes from the excitation source and produces a periodic airflow as an output that is not linear. The reason is that the glottis is not linear. If the glottis were a linear system, then a constant input would yield a constant output and all the speech sounds would always be same. We have followed a linear signal model to capture the vocal tract information.

### 6.2.1 Assumptions

First we list some facts of the speech production model. They are taken from the literature and not questioned here [80], [131], [69], [30].

- The excitation mechanism of the speech production system: There is an input to initiate a process and this is the excitation.

- The operation of the vocal-tract system: The vocal-tract is playing a significant role in deciding what to keep or discard to generate the speech.

- The lip and the nasal radiation process: The lips and the nasal radiation process give the final emphasis on the speech generation.

- Voiced and unvoiced speech: The vocal-tract changes its shape at a short time interval to generate different phonemes or finally the speech. If the vocal-tract had not changed its shape to generate the speech, all the phonemes or the speech would have been the same. The question is how to decide the time interval where the shape of the vocal-tract changes. These change at every 10 to 30 ms [66], [30]. The interval should not be too small that we can not capture the dynamic behavior of the speech signal and the interval should not be too large that we miss the dynamic changes of the speech.

- For the voiced speech, the excitation source is periodic: The excitation source for the voiced speech is a train of pulses modified by some factor which may be seen gain for the volume controller for the voiced speech and the transformation or the modification of this source is taking place in the vocal tract associated with the articulatory organs.

- The excitation is small or close to zero at its time period except at the beginning of the pitch period: The excitation is actually originated at the beginning of each pitch period to keep the process going and in between the pulses, the excitation is assumably zero.

- The excitation takes place in the lungs and generates speech waveforms when it passes through the vocal-tract. The speech signal is globally non stationary but it is locally stationary or quasi-stationary.

- The excitation source $u$ is a random white Gaussian process. The weight or the gain in the model is the loudness of the sound. This depends on the amount of the air pressure or the excitation source coming out the lungs. The gain factor is unique for a speaker and for a speech. Some other basic terms used in the model are for example pitch period and formant.

- The formant is the resonant frequency of the vocal-tract. The formants help to signify the opening and closing phenomena of the vocal folds. It is denoted as the fundamental frequency in the speech production model. The pitch period is reciprocal to the formant frequency.

## 6.3 Source Excitation Types and Formulations

The source excitation model initially considers an excitation source for three different types that are sometimes named differently: The excitation source can be periodic or voiced, or random or unvoiced, or plosive or impulsive.

In figure 6.2, each source namely voiced, unvoiced and plosive source is multiplied by a factor. This changes the loudness of the speech. It varies and it changes according to the speaker and the speech spoken by the speaker. In figure

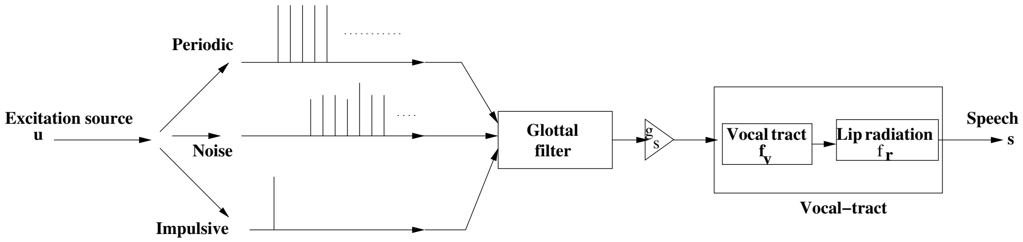


Figure 6.2: Source-excitation speech production model

6.2, the source $u[n]$ where $n \in \mathcal{Z}$ is the excitation source for the voiced, unvoiced and plosive speech. The source denoted by $u$ is characterized by following three main excitation types :

- Periodic Pulse: This produces the voiced speech
- Random Pulse: This generates the unvoiced speech
- Impulsive Pulse: This produces the plosive speech.

In the following explanation, the source is always denoted by the same notation $u$. Because the source can be only one typed. This means, the source $u$ can be either voiced, or the unvoiced or the plosive. Therefore, in our consideration, a single source notation $u$ is reasonable.

### 6.3.1 Voiced Speech Source

The source of the voiced speech is essentially periodically patterned. This means the pattern of the voiced speech does not have a precise periodic pattern. This is termed as a quasi-periodic The voiced speech mainly the English vowel such as "a", "e","i". The excitation source is formulated in equation (6.1) and shown in figure 6.3. In equation (6.1), unit pulse denoted by $\delta[n]$ is delayed by $k$. The pitch period $k$ is a difference between two pulses. In the equation, $i \in \mathcal{Z}$ and $\mathcal{Z}$ denotes integer number. The notations $n$ and $k$ should not be confused with the notations used using the same in other chapters.

$$u[n] = \sum_i \delta[n - ik] \quad \text{for the voiced case} \tag{6.1}$$

The voiced typed signal flow is shown in figure 6.4. The excitation source $u[n]$ in equation (6.1) goes to the glottal filter $f_g[n]$. This is then modified by the factor $g_s$ and generates the output $g_g[n]$. This then goes to the vocal-tract filter $f_v[n]$

and generates $v_f[n]$. Then it goes to the lip radiation filter $f_r[n]$ and generates the voiced speech $s[n]$. The process is shown also in figure 6.2. The formulation of the periodic pulse source and its voiced speech production is shown in equation (6.2). In figure 6.2, each line at the left corner indicates a unit pulse which is weighted by a factor $g_s$. In equation (6.1), "." denotes mulitplication. In figure 6.4, the vertical lines are the pulses. There it is assumed to be periodic and they are repeated in a periodic manner. $u[n]$ is the function of $n$ on the x-axis and the amplitude of $u[n]$ is on the y-axis where $n = 0, 1, 2, 3, \cdots$. The downward vertical arrow $g_s$ is a weight. The circle with "x" denotes the multiplication sign. The sign "$\otimes$" is the symbol of the convolution sum.

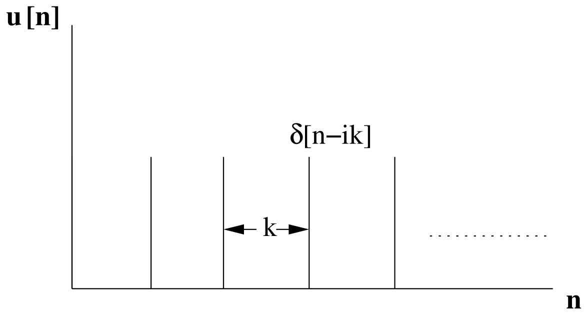


Figure 6.3: Excitation source of the voiced speech

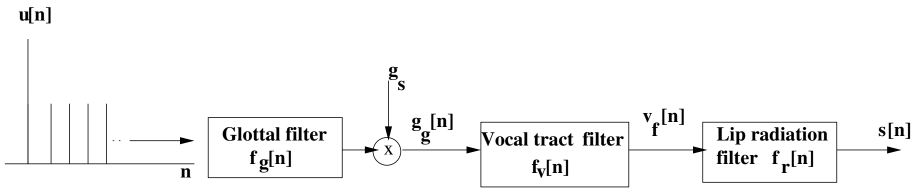


Figure 6.4: Voiced speech in source-excitation model

$$\begin{aligned} g_g[n] &= g_s.(f_g \otimes u)[n] \\ v_f[n] &= (f_v \otimes g_g)[n] \\ s[n] &= (f_r \otimes v_f)[n] \end{aligned} \tag{6.2}$$

### 6.3.2 Unvoiced Speech Source

The excitation source of an unvoiced speech is a random white Gaussian noise formulated in equation (6.3). In the equation, $w[n]$ is the white Gaussian noise. Its mean is zero and its variance is one. The English alphabet "F", "V" are some examples of the unvoiced speech. Here, the excitation source $u[n]$ defined in equation (6.3) comes from the lungs and the larynx. Then a multiplication of the source by a factor $g_s$ goes to the vocal-tract and the lip radiation filter and generates the unvoiced speech $s[n]$.

$$u[n] = w[n] \quad \text{for unvoiced case} \tag{6.3}$$

The transformation of the unvoiced speech is shown in figure 6.5. The figure 6.5 shows the source $u[n]$ is multiplied by $g_s$. This is then the input to the vocal-tract filter and generates $v_f[n]$ and the lip radiation filter $f_r[n]$ to generate the unvoiced speech $s[n]$. The formulation of the unvoiced speech is shown in equation (6.4). The glottal filter has no influence on the generation of the unvoiced speech. In equation (6.4), the unvoiced speech $s[n]$ is the output of the vocal-tract filter and the lip radiation filter for the weighted source $u[n]$.

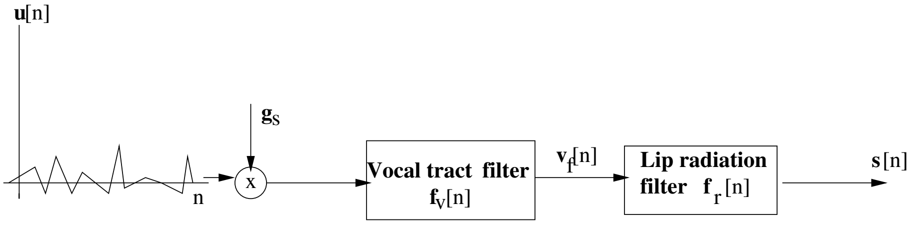


Figure 6.5: Unvoiced speech in source-excitation model

$$\begin{aligned} v_f[n] &= (g_s.u \otimes f_v)[n] \\ s[n] &= (f_r \otimes v_f)[n] \end{aligned} \tag{6.4}$$

### 6.3.3 Plosive Speech Source

As mentioned in section 6.2, the excitation source $u[n]$ is generated from the lungs. It is weighted by the gain $g_s$ and then goes to the vocal-tract filter and the lip radiation to generate plosive speech. Examples of such speech are "B", "P". The plosive source is an impulse formulated in equation (6.5) and it is 1 at

$n = 0$. With the use of $\delta[n]$ as the unit pulse we get:

$$u[n] = \delta[n]$$

This says:

$$u[n] = \begin{cases} \delta[n] = 1 & \text{for } n = 0 \\ 0 & \text{for } n \neq 0 \end{cases} \tag{6.5}$$

In figure 6.6, also shown in figure 6.2, the source is influenced by $g_s$ which is a volume controller and produces $v_f[n]$ when it goes through the vocal-tract filter $f_v[n]$. Finally, $s[n]$ is the response of $v_f[n]$ of the lip radiation filter $f_r[n]$.

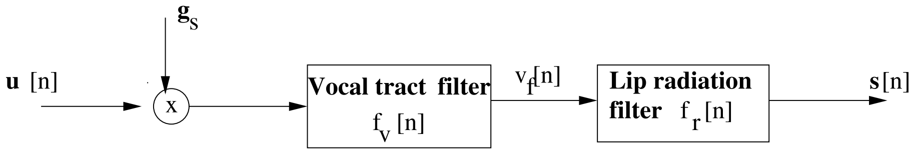


Figure 6.6: Plosive speech in source-excitation model

In figure 6.6, the only horizontal line at the left is the impulse signal. This is denoted by $u[n]$. This is weighted by the factor $g_s$ and goes to the vocal tract filter and lip radiation filter in order to generate the plosive speech. The mathematical formulation of this in equation (6.6).

$$\begin{aligned} v_f[n] &= (g_s.u \otimes f_v)[n] \\ s[n] &= (f_r \otimes v_f)[n] \end{aligned} \tag{6.6}$$

Next we will explain the vocal tract filter, glottal filter and the lip radiation filter in order to explain the simple speech production model.

## 6.4 Systems of the Source Excitation Model

In this section we define the glottal filter, the vocal tract filter and the lip radiation filter. These are used in defining the source excitation model using the vocal tract only. The definitions are given in z-domain in order to emphasize the system design using pole and zero as well as the transfer function.

In figure 6.7, we see the source is going through the glottal filter, vocal tract filter and lip radiation filter. These filters are shown here.

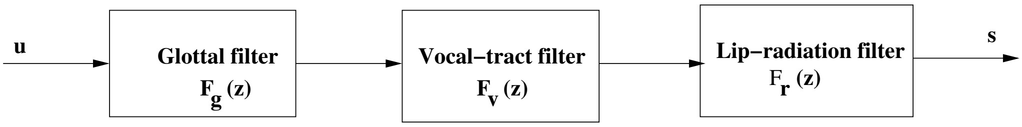


Figure 6.7: Speech Production Systems

### 6.4.1 Glottal Filter

In equation (6.7), $F_g(z)$ is the transfer function in the z-domain and this represents the glottal filter $f_g[n]$. In equation (6.7), $\beta = e^{-cT}$ and $c$ is the speed of the sound and $T$ is the sampling interval length or sampling period. The concept behind this is $cT << 1$ and thus $e^{-cT} \approx 1$.

The glottal filter given in equation (6.7) is a second order low pass filter, $F_g(z)$.

$$F_g(z) \equiv \frac{1}{(1-\beta z^{-1})} \frac{1}{(1-\beta z^{-1})} \equiv \frac{1}{(1-\beta z^{-1})^2} \equiv \frac{1}{(1-z^{-1})^2} \qquad (6.7)$$

### 6.4.2 Vocal-tract Filter

The shape of the vocal-tract changes slowly when it produces different kinds of sounds. This filter varies according to a speaker and the speech sounds spoken by the speaker. The transfer function of the vocal-tract system is given in equation (6.8). We see equation (6.8) is an all pole filter with $p$ many poles.

$$F_v(z) = \frac{1}{1 - \sum_i^p a_i z^{-i}} \qquad (6.8)$$

### 6.4.3 Lip Radiation Filter

In equation (6.9), $F_r(z)$ is the z-transform of $f_r[n]$. It is responsible for the speech sound that comes through the lips. In equation (6.9), the lip radiation filter has a single zero. and $a_r$ takes the value close to 1. The role of pre-emphasis filter discussed in chapter 4 in section 4.3. This takes over by the lip radiation filter.

$$F_r(z) = 1 - a_r z^{-1} \qquad (6.9)$$

Next we will explain how the speech production model is represented by using only the vocal-tract system.

## 6.5 Source Excitation Model using Vocal-tract

In the source excitation model the excitations for the voiced, and unvoiced are considered. The excitation source for the unvoiced speech and plosive speech are considered as white random noise. Therefore the source excitation model has only two main excitation types: i) A train of pulses for the voiced speech and ii) White noise for the unvoiced speech.

In figure 6.8 we see that the excitation source responsible for the voiced or the unvoiced speech is going through a time varying linear filter to generate the speech output which can be a voiced speech or an unvoiced speech. The time varying linear filter is a combination of the glottal filter, the vocal tract filter and the lip radiation filter but all these are represented by by a transfer function and this is known as the vocal model. How all these filters namely the glottal filter, the vocal tract filter and the lip radiation filter are manipulated to replace these by only the vocal tract filter is the discussion of this section. The vocal tract filter is the most common speech production model used in the speech research. This conventional speech production model is an all pole model. In this model, the system for the voiced and unvoiced speech output is modeled by poles only.

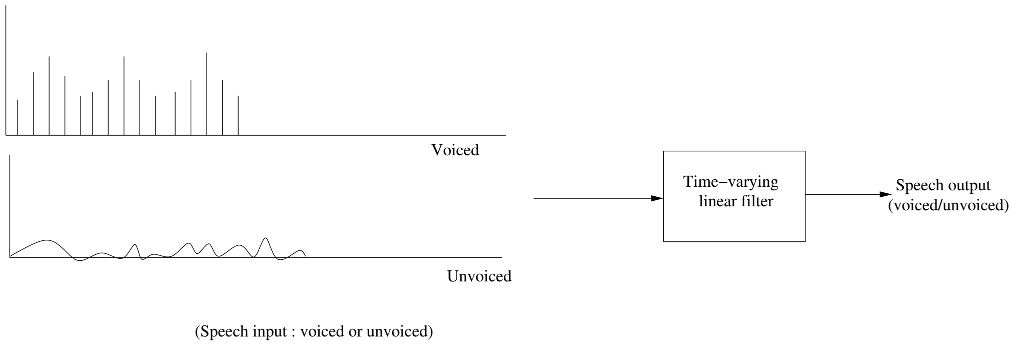

Voiced
Unvoiced
(Speech input : voiced or unvoiced)
Time−varying linear filter
Speech output (voiced/unvoiced)

Figure 6.8: Stochastic source-excitation model

The lip radiation filter has a zero. Then the glottal filter modeled by two first order low pass filters has two poles. The lip radiation represented by the high pass filter has one zero and this cancels the spectral effect of one of the glottal poles in case they are matched.

A presence of a second zero near $z = 1$ would effectively eliminate the spectral effects of the larynx and the lips. In such a case, the analysis could mainly be focussed on the coefficients of the vocal-tract parameters only. The pre-emphasis filter discussed in chapter 4 in section 4.3 is a first order high pass filter and it has only one zero. This has one coefficient for the zero which is less than 1. The pre-emphasis filter boosts the high frequency of the formants which have been suppressed in the high frequency region by the glottis. The glottis is a vocal organ which contains the vocal folds and the opening space in between the vocal folds.

We use $U(z)$ and $S(z)$. These are the z-transforms of $u[n]$ and $s[n]$. The glottal filter ($F_g(z)$) has no contribution in generating the unvoiced and the plosive speech. Therefore, different from figure 6.2, the effect of the glottal filter is not considered.

**Voiced Case** The voiced speech is generated when the vocal-tract is excited by a series of periodic pulses. The variation in the voiced speech is very smooth within a period. For this reason, it is analyzed as an essentially periodic signal. The vocal tract filter for the voiced source given in equation (6.10) is a multiplication of the gain $g_s$ and the transfer functions of the glottal filter $F_g(z)$, the vocal-tract filter $F_v(z)$, and the lip radiation filter $F_r(z)$. As mentioned $a_r$ is close to 1. Thus we simplified equation (6.10) to equation (6.11) which has $p+1$ number of poles.

$$\begin{aligned} H(z) &= \frac{S(z)}{U(z)} = g_s.F_g(z)F_v(z)F_r(z) \\ H(z) &= g_s.\frac{1}{(1-z^{-1})^2}.\frac{1}{1-\sum_{i=1}^{p} a_i z^{-i}}.(1-a_r z^{-1}) \end{aligned} \tag{6.10}$$

$$H(z) = \frac{g_s}{1-\sum_{i=1}^{p+1} a_i z^{-i}} \tag{6.11}$$

**Unvoiced Case** The vocal-tract filter for the unvoiced source is formulated in equation (6.12). We used the z-transformation of the system and replaced the values of the system discussed above in equation (6.12). The glottal filter remains open which means it is not participating in generating the unvoiced speech.

$$\begin{aligned}H(z) = \frac{S(z)}{U(z)} = g_s F_v(z) F_r(z) \\ H(z) = g_s . \frac{1}{1 - \sum_{i=1}^{p+2l} a_i z^{-i}} .(1 - a_r z^{-1})\end{aligned} \tag{6.12}$$

For the unvoiced sound there is an effect from the nasal sound which is realized mainly by several zeros. In this manipulation, the zeros are replaced by placing more poles in order to make the model all pole only [10].

A zero can be replaced by two poles if the magnitude of the zero is small enough for example less than 1. This assumption is used in equation (6.13) to avoid the zero in the modeling. Thus equation (6.12) is simplified to equation (6.13) .

$$\begin{aligned}H(z) = \frac{S(z)}{U(z)} = g_s . \frac{1}{1 - \sum_{i=1}^{p+2l} a_i z^{-i}} .(1 - a_r z^{-1}) \\ H(z) = \frac{g_s}{1 - \sum_{i=1}^{p+2l+2} a_i z^{-i}}\end{aligned} \tag{6.13}$$

In equation (6.13), we have $2l$ poles for $l \in \mathcal{Z}$ and we have $l$ many zeros for an effect of the nasal source. In equation (6.13), 2 is used for the zero in the lip radiation filter. By replacing all zeros with poles, we arrive at equation (6.13). This is now the vocal-tract system with poles only. Therefore, it is called an all pole filter. The all pole model is a simple parameter estimation model between an all zero and a pole zero model because the relation of the pole coefficients and the autocorrelation function yields a set of simultaneous linear equations and the estimation of the parameters of the all pole model can be performed by computing the estimates of the autocorrelation terms [10]. This is discussed in chapter 8.

By taking a sufficient number of poles, the overall transfer function shown in equation (6.16) for the voiced and unvoiced case is rewritten in equation (6.15).

$$H(z) = \frac{S(z)}{U(z)} = \frac{g_s}{1 - \sum_{i=1}^{p+2l+2} a_i z^{-i}} \tag{6.14}$$

Equation (6.15) is now the speech production system that represents the vocal tract in order to generate voiced and unvoiced speech. Thus we obtain a single transfer function with poles only and this includes the voiced, unvoiced and plosive speech.

$$H(z) = \frac{g_s}{1 - \sum_{i=1}^{p} a_i z^{-i}} \tag{6.15}$$

This is an all pole filter and the pictorial definition is in figure 6.9. There we see that the weighted source $g_s u$ generates $s$ through the system $\sum_{i=1}^{p} a_i z^{-i}$.

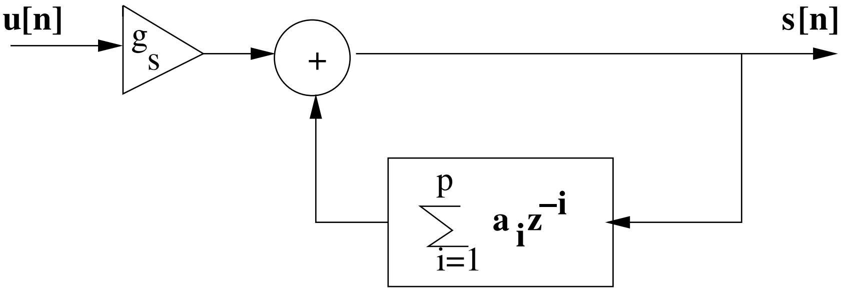


Figure 6.9: Simplified speech production model

$$H(z) = \frac{S(z)}{U(z)} = \frac{g_s}{1 - \sum_{i=1}^{p+2l+2} a_i z^{-i}} \tag{6.16}$$

Equation (6.15) is known as auto-regressive (AR) model. It is most commonly used for a parametric signal modeling. This is the background of the speech production model by the AR process. This is described in chapter 7.

In equation (6.15), $a_i$ are unknown. We need a technique to find out the values of $a_i$. For this, we use a linear prediction (LP). This is discussed in chapters 8 and 9.

# Chapter 7

# Vocal- tract Model: AR Model

**Outline of the chapter** In this chapter we discuss auto-regressive (AR) parametric signal modeling. It contains mainly known facts and relations. We repeat them here because they are used several times in this thesis. In chapter 6 we have seen that the source excitation model is modeled by an AR process. This model can be stochastic and deterministic. The source excitation model is in fact the stochastic typed AR model. We introduce this here. The model parameters have to be solved and this is discussed in the next chapters 7 and 8.

We first introduce the notion of parametric signal modeling. For this we read namely[109], [63], [92], [8], [10].

## 7.1 Analysis of Parametric Signal Modeling

A non-stationary signal such as the speech signal is generally analyzed in a small segment. This small section can be modeled by using a parametric signal model or by a non-parametric signal model. In a parametric signal modeling this small segment is modeled by some parameters. These parameters may change from segment to segment. This generally happens when a non-stationary signal such as one from speech is modeled. In this modeling, most often a complicated process such as a speech signal can be represented by a smaller number of parameters than the actual samples in the signal. The parameters capture changes or dynamics of the signal. That means the signal parameters reflect the changes of the signal. The reduction of the parameters often requires an approximation, estimation and some constraints or some additional information. A common approximation is that the system is driven by some known input where the input is most often assumed as a unit sample signal or white Gaussian noise. On the other hand, in the non-parametric signal modeling, the signal is most often characterized by the measurements of the frequency response and this may be a large number of frequencies. Examples of such type of modeling approach are overlap-add and the overlap-save methods [9]. The optimal spectrum of the excitation in the parametric case will be different from that in the non-parametric case: this is principally because the parametric model combines the information available from all frequencies in only a few parameters. In a direct non-parametric frequency response measurement there is no relation between the measurements at the various frequencies and therefore the excitation should be designed to achieve a predefined accuracy in the frequency bands of interest. An example is maximizing the absolute or relative accuracy of the measurements. In a parametric approach, the energy will be concentrated on the frequencies where it contributes most to the knowledge about the model parameters. We paraphrased the above based on our studies given in [109], [63], [92], [8], [10]. For this some factors are [63]:

- Model type
- Model order
- Approach to estimate model parameters

According to the list we specify our parametric model as: The AR typed model is discussed next in this chapter, the model order is discussed in chapter 2 in section 8.1.1 and least squares (LS) approach is discussed in chapter 8.

The AR parametric signal modeling assumes that some excitation source, for example the white Gaussian Noise (WGN) $u$ generates some random output $s$ through some system $h$. The goal is now to estimate the parameters of the system $h$ which has the input $u$ and the estimated output $\hat{s}$. We want that the difference between $s$ and $\hat{s}$ is minimal.

The AR parametric signal model can be stochastic and deterministic. We only discuss the stochastic AR parametric signal model for the speech production model.

## 7.2 Overview: Auto-regressive (AR) Model

The basic AR parametric modeling is based on the auto-regressive moving average (ARMA) parametric modeling. This is visualized in figure 7.1 The basic parametric modeling, namely the auto-regressive (AR) model, the moving average (MA) model and the ARMA model uses mostly the least squares criteria in order to estimate the model parameters. Here we briefly introduce the AR, MA and ARMA parametric modeling in an overview but we have used only the AR parametric modeling and this is mainly discussed in this chapter. The AR modeling is commonly used for the signal modeling because it is easily tractable for the parameter estimation [10]. This is also discussed in chapter 6. In the parametric modeling, given a set of time series observations, we determine the parameters that generate the series. For this we need to obtain the best estimate of the parameters that can closely replicate the process. The best estimation of this modeling most commonly uses the least squares criteria. The details can be found in [92], [10].

Suppose the speech $s[n]$ is a response of a system $h[n]$ which has the excitation $u[n]$. This means $u[n]$ is the input, $s[n]$ is the output and $h[n]$ is the system. If we assume $u[n]$ is white Gaussian noise and $h[n]$ is modeled by the ARMA model defined in equation (7.1). The model has two parts: One is for AR model which order is $p$ and another is for MA model which orderis $q$. For the white noise as input $u$, the system $h$ generates $s$. We describe this now in the z-domain where $S(z)$, $U(z)$ and $H(z)$ are the z-domain representation of $s[n]$, $u[n]$ and $h[n]$.

$$H(z) = \frac{B(z)}{A(z)} = \frac{1 + \sum_{i=1}^{q} b_i z^{-i}}{1 + \sum_{i=1}^{p} a_i z^{-i}} \tag{7.1}$$

The expansions of $B(z)$ and $A(z)$ are given in equations (7.2) and (7.3)

$$B(z) = 1 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_q z^{-q} \tag{7.2}$$

$$A(z) = 1 + a_1 z^{-1} + \cdots + a_p z^{-p} \tag{7.3}$$

The input-output relation is given in equation (7.4).

$$s[n] + a_1 s[n-1] + \cdots + a_p s_[n-p] = u[n] + b_1 u[n-1] + \cdots + b_q u[n-q] \tag{7.4}$$

$$s[n] + \sum_{i=1}^{p} a_i s[n-i] = u[n] + \sum_{i=1}^{q} b_i u[n-i] \tag{7.5}$$

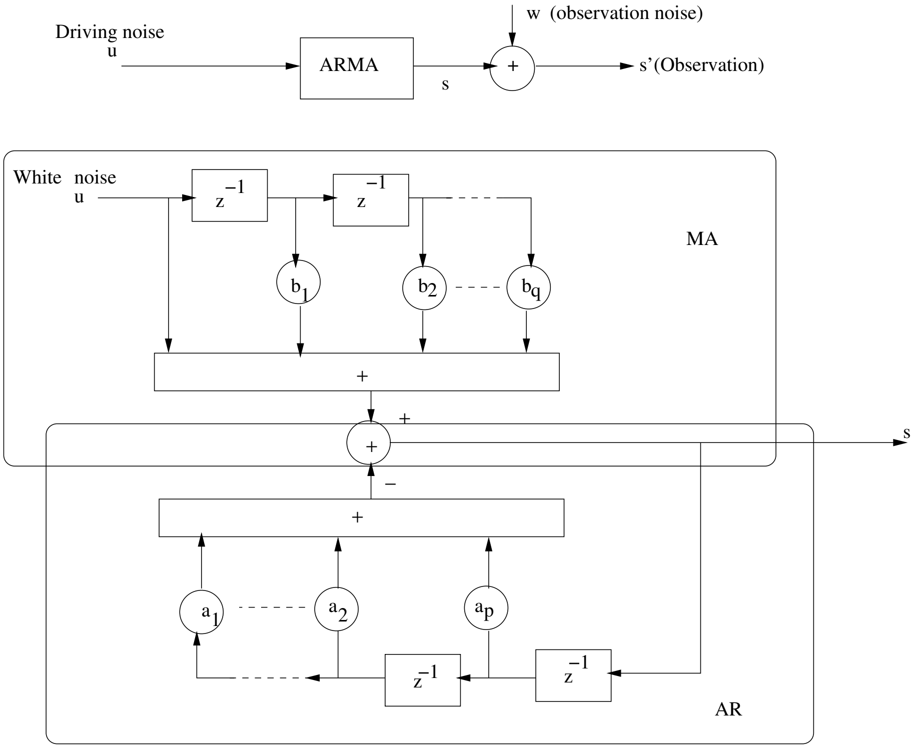


Figure 7.1: Moving average autoregressive ( ARMA) filter and perceptional site [121]

Taking the z-transform of equation (7.5), we arrive at equation (7.6).

$$S(z) + \sum_{i=1}^{p} a_i S(z) z^{i} = U(z) + \sum_{i=1}^{q} b_i U(z) z^{-i} \tag{7.6}$$

$$S(z)(1 + \sum_{i=1}^{p} a_i z^{-i}) = U(z)(1 + \sum_{i=1}^{q} b_i z^{-i}) \tag{7.7}$$

$$\frac{S(z)}{U(z)} = \frac{1 + \sum_{i=1}^{q} b_i z^{-i}}{1 + \sum_{i=1}^{p} a_i z^{-i}} \tag{7.8}$$

$$H(z) = \frac{S(z)}{U(z)} = \frac{1 + \sum_{i=1}^{q} b_i z^{-i}}{1 + \sum_{i=1}^{p} a_i z^{-i}} \tag{7.9}$$

Thus we have the following equation which is the same that is shown in equation (7.1).

$$H(z) = \frac{B(z)}{A(z)} = \frac{1 + \sum_{i=1}^{q} b_i z^{-i}}{1 + \sum_{i=1}^{p} a_i z^{-i}}$$

If $b_i$ equals zero for all $i$, then the ARMA system given in equation (7.1) represents the system of the all pole model written in equation (7.10), but if $a_i$ equals zero for all $i$, then the ARMA system represents the system of the all zero model written in equation (7.11). Thus the ARMA modeling can be seen as a combination of the MA and the AR modeling even one can make use of the AR, MA and ARMA model independently for the signal model. An overview of ARMA is shown in figure 7.1 where we see $s$ is reposed of the ARMA system to the random white noise input. The $s$ is corrupted by some observation noise and generates $s^{'}$.

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 + \sum_{i=1}^{p} a_i z^{-i}} \quad \text{for} \quad b_i = 0 \tag{7.10}$$

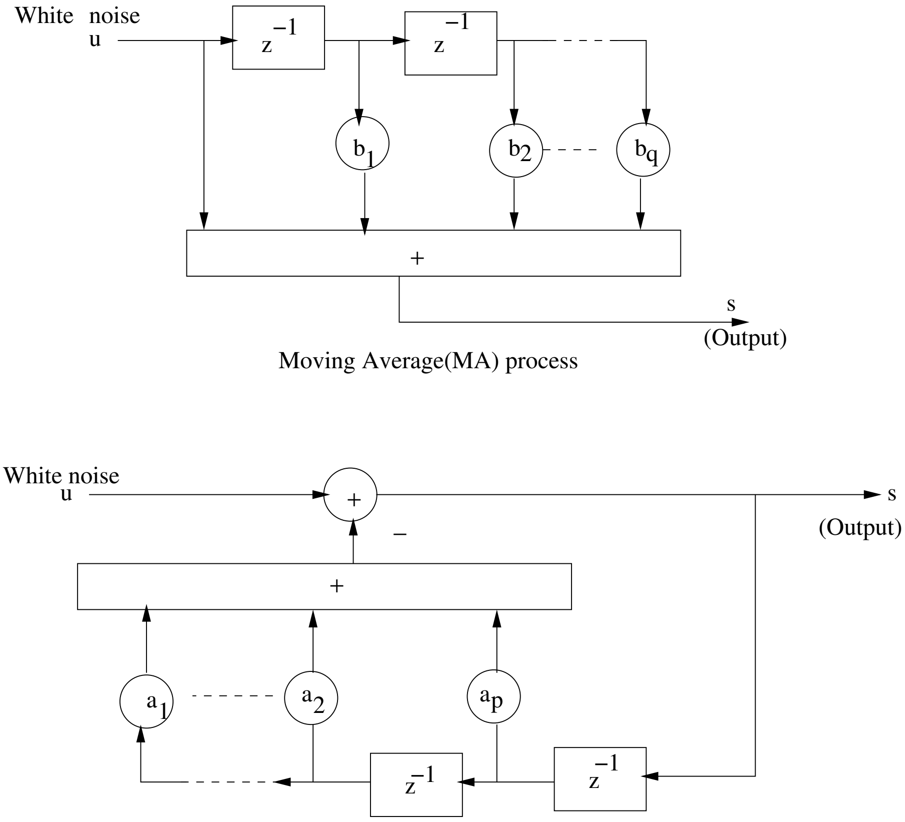
$$H(z) = B(z) = 1 + \sum_{i=1}^{p} b_i z^{-i} \quad \text{for} \quad a_i = 0 \tag{7.11}$$

The polynomials of $A(z)$ and $B(z)$ given in equations (7.2) and (7.3) are characterized by the location of the poles and the zeros in the z-domain, then the ARMA model has $p$ and $q$ many poles and zeros in $A(z)$ and in $B(z)$. Therefore, $A(z)$ and $B(z)$ are called an all-pole model and all-zero model. This gives a background and an overview of AR modeling.

An overview is shown in figure 7.1. On the top of figure 7.1 we see a very general description. The upper half describes MA and the lower half describes AR. The individual section of the ARMA filter is shown in figure 7.2. As said we use the AR model for our signal modeling, we explain this in details next.

## 7.3 Analysis of Stochastic AR Process

The speech signal changes with time and its different phonemes have different characteristics in the waveform. The phoneme is the fundamental unit of the sound. In equation (7.12), we see that the speech signal $s[n]$ is a linear combination of its past $p$ samples and the excitation $u[n]$ multiplied by a weight



Moving Average(MA) process

Autoregressive (AR) process

Figure 7.2: Moving average (all-zero MA filter) and auto-regressive (all-pole AR filter) [121]

$g_s$.

$$s[n] = \sum_{i=1}^{p} a_i s[n-i] + g_s u[n] \tag{7.12}$$

Now if we take a z-transform of equation (7.12), we arrive at equation (7.15).

$$S(z) = \sum_{i=1}^{p} a_i S(z) z^{-i} + g_s U(z) \tag{7.13}$$

$$S(z)(1 - \sum_{i=1}^{p} a_i z^{-i}) = g_s U(z) \tag{7.14}$$

$$\frac{S(z)}{U(z)} = \frac{g_s}{1 - \sum_{i=1}^{p} a_i z^{-i}} = \frac{g_s}{A(z)} \tag{7.15}$$

When we compare equation (7.10) and equation (7.15), we see the difference between the two equations is the factor $g_s$. A difference is also in the sign between equation (7.10) and equation (7.15); that is the plus (+) or minus (−). This is not making any big difference because the response is mainly depending on the value of the coefficients. This says that the input and output are not the same in the two equatiosn (7.10) and in equation (7.15). In the later equation there is an additional factor. Equation (7.15) is a stochastic AR modeling. We can say equation (7.10) is a deterministic AR modeling.

In equation (7.12), we see that the speech signal $s[n]$ is equal to the linear combination of its past $p$ samples and the excitation $u[n]$ is multiplied by a weight $g_s$. In the stochastic sense, the second term in the equation (7.12) is a disturbance or error. The first term i.e. the sum in equation (7.12) is a linear estimate of $s$ denoted by $\hat{s}[n]$. But if $g_s u[n]$ is zero in equation (7.12), then $s[n]$ is equal to its approximated prediction $\hat{s}[n]$. In such case if we know the coefficients $a_i$, then $s[n]$ is equal to $\hat{s}[n]$.

According to the source excitation model described in chapter 6 in section 6.2.1, at the beginning of the pitch period between the pitch pulses, the excitation is zero, therefore in equation (7.12) $g_s u[n]$ is zero and $s[n]$ can be approximately equal to its predicted value $\hat{s}[n]$.

$$s[n] = \sum_{i=1}^{p} a_i s[n-i] = \hat{s}[n] \tag{7.16}$$

But the value of the $a_i$ is unknown and we need to find the solutions of $a_i$. We obtained the value of $a_i$ by using linear prediction(LP) analysis. This is discussed in chapter 8.

In equation (7.16), it is shown that the actual value of $s[n]$ is equal to the approximated $\hat{s}[n]$ because there is no excitation between the pitch pulses and the approximate $\hat{s}[n]$ is the linearly weighted summation of the past $p$ samples. From equation (7.12) and equation (7.16), we find equation (7.17).

$$\begin{aligned} s[n] &= \hat{s}[n] + g_s u[n] \\ g_s u[n] &= s[n] - \hat{s}[n] \end{aligned} \tag{7.17}$$

Now we can say if the prediction of $s[n]$ is correct, then in equation (7.17) $g_s u[n]$ is zero. That means we have the best prediction. In the linear prediction sense, $g_s u[n]$ is termed as the error $e[n]$. Therefore, equation (7.17) can be rewritten by equation (7.18). This says the error is the difference between the actual sample and the predicted sample.

$$e[n] = s[n] - \hat{s}[n] \tag{7.18}$$

In the least squares criterion, the total error is minimized by taking the expectation of the square errors. It is described in defined in chapter 8 in section 8.2.

## 7.4 Analysis between AR and LP filters

The AR coefficients $a_1, a_2, \cdots, a_p$ are the parameters of the vocal tract. One common approach to estimate these unknown AR parameters is by using the linear prediction (LP). Thus the LP can be seen here as a synthesis filter or an inverse filter and the AR modelling in the vocal tract can be seen as an analysis filter. This relation is shown in figure 7.3 and this relation is known as deconvolution.

If a system is cascaded by two systems, the second system often can recover the first system; the action of the cascaded system is called deconvolution [96]. The output of the all pole filter $s[n]$ is the input to the LP filter. The output of the LP filter is then the error signal $e[n]$. This is shown in figure 7.3 where the LP filter $H_l(z)$ is acting as an inverse filter of the AR filter $H(z)$. This is again discussed in chapter 8 in section 8.2.1 and the $H(z)$ and $H_l(z)$ are shown again in that chapter in figure 8.2.

In figure 7.3, the deconvolution happens between the AR filter $H(z)$ and the

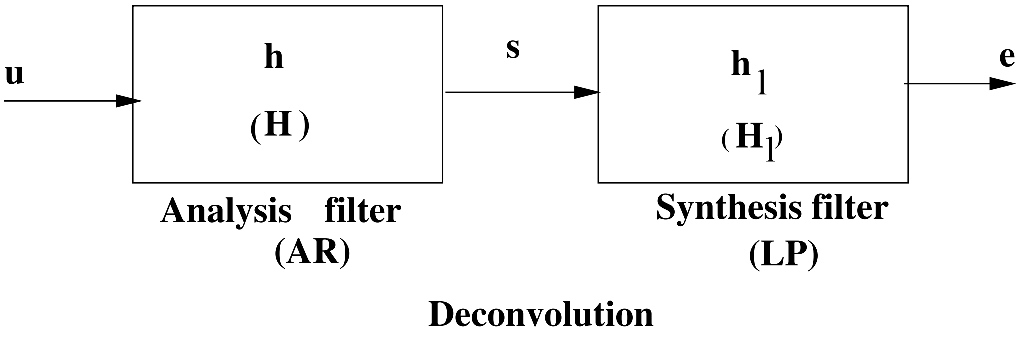


Figure 7.3: Analysis AR filter and Inverse LP filter: Deconvolution

LP filter $H_l(z)$ where the input $u$ of the AR filter generates the output $s$ and the output of the AR filter goes to the LP filter to generate the excitation input $u$ which is indirectly equal to the error $e$ signal defined in section 7.3, the output of the LP filter. The relation of $u$ and $e$ is shown in equation (7.17).

In figure 7.3, by the synthesis we mean that the original input of the system can again be estimated by the LP filter. In this synthesis or recovery process, the LP filter as a synthesis filter recovers the input of the analysis filter which is the AR filter.

The AR filter is known as an all pole filter. Conversely, we may say the LP filter is an all zero filter. But this is not a standard term. A detailed architectural view of this deconvolution process in shown in figure 7.4. This also says that the input $u$ is modified by $g_s$ and generates $s$ through the all pole filter and this output as input goes to the all zero filter and generates $u$.

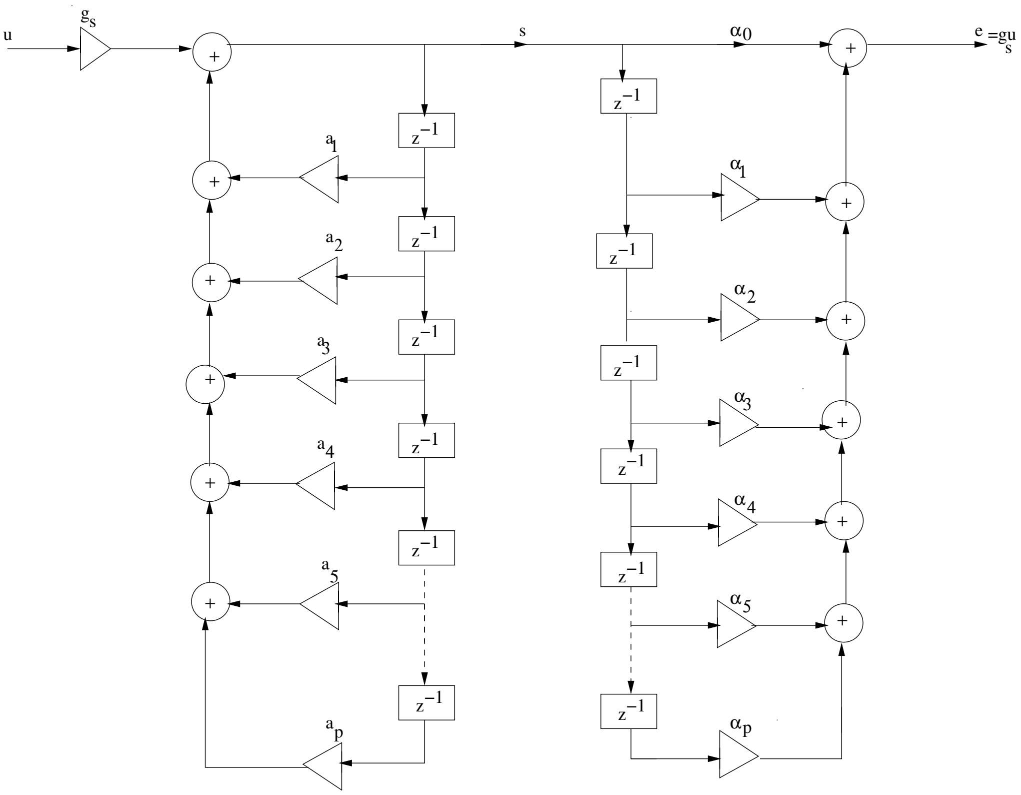


Figure 7.4: All-pole AR filter and all-zero LP filter: Deconvolution

# Chapter 8

# Estimation of AR Parameters: Linear Prediction (LP)

**Outline of the chapter** In this chapter we give a formulation of the linear prediction(LP) for the all pole modeling described in chapter 7. The LP approach uses a least squares approximation that gives a set of linear equations in order to find an approximate solution for the AR parameters. The LP approach is derived using an auto-correation approach. This is also known as the Yule-Walker approach.

The parameters of the AR process that are used for the speech production model are unknown. The relation between the AR and the LP is shown in section 7.4. The statistical properties such as mean, correlation, variance of the speech signal are used for the LP analysis to approximate the best speech parameters. The best means the predicted outcome is closest to the desired one such that the difference between the two is least. This is explained in section 8.2.

In the LP analysis the current speech sample is modeled by linear combinations of its $p$ most recent past samples. $p$ is the prediction model order. The LP analysis uses mean squared error criteria for the best predicted samples such that it is closest to the real sample.

A major part of the chapter is devoted to description of the mean squared error and its computational aspects.

Some assumptions of using LP First we mention the assumptions used in the LP based speech signal modeling.

- The excitation source and the vocal-tract system are independent from each other.

- Each excitation actuates at the beginning of each segment and remains active until the end of the segment.

- The vocal tract system is modeled using the AR process where the excitation source is white noise.
- The vocal-tract changes its shape slowly. Its characteristics changes at every 10 to 30 ms time interval.
- The parameters are computed at each short time interval at each 10 to 30 ms intervals.

## 8.1 Signal Analysis

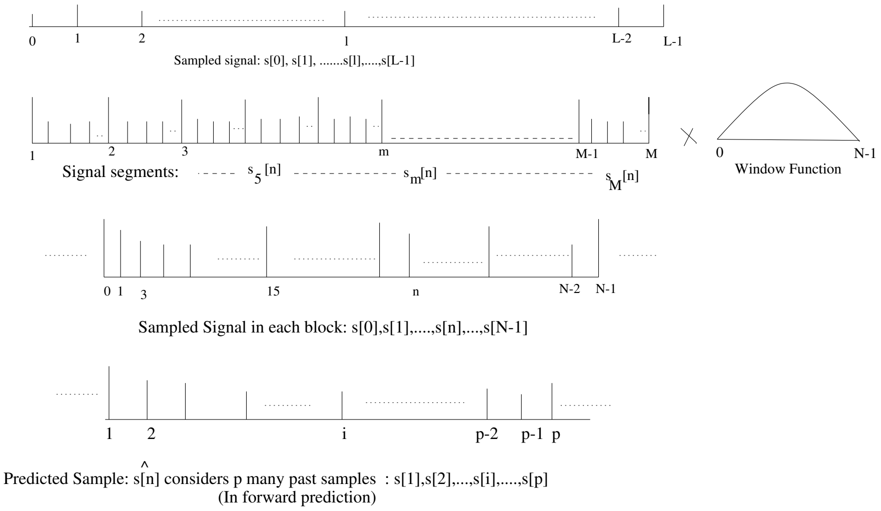

0
1
2
l
L-2
L-1
Sampled signal: s[0], s[1], .......s[l],....,s[L-1]
1
2
3
m
M-1
M
Signal segments:
s5 [n]
sm[n]
sM[n]
0
N-1
Window Function
0 1
3
15
n
N-2
N-1
Sampled Signal in each block: s[0],s[1],....,s[n],...,s[N-1]
1
2
i
p-2
p-1
p
Predicted Sample: ŝ[n] considers p many past samples : s[1],s[2],...,s[i],....,s[p]
(In forward prediction)

Figure 8.1: Short time speech signal processing

Figure 8.1 explains how the signal is processed for the analysis. Initially, the analog speech waveform $s(t)$ is digitized into $s[l]$ and $l = 0, 1, 2, 3, \cdots, L-1$. These are blocked into segments $\mathbf{s}_m$ and $m = 1, 2, \cdots, M$. Each block $\mathbf{s}_m$ has $N$ many samples as $n = 0, 1, 2, \cdots, N-1$ and each current sample $s[n]$ is modeled by linear combinations of its previous $p$ many samples. $p$ is prediction or model order. The decision of the number of model parameters is discussed in section 8.1.1. It is indexed by $i$ and $i = 1, 2, \cdots, p$. Thus in equation (8.1), $s$ is $M \times N$ dimensional. Generally a signal is blocked using a window function. In such case the signal is multiplied by a window function. This is shown in figure 8.1. The windowing is generally used to control the effect of the sidelobes in the spectral estimation [121]. A typical length of the window function is equal to the length of the signal block. However, in a covariance or Burg or ULS based LP approach, a windowing of a signal is not necessarily needed.

$$\begin{bmatrix} \mathbf{s}[0] \\ \mathbf{s}[1] \\ \vdots \\ \mathbf{s}[n] \\ \vdots \\ \mathbf{s}[L-1] \end{bmatrix}_{L\times 1} \rightarrow \text{Segemented Signals} \rightarrow \begin{bmatrix} s_1[0]\cdots s_1[n]\cdots s_1[N-1] \\ \vdots \\ s_k[0]\cdots s_k[n]\cdots s_k[N-1] \\ \vdots \\ s_m[0]\cdots s_m[n]\cdots s_m[N-1] \\ s_M[0]\cdots s_M[n]\cdots s_M[N-1] \end{bmatrix}_{M\times N} \tag{8.1}$$

Next we give a physical explanation of the order of the model.

### 8.1.1 Order of the Model

The order of the model is equal to the number of the parameters that represent the speech. We explain here the relation between the model order and vocal tract tube and how the model order and parameters are related. This is formulated in equation (8.2) [87].

The order has to be large enough to represent each formant. Similarly, the number of the coefficients needs to be sufficient to approximate the parameters of the voice articulator. Important numbers are the length of the vocal-tract, the joined structure of the nasal and oral cavities, and the excitation sources. Each formant is represented by a complex conjugate pole pair. Therefore there is the number 2 in equation (8.2) [87]. Additionally 4 represents the number of vocal tract sections. In equation (8.2), ≈ is used because this is an approximation of the model order $p$ and the precise number of involved coefficients may not be known.

$$p \approx 2 \times (\text{Number of Formants}) + 4 \tag{8.2}$$

The number of formants is the division of the Nyquist rate $f_s/2$ where $f_s$ is the sampling frequency by the average spacing of the neighboring formants $f_{n+1}$ and $f_n$ where $n$ denotes the formant number.The average distance between neighboring formants is approximated by $\frac{c}{2l}$ (see chapter 2, section 2.4.1).

The order of the model plays an important role in the modeling problem. It determines the number of parameters to be estimated and the computational complexity of the estimation algorithm. The quality of the spectral analysis is also influenced by the order of the model. If the order of the model is too low, it will display poor resolutions and if the model order is too large, it creates false spectral peaks known as spectral splittings in the spectral analysis. In such case, one single peak may be divided into two separate ones and generate some alias or misleading spectral peaks in the frequency spectrum.

## 8.2 Derivation of LP and Errors

The LP predictor coefficients are $\alpha_i$ for $i$ for $i = 1, 2, \cdots, p$. This is shown in equation (8.3); there the predicted signal of the actual $s$ is denoted by $\hat{s}$.

$$\hat{s}[n] = \sum_{i=1}^{p} \alpha_i s[n-i] \tag{8.3}$$

Now if we take the z-transforms of the terms in equation (8.3), we have equation (8.4). The symbols used in this equation are introduced in chapter 6 in section 7.3

$$\hat{S}(z) = \sum_{i=1}^{p} \alpha_i S(z) z^{-i} \tag{8.4}$$

An expanded form of equation (8.3) is:

$$\hat{s}[n] = \alpha_1 s[n-1] + \alpha_2 s[n-2] + \cdots + \alpha_p s[n-p] \tag{8.5}$$

Equation (8.5) is an expansion of equation (8.3) where it shows that the estimated speech signal is equal to the summation of the past $p$ samples which are multiplied by the weighted coefficients. Equation (8.5) is known as forward linear prediction (FLP). The backward linear prediction (BLP) is predicted using $p$ many future samples; it is defined in the next chapter. Now from equation (8.3), equation (8.5), and equation (8.7) we obtain the error in equation (8.6).

$$e[n] = s[n] - \hat{s}[n] = s[n] - \sum_{i=1}^{p} \alpha_i s[n-i] \tag{8.6}$$

Now equation (8.7) is telling us that if the prediction is almost correct, then the actual speech sample is equivalent to the predicted speech samples.

$$s[n] \approx \alpha_1 s[n-1] + \alpha_2 s[n-2] + \cdots + \alpha_p s[n-p] \tag{8.7}$$

In fact, the error may not be exactly zero since the process is stochastic but the goal is to get this error to a very close proximity to zero. It changes over time even if the signal is a quasi-stationary process.

Thus the equations (equation (8.3), equation (8.5) and equation (8.7)) are only approximately true because they use the estimated coefficients $\alpha_i$.

The error signal in equation (8.6) is the error for a single true signal and a single estimated one. The mean error and the mean squared error are given in equation (8.8) and equation (8.9). For the mean error, we take the expectation of the difference between the true sample and the predicted samples and it is computed for $n = 0, 1, \cdots, N-1$ and the mean squared error is the expectation of the squared error given by equation (8.9).

$$E(|e|) = \frac{1}{N}\{\sum_{n=0}^{N-1}(s[n] - \sum_{i=1}^{p}\alpha_i s[n-i])\} \tag{8.8}$$

$$E\{(e)^2\} = \frac{1}{N}\{\sum_{n=0}^{N-1}(s[n] - \sum_{i=1}^{p}\alpha_i s[n-i])^2\} \tag{8.9}$$

### 8.2.1 Deconvolution phenomenon

The output or the response of a system will be known if we know the input and impulse response of a system. In such cases the output will be the sum of the multiplication of the inputs with the time shifted version of the impulse responses of the system or the sum of the multiplication of the impulse responses of the system with the time shifted versions. This is the convolution. It is expressed in equation (8.10). There, $\otimes$ is a convolution symbol.

$$s[n] = \sum_{k=-\infty}^{\infty} u[n]h[n-k] = u[n] \otimes h[n] \tag{8.10}$$

The deconvolution is the action that recovers the effect of the convolution i.e. the deconvolution does the opposite task of the convolution. With respect to equation (8.10), the deconvolution will restore $u[n]$ when we have $h[n]$ and $s[n]$. When we are trying to find the source excitation $u$ by having the output $s$ and the system coefficients $\alpha$, this action is also be called a deconvolution operation.

In figure 8.2 the vocal-tract filter is expressed by the AR process shown in equation (8.11). In the same figure, the LP filter is expressed by equation (8.12). In figure 8.2, the AR model denoted by $H(z)$ at the left side is modeled by

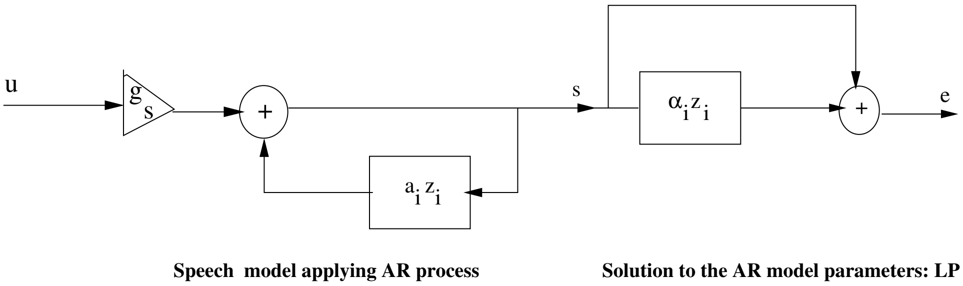

u
g
s
+
s
a
i
z
i
α
i
z
i
+
e
Speech model applying AR process
Solution to the AR model parameters: LP

Figure 8.2: Speech production system and linear prediction analysis

equation (8.11) and the LP system denoted by $H_l(z)$ at the right side is modeled by equation (8.12). The $a_i$ parameters where $i = 1, 2, \cdots, p$ in the left side of figure 8.2 are unknown and are solved using the LP system on the right side. The LP provides the solution to the unknown $a_i$ parameters where $i = 1, 2, \cdots, p$ of the vocal-tract system on the left side. The symbols used in eq (8.12) are introduced in chapter 7 in section 7.3. The relation between $H(z)$ and $H_l(z)$ are discussed in chapter 7 in section 7.4 and shown in figure 7.3 in that chapter.

$$H(z) = \frac{S(z)}{U(z)} = \frac{g_s}{1 - \sum_i^p a_i z^{-i}} \tag{8.11}$$

The solution to $a_i$ parameters is estimated by the $\alpha_i$ and incorporated in eq (8.12) where the running index $i$ in our case is same for the both systems $H(z)$ and $H_l(z)$. $i = 1, 2, \cdots, p$. The goal is to estimate $\alpha_i$ so that it is a close capture of $a_i$.

$$H_l(z) = \frac{E(z)}{S(z)} = 1 - \sum_i^p \alpha_i z^{-i} \tag{8.12}$$

The terms gain and error in the mean squared sense that come up in the LP prediction are introduced next.

#### 8.2.1.1 Gain and Errors

A physical explanation of the gain in the LP analysis comes from the volume controller which is different for each speech and for each speaker. Analytically this may come from an energy level for the frames and it may not be the same for each frame. Multiplying equation (8.11) and equation (8.12), we find $g_s$ as shown by equation (8.13). If the coefficients are correct, then we get $\alpha_i = a_i$.

$$\frac{S(z)}{U(z)}.\frac{E(z)}{S(z)} = \frac{E(z)}{U(z)} = g_s \frac{1 - \sum_i^p \alpha_i z^{-i}}{1 - \sum_i^p a_i z^{-i}} \tag{8.13}$$

Taking an inverse z-transform of $\frac{E(z)}{U(z)} = g_s$ (equation (8.13)), we find the error signal written in equation (8.14) by using equation (8.13). Equation (8.14) is the time domain representation of equation (8.13).

$$e[n] = g_s u[n] \tag{8.14}$$

Now we extend the error term to the minimum mean squared error in the next section.

## 8.3 Mean Squared Error (MSE) and its Minimization

We want to find the LP coefficients which are closest to the AR coefficients. These coefficients are the vocal-tract parameters to represent the true speech. The computational aspects of MSE in the LP case discussed next.

The mean squared error criterion emphasizes the effect of large errors much more than the absolute error criterion and MSE is more sensitive to outliers than the absolute error criterion [24].

### 8.3.1 Computational Aspects

When the Euclidean distance describes a distribution and the squared Euclidean errors are considered, then the underlying distribution of the process is presumably a Gaussian distribution. The zero mean signal can be obtained by taking the mean of the signal and then subtracting the mean from the signal.

The mean of the discrete random process is the average summation of the ensemble. Applying the concept of the ergodic mean convergence in the stochastic process, we compute the mean squared of the error of the speech signal. This is shown in equation (8.15). There we see for the time interval the difference between the time averaged mean $\mu_n$, and ensemble averaged mean $\mu$, the time averaged correlation $r_n$ and the ensemble averaged autocorrelation $r$ converges to

0.

$$\begin{aligned}\lim_{n\to\infty} E(\mu_n - \mu) = 0\\ \lim_{n\to\infty} E(r_n - r) = 0\end{aligned} \tag{8.15}$$

The mean squared error is denoted by $\eta(n)$ for each speech signal $s[n]$.

To find the minimum mean squared error, we need to differentiate $E\{e^2\}$ in equation (8.16) with respect to $\alpha_k$ for $k \in \mathcal{Z}$. The error $e$ is computed for $n$ where $n = 0, 1, 2, \cdots, N-1$. We have to differentiate equation (8.16) $p$ many times with respect to $\alpha_k$, hence there are $p$ many equations. It is formulated in equation (8.17).

The derivation is given in the following equations. There, $\eta'_k = \frac{\partial \eta}{\partial \alpha_k}$ for $n = 0, 1, \cdots, N$ and $k = 1, 2, \cdots, p$.

$$\eta = E\{e^2\} = \frac{1}{N}\sum_{n=0}^{N-1}[s[n] - \hat{s}[n]]^2 \tag{8.16}$$

$$\eta'_k = \frac{\partial[E\{e^2\}]}{\partial \alpha_k} = \frac{\partial}{\partial \alpha_k}\frac{1}{N}\sum_{n=0}^{N-1}[s^2[n] - 2s[n]\hat{s}[n] + \hat{s}^2[n]]\} \tag{8.17}$$

Equation (8.17) tells us the expected value of the summation of the squared error signal for $N$ many speech samples. We can think of using the $N$ many samples in a segment for a short time interval.

For minimizing the mean squared error, we set equation (8.17) equals to 0 and compute the partial derivative with respect to $\alpha_k$ for $k = 1, 2, \cdots, p$ times.

$$\eta'_k = \frac{\partial[E\{e^2\}]}{\partial \alpha_k} = 0 \tag{8.18}$$

From equation (8.17) and equation (8.18) we arrive at equation (8.19).

$$\begin{aligned}\frac{\partial}{\partial \alpha_k}\{\sum_{n=0}^{N-1}[s^2[n] - 2s[n]\hat{s}[n] + \hat{s}^2[n]]\} = 0\\ -2\sum_{n=0}^{N-1} s[n]\frac{\partial \hat{s}[n]}{\partial \alpha_k} + 2\sum_{n=0}^{N-1}\hat{s}[n]\frac{\partial \hat{s}[n]}{\partial \alpha_k} = 0\end{aligned} \tag{8.19}$$

By rearranging equation (8.19), we find equation (8.20).

$$\sum_{n=0}^{N-1} s[n]\frac{\partial \hat{s}[n]}{\partial \alpha_k} = \sum_{n=0}^{N-1} \hat{s}[n]\frac{\partial \hat{s}[n]}{\partial \alpha_k} \tag{8.20}$$

$$\frac{\partial \hat{s}[n]}{\partial \alpha_k} = s[n-k] \tag{8.21}$$

The derivation of $\hat{s}$ with respect to $\alpha_k$ is written in equation (8.22). $\hat{s}$ is shown in equation (8.3).

$$\sum_{n=0}^{N-1} s[n]s[n-k] = \sum_{n=0}^{N-1} \hat{s}[n]s[n-k] = \sum_{n=0}^{N-1}\sum_{i=1}^{p} \alpha_i s[n-i]s[n-k] \tag{8.22}$$

Now substituting the value of $\hat{s}$ (shown in equation (8.3) ) in equation (8.20) we arrive at equation (8.23). In equation (8.19), index $k$ is dummy variable of the differentiation of equation (8.16) with respect to $\alpha_k$.

$$\sum_{n=0}^{N-1} s[n]s[n-k] = \sum_{n=0}^{N-1}\sum_{i=1}^{p} \alpha_i s[n-i]s[n-k] \tag{8.23}$$

Equation (8.23) says $s[n]$ can be obtained by taking $i$ many past $\alpha$ coefficients and we solve the minimal average error problem by solving the above equations. There are many standard methods for this. The smaller the number of coefficients is the faster the solutions are obtained. That would indicate one should use only very few coefficients. On the other hand, however, reducing the number of coefficients is also limited because with a very small number of coefficients we cannot expect a good approximation. In the speech signal, these coefficients are real valued numbers.

Suppose $r[i]$ denotes the autocorrelation between $s[n]$ and $s[n-i]$ as shown in equation (8.24). There the autocorrelation function measures the similarity between the process $s$ at time instances $n$ and $n-i$. The autocorrelation function for a real valued wide sense stationary process is a symmetric function. Therefore, for fixed $n$, $r[i] = r[-i]$ and we can say $r[-i] = E\{s[n]s[n-i]\}$. In equation (8.24), $i = 1, 2, \cdots, p$ and $n = 0, 1, 2, \cdots, N-1$.

$$r[i] = E\{s[n]s[n-i]\} \tag{8.24}$$

In equation (8.25), $r[i, k]$ denotes the autocorrelation between $s[n-i]$ and $s[n-k]$.

In equation (8.25) by symmetry we have $r[i,k] = r[k,i]$.

$$r[i,k] = E\{s[n-i]s[n-k]\} \tag{8.25}$$

From the equations (8.23), (8.24) and (8.25), we obtain (8.26)

$$r[i] = \sum_{k=1}^{p} \alpha_k r[i,k] \tag{8.26}$$

Expanding equation (8.26), we arrive at $p$ many linear equations shown in equation (8.27).

$$\begin{aligned}
\alpha_1 r[0] + \alpha_2 r[1] + \alpha_3 r[2] + \cdots\cdots + \alpha_p r[p-1] &= r[1] \\
\alpha_1 r[1] + \alpha_2 r[0] + \alpha_3 r[1] + \cdots\cdots + \alpha_p r[p-2] &= r[2] \\
\alpha_1 r[2] + \alpha_2 r[1] + \alpha_3 r[0] + \cdots\cdots + \alpha_p r[p-3] &= r[3] \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots&\cdots\cdots \\
\alpha_1 r[p-1] + \alpha_2 r[p-2] + \alpha_3 r[p-3] + \cdots\cdots + \alpha_p r[0] &= r[p]
\end{aligned} \tag{8.27}$$

Equation (8.27) is written using a matrix vector form by equation (8.28).

$$\begin{bmatrix} r[0] & r[1] & \cdots & r[p-2] & r[p-1] \\ r[1] & r[0] & \cdots & r[p-1] & r[p-2] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r[p-1] & r[p-2] & \cdots & r[1] & r[0] \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_p \end{bmatrix} = \begin{bmatrix} r[1] \\ r[2] \\ \vdots \\ r[p] \end{bmatrix} \tag{8.28}$$

In equation (8.28), $\mathbf{\Phi}$ is a $p \times p$ dimensional matrix, $\mathbf{r}$ is $p \times 1$ dimensional vector and the coefficients $\alpha$ is $p \times 1$ dimensional vector.

$$\mathbf{\Phi} = \begin{bmatrix} r[0] & r[1] & \cdots & r[p-2] & r[p-1] \\ r[1] & r[0] & \cdots & r[p-1] & r[p-2] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r[p-1] & r[p-2] & \cdots & r[1] & r[0] \end{bmatrix} \quad \alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_p \end{bmatrix} \quad \mathbf{r} = \begin{bmatrix} r[1] \\ r[2] \\ \vdots \\ r[p] \end{bmatrix}$$

In equation (8.28), $\mathbf{\Phi}$ is symmetric and Toeplitz. Equation (8.28) is called the Yule-Walker equation.

$$\mathbf{\Phi}\alpha = \mathbf{r} \tag{8.29}$$

A direct solution of the equation (8.29) is the solution for the $\alpha_i$ coefficients for $i = 1, 2, \cdots, p$. The solution to (8.29) is obtained multiplying both sides of equation (8.29) by the inverse of $\mathbf{\Phi}$ matrix shown in equation (8.30).

$$\alpha = \mathbf{\Phi}^{-1}\mathbf{r} \tag{8.30}$$

At lag 0, equation (8.31) $r$ is the mean squared value of the $s[n]$. This is actually the energy of the signal.

$$r[0] = E\{s[n]s[n]\} = E\{s[n]^2\} \tag{8.31}$$

How many coefficients are enough in equation (8.29) is discussed in section 8.1.1.

For the sampling frequency is 16 kHz, the model order is 12. The solutions to the $\alpha_i$ in equation (8.29) is obtained using the Levinson-Durbin recursion approach [80]. The AR parameters can be approximated by using different types of LP approaches. Some of them including our LP approach for the AR parameters approximation are discussed in the next chapter.

# Chapter 9

# LPC Solution Approaches

**Outline of the chapter** This chapter is a continuation of the investigations in the last one. The linear prediction (LP) can be approached using auto-correlation, auto-covariance, Burg and unconstrained least squares (ULS) approaches. These approaches are briefly discussed in this chapter. We have applied the ULS approach for parametric signal modeling.

First we list some LP parametric approaches that have partially been introduced in the last chapter.

- Autocorrelation Approach: This approach uses the Yule-Walker equation to extract the parameters. The Levinson-Durbin algorithm is used for the parametric solution.

- Covariance Approach: The covariance approach uses the Cholesky decomposition for its parametric solution.

- Burg Approach: The Burg method is an order and time recursive approach. This also uses the Levinson Durbin algorithm.

- ULS Approach: The unconstrained parametric solution is an order and time recursive approach. Both order and time are used to extract and update the parameters. This does not use the Levinson-Durbin algorithm to extract the parameters.

Next we introduce the above listed approaches.

## 9.1 Autocorrelation Approach

In autocorrelation approach, the number $N$ of non-zero samples of a certain length $l$ is nonzero and zero outside of the length $l$. The averaged autocorrelation function is replaced by the time averaged autocorrelation function.

This method is the most straight forward one for the AR model parameters. In this approach, the ensemble autocorrelation $r[i]$ is replaced by the corresponding time-averaged autocorrelation computed from a given block of data. In the previous chapter 8 we have discussed errors and the minimization of the mean squared error and the use of a windowed signal. This is an important use of auto-correlation that we will not repeat here.

In chapter 8, we have explained how we have $p$ many linear equations by equation (8.21). The LP solution using this approach needs the inversion of the $\mathbf{\Phi}$ matrix and the multiplication of a $p \times p$ matrix with a $p \times 1$ length $\mathbf{r}$ vector. Here $\mathbf{\Phi}$ is a Toeplitz matrix and in this matrix, the diagonal entries are identical. $\mathbf{\Phi}$ is also a symmetric matrix and thus we have $r(i,k) = r(k,i)$. The solution for the parameters can be obtained by using a Gaussian elimination approach or by the Levinson-Durbin recursion. This algorithm is discussed in [80]. The Levinson approach is efficient to solve the parameters and it uses the properties of Toeplitz matrix.

In figure 9.1, fig a is a single speech frame in the time domain, fig b shows the pole-zero plot using LP analysis which shows the stability of the model because its poles are inside the unit circle. fig c is the log based FFT spectrum of the LP coefficients. fig d is the spectrum of the residual signal obtained by LP analysis and the input that is the random white Gaussian noise and is again obtained by the deconvolution i.e. the spectral analysis of the deconvolution of the LP parameters. fig e is the spectral analysis of the residual signal and fig f is the verification of the excitation of the signal. The implementation in figure 9.1 has a close replication that is given in [127] but we modify the implementation using our own data for our own experiments.

The advantage of the autocorrelation approach is that it ensures the stability of the system model. The Levinson-Durbin recursion makes the computations efficient. We have introduced the computations of the LPC autocorrelation problem solving approach using Levinson-Durbin recursion algorithm in chapter 8 but a detailed computational aspect of this algorithm is not discussed in the thesis. The disadvantage of the autocorrelation approach is that it uses windowing in the segmentation process and therefore the true spectrum might not be obtained

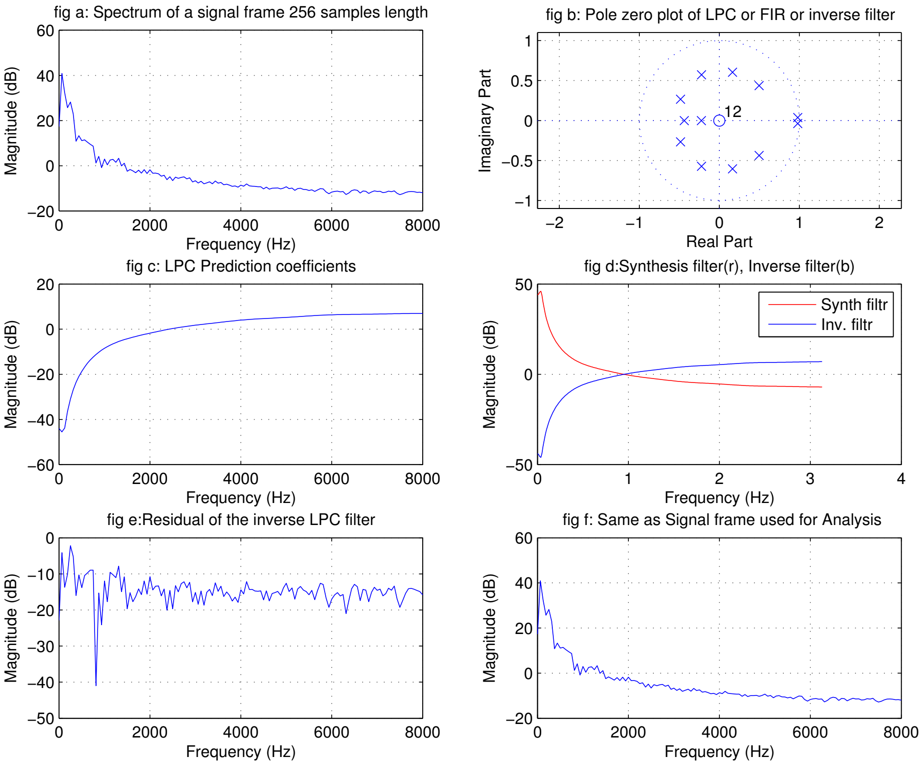


Figure 9.1: LP by autocorrelation using Yule-Walker approach: Öffne die Tür

in the spectral analysis.

In figure 9.2, we see the analysis of the signal model using auto-correlation approach. In a, we see the segment of the speech signal. In b, we see the frequency response of the LP filter which parameters extracted by auto-correlation approach or Yule-Walker equation. In c, we wee the pole position of the covariance approach and in d, we see the excitation which is the output of the filter and this is white noise. Here the peak values in figure b indicate the coefficients of the AR parameter approximation. These are normally called the formant of the speech signal. The order of the model is 12. Therefore, in figure b, we see six peaks. These peaks are smaller as frequency increases. This is because the speech signal is a low frequency signal.

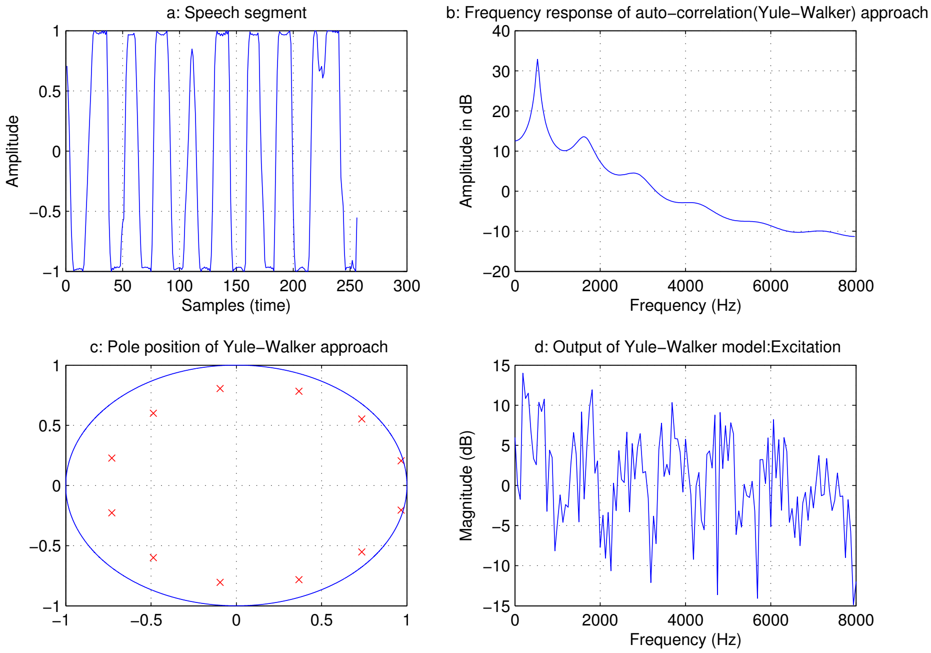


Figure 9.2: Signal model: Auto-correlation (Yule-Walker) approach

## 9.2 Covariance Approach

In the covariance approach, the minimum mean squared error is computed using the the derivative of equation (9.1) with respect to $a_k$ for $k \in \mathcal{Z}$ and $k =$

$1, 2, \cdots, p$. The starting point is mean squared error.

$$\eta = \{E(e^2)\} = \frac{1}{N} \sum_{n=p}^{p+N-1} \left(s[n] - \sum_{i=1}^{p} a_i s(n-i)\right)^2 \tag{9.1}$$

The summation limit can be any point starting from $n = p$ to $n = N$. Here the truncation of the signal is not essential and an explicit signal windowing is not done. Therefore, in this approach the spectral distortions from the rectnular windowed signal do not occur.

Instead of when using a correlation approach applying the Yule-Walker is to obtain the auto-correlation matrix $\mathbf{\Phi}$ described in section 8.3.1 in equation (8.30). Here in the covariance approach, the covariance matrix $\mathbf{C}$ in equation (9.5) derived from equation (9.3) is positive definite and symmetric but it is not Toeplitz. In equation (9.2), $n = 0, 1, \cdots, N-1$ and $i, j = 1, 2, \cdots, p$

$$c_n[i, 0] = \sum_{j=1}^{p} \alpha_i c_n[i, j] \tag{9.2}$$

An extended equation (9.2) for $c_n[i, k]$ is shown in equation (9.3). In the equation we take $i, k = 1, 2, \cdots, p$.

$$c_n[i, k] = \sum_{n=l}^{l+N-1} s[n-i]s[n-k] \quad ; \forall\, l \in \mathcal{Z} \tag{9.3}$$

$c_l[i, k]$ in equation (9.3) reflects $p$ many linear equations that can be written in a matrix vector form similar to equation (8.27) as it is shown in equation (9.5)

$$\begin{bmatrix} c[1,1] & c[1,2] & \cdots & c[1,p-1] & c[1,p] \\ c[2,1] & c[2,2] & \cdots & c[2,p-1] & c[2,p] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ c[p,1] & c[p,2] & \cdots & c[p,p-1] & c[p,p] \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_p \end{bmatrix} = \begin{bmatrix} c[1,0] \\ c[2,0] \\ \vdots \\ c[p,0] \end{bmatrix} \tag{9.4}$$

Equation (9.4) has $p$ many linear equations in a matrix form denoted by $\mathbf{C}$. The LP coefficient vector in the covariance method is denoted by $\alpha$ and the covariance vector is denoted by $\mathbf{c}$. These notations are used in equation (9.5). $\mathbf{C}$ in equation (9.5) is symmetric and positive definite but not Toeplitz as it is in the auto-correlation approach discussed in the previous chapter 7.

$$\mathbf{C} = \begin{bmatrix} c[1,1] & c1,2] & \cdots & c[1,p-1] & c[1,p] \\ c[2,1] & c[2,2] & \cdots & c[2,p-1] & c[2,p] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ c[p,1] & c[p,2] & \cdots & c[p,p-1] & c[p,p] \end{bmatrix} ; \quad \alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_p \end{bmatrix} ; \quad \mathbf{c} = \begin{bmatrix} c[1,0] \\ c[2,0] \\ \vdots \\ c[p,0] \end{bmatrix} \tag{9.5}$$

Equation (9.4) in a compact form is now written in equation (9.6).

$$\mathbf{c} = \mathbf{C}\alpha \tag{9.6}$$

The solution to the LP parameters is then rewritten using equation (9.6) in equation (9.7). Finding the solution to the LP parameters $\alpha$ in equation(9.7) requires computing the inverse matrix i.e. $\mathbf{C}^{-1}$. The covariance matrix $\mathbf{C}$ is symmetric, positive definite but not Toeplitz, therefore Levinson Durbin recursion is not used, instead the Cholesky decomposition is used in equation (9.7) for the parametric solution. In Cholesky decomposition, the covariance matrix denoted by $\mathbf{C}$ is divided into lower and upper triangular matrix [24]. We have not shown the derivation of the Cholesky decomposition approach in the text.

$$\alpha = \mathbf{C}^{-1}\mathbf{c} \tag{9.7}$$

The covariance approach works on the the whole data set or on the segments of the signal. The autocorrelation approach is applied on the finite length signal and any signal beyond the finite length is zero. The covariance approach is not quite practical for implementing the model using real time data because this works on the whole signal or on the blocks and the samples are not zero outside the processed block while the speech signal is generally processed as a finite length signal.

Similar to figure 9.2, in figure 9.3, we see the analysis of the signal model using covariance approach. In a, we see the segment of the speech signal. In b, we see the frequency response of the LP filter which parameters extracted by covariance approach. In c, we wee the pole position of the covariance approach and in d, we see the excitation which is the output of the filter and this is white noise.

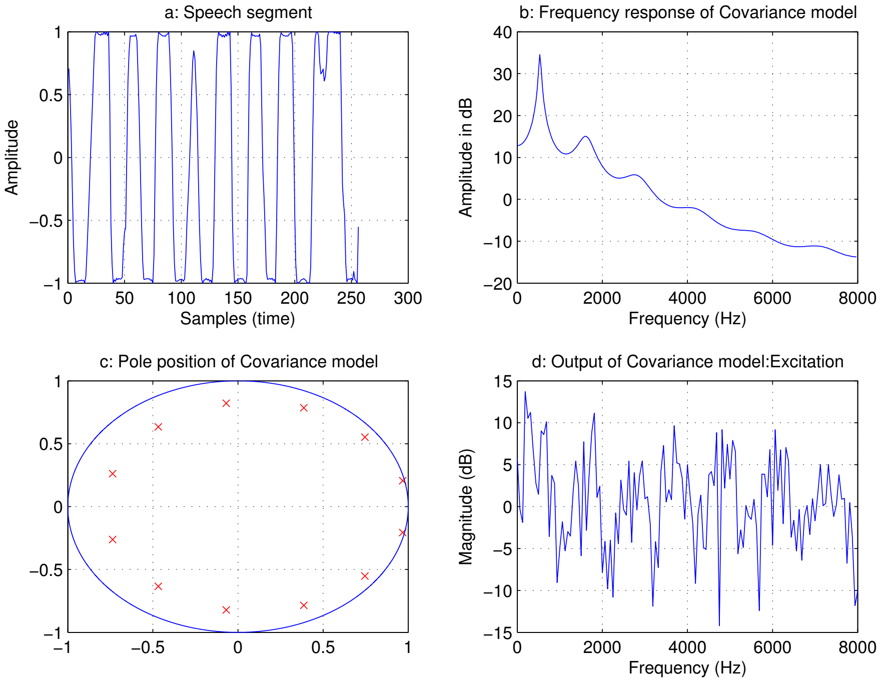


Figure 9.3: Signal model: Covariance approach

## 9.3 The Burg Approach

The Burg approach is an order recursive least-squares linear predictor. There order recursive means that if the model is of $p$ th order, we can compute the model parameters of the model order $p+1$. The autocorrelation and covariance approach are fixed order algorithms meaning that they are not order recursive. This says if we change the order of the model, we need to repeat the whole computations. The Burg approach uses both the forward and backward error minimization approach. We have introduced the forward prediction and its error, now we will introduce the backward prediction (BP) and its error. The order recursive algorithm interconnects the optimum filtering and the FLP and the BLP problems. The optimum filtering refers to the system which response is closest to the desired response. The Burg approach needs to consider the time instance $n$ and the order $p$ such that $i = 1, 2, \cdots, p$.

Some terminologies of the Burg approach as the excitation, the speech input and the forward and backward prediction errors are shown in figure 9.4. In figure 9.4, we observe how the forward and backward predicted value are estimated from the same observation using the same amount of samples. We name now the forward prediction error $e[n]$ as $e^f[n]$ for an easier manipulation and it is written in equation (9.8) where $f$ denotes forward.

$$e^f[n] = s[n] - \hat{s}[n] = s[n] - \sum_{i=1}^{p} \alpha_i s[n-i] \tag{9.8}$$

**Backward Prediction** In the backward prediction shown in equation (9.9) the current sample is computed from $p$ many future samples. The equation is expanded in equation (9.10). $\beta_i$ is the backward prediction coefficient for $i = 1, 2, \cdots, p$.

$$\hat{s}[n-p] = \sum_{i=1}^{p} \beta_i s[n-i+1] \tag{9.9}$$

$$\hat{s}[n-p] = \beta_1 s[n-p+1] + \beta_2 s[n-p+2] + \beta_3 s[n-p+3] + \cdots + \beta_p s[n] = \sum_{i=1}^{p} \beta_i s[n-i+1] \tag{9.10}$$

Now if we write the signal $s[n-p]$ using backward prediction we arrive at equation (9.11). There $e^b$ is backward prediction error where $b$ denotes backward.

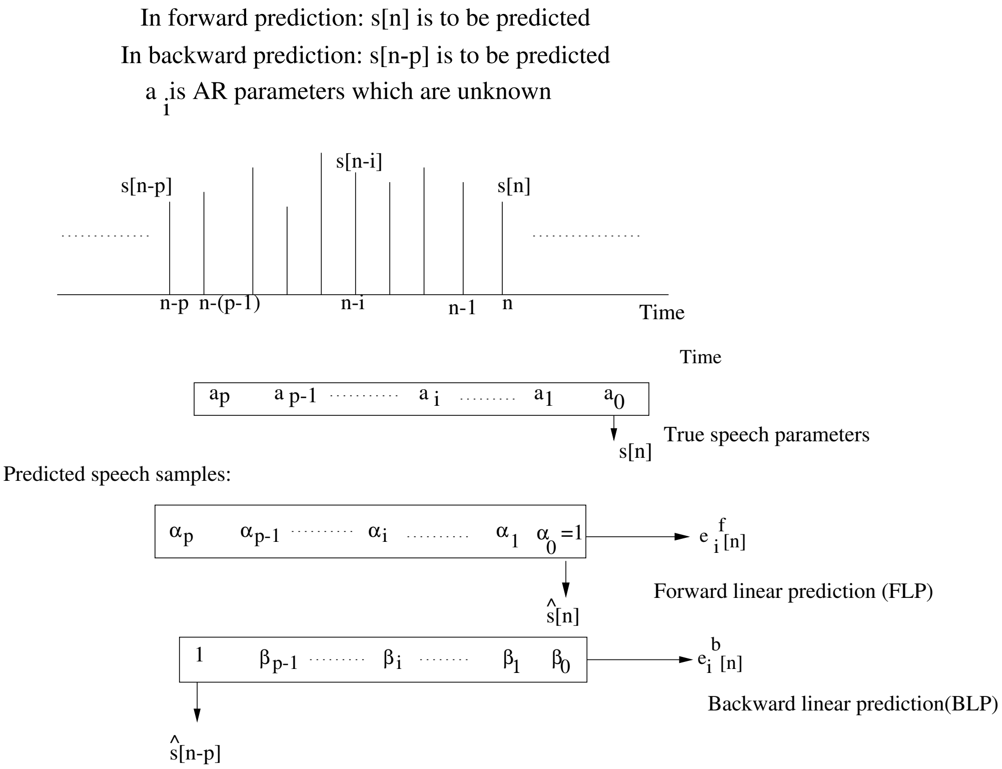

In forward prediction: s[n] is to be predicted
In backward prediction: s[n-p] is to be predicted
a i is AR parameters which are unknown
s[n-p]
s[n-i]
s[n]
n-p
n-(p-1)
n-i
n-1
n
Time
Time
ap
a p-1
a i
a1
a0
s[n]
True speech parameters
Predicted speech samples:
αp
αp-1
αi
α1
α0 =1
e i f [n]
ŝ[n]
Forward linear prediction (FLP)
1
βp-1
βi
β1
β0
ei b [n]
Backward linear prediction(BLP)
ŝ[n-p]

Figure 9.4: Visualization of the forward and backward linear prediction [24]

$$\hat{s}[n-p]] = \sum_{i=1}^{p} \beta_i s[n-i+1] + e^b[n] \tag{9.11}$$

Now the backward prediction error in equation (9.12) is the difference between the current sample and the current backward predicted sample. In equation (9.12), $'$ denotes transpose. We have used $T$ as a time frame in chapter 13 even though $T$ is a conventional transpose notation. For fixed order, we simply write $e^b[n]$.

$$e^b[n] = s[n-p] - \hat{s}[n-p] = s[n-p] - \beta^{'}\mathbf{s}[n-p] \tag{9.12}$$

The backward prediction gives rise to the following linear equations in equation (9.13). By rearranging equation (9.13) we arrive at equation (9.14). Here $r$ is again autocorrelation function.

$$\begin{aligned}
\beta_0 r[p-1] + \beta_1 r[p-2] + \beta_2 r[p-3] + \cdots\cdots + \beta_{p-1} r[0] &= r[p] \\
\beta_0 r[p-2] + \beta_1 r[p] + \beta_2 r[p-4] + \cdots\cdots + \beta_{p-1} r[1] &= r[p-1] \\
\beta_0 r[p-3] + \beta_1 r[p-4] + \beta_2 r[p-5] + \cdots\cdots + \beta_{p-1} r[2] &= r[p-2] \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots&\cdots \\
\beta_0 r[0] + \beta_1 r[1] + \beta_2 r[2] + \cdots\cdots + \beta_{p-1} r[p-1] &= r[1]
\end{aligned} \tag{9.13}$$

$$\begin{aligned}
\beta_{p-1} r[0] + \beta_{p-2} r[1] + \beta_{p-3} r[2] + \cdots\cdots + \beta_0 r[p-1] &= r[p] \\
\beta_{p-1} r[1] + \beta_{p-2} r[0] + \beta_{p-3} r[1] + \cdots\cdots + \beta_0 r[p-2] &= r[p-1] \\
\beta_{p-1} r[2] + \beta_{p-2} r[1] + \beta_{p-3} r[0] + \cdots\cdots + \beta_0 r[p-3] &= r[p-2] \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots&\cdots \\
\beta_{p-1} r[p-1] + \beta_{p-2} r[p-2] + \beta_{p-3} r[p-3] + \cdots\cdots + \beta_0 r[0] &= r[1]
\end{aligned} \tag{9.14}$$

Now we can write the backward prediction by equation (9.15) similar to the forward prediction equation (8.27) derived in chapter 8.

$$\begin{bmatrix} r[0] & r[1] & \cdots & r[p-2] & r[p-1] \\ r[1] & r[0] & \cdots & r[p-1] & r[p-2] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r[p-1] & r[p-2] & \cdots & r[1] & r[0] \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix} = \begin{bmatrix} r[p] \\ r[p-1] \\ \vdots \\ r[1] \end{bmatrix} \tag{9.15}$$

Now if we compare equation (9.15) with equation (8.27), we obtain the matrix $\mathbf{\Phi}$, the backward prediction coefficients $\beta$ and the vector $\mathbf{r}^b$ in a matrix and vector form.

$$\mathbf{\Phi} = \begin{bmatrix} r[0] & r[1] & \cdots & r[p-2] & r[p-1] \\ r[1] & r[0] & \cdots & r[p-1] & r[p-2] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r[p-1] & r[p-2] & \cdots & r[1] & r[0] \end{bmatrix} \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix} \mathbf{r}^b = \begin{bmatrix} r[p] \\ r[p-1] \\ \vdots \\ r[1] \end{bmatrix}$$

Since the matrix $\mathbf{\Phi}$ is Toeplitz and symmetric in equation (8.27), we can rearrange this equation (8.27). Now if we compare equation (9.15) with equation (8.27), and rewrite equation (8.27) by equation (9.16), we find the relation between backward prediction coefficients $\beta$ and forward prediction coefficients $\alpha$.

$$\begin{bmatrix} r[0] & r[1] & \cdots & r[p-2] & r[p-1] \\ r[1] & r[0] & \cdots & r[p-1] & r[p-2] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r[p-1] & r[p-2] & \cdots & r[1] & r[0] \end{bmatrix} \begin{bmatrix} \alpha_p \\ \alpha_{p-1} \\ \vdots \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} r[p] \\ r[p-1] \\ \vdots \\ r[1] \end{bmatrix} \tag{9.16}$$

From equation (9.15), equation (9.16) and equation (8.21) given in chapter 8, we interconnect the BLP coefficients. $\beta$ is the reverse version of FLP coefficients $\alpha$ which is also shown in equation (9.17).

$$\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix} = \begin{bmatrix} \alpha_p \\ \alpha_{p-1} \\ \vdots \\ \alpha_1 \end{bmatrix} = \alpha^b \tag{9.17}$$

In equation (9.10), $\mathbf{r}[i] = E[s[n]s[n-i]]$ for $i = 1, 2, \cdots, p$ and $n = 0, 1, \cdots, N-1$

Thus the solution to the BLP is written in equation (9.18).

$$\begin{aligned} \beta &= \mathbf{\Phi}^{-1}\mathbf{r} \\ \alpha^b &= \mathbf{\Phi}^{-1}\mathbf{r} \end{aligned} \tag{9.18}$$

In the Burg level approach, the prediction coefficients are achieved by minimizing the average of the forward and backward errors shown in equation (9.19). $e^f$ and $e^b$ are defined in equation (9.8) and equation (9.12). Here we used the forward and backward quadratic errors.

$$e_p^{fb} = \frac{1}{N}\sum_{n=0}^{N-1}\sum_{i=1}^{p}[\{e^f[n,i]\}^2 + \{e^b[n,i]\}^2] \tag{9.19}$$

As mentioned, the quality of the error is depending on the selection of the coefficients $\alpha$. These are obtained by equation (9.20) Levinonson-Durbin recursion approach [80]. In equation (9.20), $\kappa$ is the reflection coefficients. One way to find this is lattice filter realization. Thus the error interpretation can be done using lattice filtering. Next we have briefly discussed this.

$$\alpha_i[n] = \alpha_{i-1}[n] + \kappa_i\alpha_{i-1}[i-n] \tag{9.20}$$

**Benefits of FLP and BLP Error Computations:** The BLP and FPL convey the same statistical information of the signal but a combination of both BLP error and FLP error generate more error points. The results in improved estimate of the AR parameters [121]. Thus $(N-p)$ forward and $(N-p)$ backward LP errors may summarized as :

$$\mathbf{e}_p^{fb} = \begin{bmatrix}\mathbf{e}_p^f \\ \mathbf{e}_p^b\end{bmatrix} \mathbf{e}_p^{fb} = \begin{bmatrix}\mathbf{S}_p \\ \mathbf{S}_p^{'}\mathbf{J}\end{bmatrix} \mathbf{e}_p^{fb} = \begin{bmatrix}1 \\ \alpha_p^{fb}\end{bmatrix} \tag{9.21}$$

$$\mathbf{S}_p = \begin{bmatrix} s[p+1] & s[p] & \cdots & s[1] \\ s[p+2] & s[p+1] & \cdots & s[2] \\ \vdots & \vdots & \vdots & \vdots \\ s[N-1] & s[N-2] & \cdots & s[N-p+1] \\ s[N] & s[N-1] & \cdots & s[N-p] \end{bmatrix}; \quad \mathbf{s} = \begin{bmatrix} s[p+1] \\ s[p+2] \\ \vdots \\ s[N-1] \\ s[N] \end{bmatrix} \tag{9.22}$$

$$\mathbf{J} = \begin{bmatrix} 0 & 0 & \dots & \cdots & 1 \\ 0 & 0 & 0 & \dots & 0 \\ . & . & . & . & . \\ 0 & 1 & 0 & \dots & 0 \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix} \tag{9.23}$$

Similarly,

$$\mathbf{S}_p^{'}\mathbf{J} = \begin{bmatrix} s[1] & s[2] & \cdots & s[p+1] \\ s[2] & s[1] & \cdots & s[p] \\ \vdots & \vdots & \vdots & \vdots \\ s[N-p+1] & s[N-p+2] & \cdots & s[N-p] \\ s[N-p] & s[N-p+1] & \cdots & s[N] \end{bmatrix}; \quad \mathbf{s}^{'} = \begin{bmatrix} s[N] \\ s[N-1] \\ \vdots \\ s[p+2] \\ s[p+1] \end{bmatrix} \tag{9.24}$$

$$\mathbf{e}_p^f = \begin{bmatrix} e_p^f[p+1] \\ e_p^f[p+2] \\ \vdots \\ e_p^f N-1] \\ e_p^f[N] \end{bmatrix}; \mathbf{e}_p^b = \begin{bmatrix} e_p^b[p+1] \\ e_p^b[p+2] \\ \vdots \\ e_p^b N-1] \\ e_p^b[N] \end{bmatrix}; \alpha_p^{fb} = \begin{bmatrix} \alpha_p[1] \\ \alpha_p[2] \\ \vdots \\ \alpha_p[p-1] \\ \alpha_p[p] \end{bmatrix} \tag{9.25}$$

### 9.3.1 Lattice FIR Filter

The lattice structure is useful in modeling the layer or the cross section of the tube. Each section or the stage of the lattice filter indicates the cross sectional area of the vocal tract tube. The lattice predictor combines the forward prediction (FP) error and backward prediction (BP) error in a single cascaded structure. This gives the lattice prediction coefficient. Figure 9.5 is a $p^{th}$ order lattice structured filter. Each rectangular box in the figure is embedded with the backward and forward error formulation and a computation of the reflection coefficients which are known as PARCOR coefficients. Changing the filter length leads to a completely new set of filter coefficients. The order of the predictions and the stages of the lattice predictor are the same. If the prediction order is $p$, then the lattice structure has $p$ many stages. The the prediction coefficients $\alpha$ can be directly computed from the lattice filtering. Reflection coefficients and its relation to the vocal tract model are introduced in chapter 2 in section 2.4.2. We show here how the reflection coefficients can be computed from the LP coefficients. A detailed description of the lattice filter and its structure can be found in [60], [24].

### 9.3.2 Reflection Coefficients and Linear Prediction Coefficients

Here we show how reflection coefficients $\kappa$ can be derived from the LP coefficients.

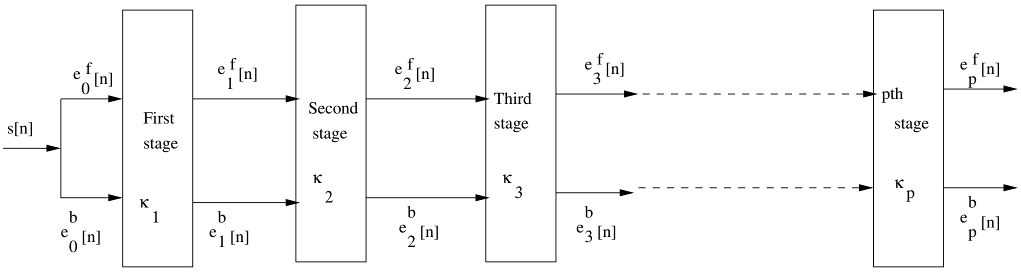


Figure 9.5: $p^{th}$ stage lattice filter

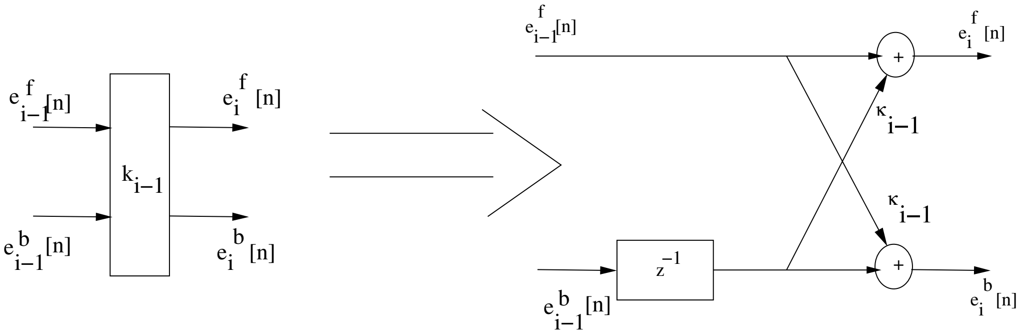


Figure 9.6: $i^{th}$ section of the lattice section in details

For the order and time recursive case, we consider the model order and the time thus we have seen $\hat{s}[n] = \sum_{i=1}^{i} \alpha_p[i]s[n-i]$ for $i = 1, 2, \cdots, p$. Now we look at figure 9.6, we say equation (9.26) holds for the input and output relation as input is denoted as $s$ and output denoted as $y$.

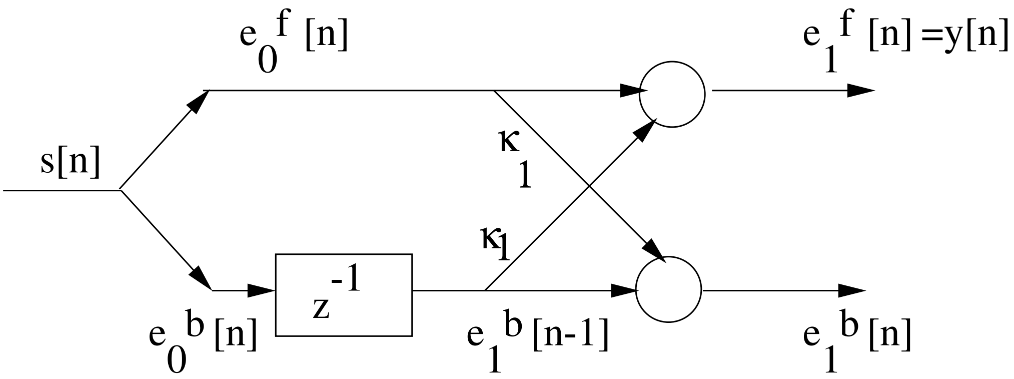


Figure 9.7: Ist order lattice structure

$$y[n] = s[n] + \alpha_1[1]s[n-1] \tag{9.26}$$

We see figure 9.7, we rewrite equation (9.26) as equation (9.27).

$$e_0^f[n] = e_0^b[n] = s[n]$$
$$e_1^f[n] = e_0^f[n] + \kappa_1 e_0^b[n-1] = s[n] + \kappa_1 s[n-1]$$

$$e_1^b[n] = \kappa_1 e_0^f[n] + e_0^b[n-1] = \kappa_1 s[n] + s[n-1] \tag{9.27}$$

Now equation (9.26) and equation (9.27) allow us to say the first order reflection coefficient $\kappa_1$ is $\alpha_1[1]$ and this is shown in equation (9.28).

$$\kappa_1 = \alpha_1[1] \tag{9.28}$$

Similarly for the order $p = 2$, we have

$$y[n] = s[n] + \alpha_2[1]s[n-1] + \alpha_2[2]s[n-2] = s[n] + \sum_{i=1}^{2} \alpha_2[i]s[n-i] \tag{9.29}$$

$$y[n] = s[n] + \alpha_2[1]s[n-1] + \alpha_2[2]s[n-2] \tag{9.30}$$

$$e_2^f[n] = e_1^f[n] + \kappa_2 e_1^b[n-1]$$
$$e_2^b[n] = e_1^f[n]\kappa_2 + e_1^b[n-1]$$

Now we arrive at equation (9.31)

$$e_2^f[n] = s[n] + \kappa_1(1+\kappa_2)s[n-1] + \kappa_2 s[n-2] \tag{9.31}$$

The first stage lattice filter is just discussed in equation (9.28), similarly the second order reflection coefficients can be obtained lattice filter gives equations (9.33) and (9.31).

$$\alpha_2[2] = \kappa_2 \quad \text{and } \alpha_2[1] = \kappa_1(1+\kappa_2) \tag{9.32}$$

$$\kappa_2 = \alpha_2[2]$$
$$\kappa_1 = \frac{\alpha_2[1]}{1+\alpha_2[2]}$$

Similar to the reflection coefficients $\kappa_1$ and $\kappa_2$, $\kappa_m$ can be computed for the $p^{th}$ ordered lattice structured filter shown in figure 9.5. The output of the $(p-1)^{th}$ stage corresponds to output of $(p-1)^{th}$ order lattice filter. Thus if $y[n]$ is the output then $y[n] = e_{p-1}^f[n]$

$$e_i^f[n] = e_{i-1}^f[n] - \kappa_i e_{i-1}^b[n-1] \tag{9.33}$$
$$e_i^b[n] = e_{i-1}^b[n-1] - \kappa_i e_{i-1}^f[n] \tag{9.34}$$

Now from equation (9.34), figure 9.5 and figure 9.6, we can define the reflection coefficient $\kappa_i$ by equation (9.35). This gives the reflection coefficients for the lattice filter.

$$\kappa_i = \frac{\sum_{n=0}^{N-1}\{e_{i-1}^b[n-1]e_{i-1}^f[n]\}}{\sum_{n=0}^{N-1}\{e_{i-1}^f[n]\}} \tag{9.35}$$

Applying equation (9.35), we get the error of the Burg approach in terms of forward and backward error prediction in equation (9.36).

$$e^{fb} = \frac{1}{N}\sum_{n=0}^{N-1}\sum_{i=1}^{p}[(e_{i-1}^f[n] - \kappa_i e_{i-1}^b[n-1])^2 + (e_{i-1}^b[n-1] - \kappa_i e_{i-1}^f[n])^2] \tag{9.36}$$

Finally minimization of equation (9.36) gives us the optimum coefficients that is the reflection coefficients in equation (9.37).

$$\kappa_i = \frac{2\sum_{n=0}^{N-1}\{e_{i-1}^{b}[n-1]e_{i-1}^{f}[n]\}}{\sum_{n=0}^{N-1} {e_{i-1}^{f}}^{2}[n] + {e_{i-1}^{b}}^{2}[n]]} \tag{9.37}$$

Now substituting $\kappa$ in equation (9.20), we obtain the coefficients of signal model using the Burg approach.

In figure 9.8, we see the analysis of the signal model using Burg approach. In a, we see the segment of the speech signal. In b, we see the frequency response of the LP filter. In c, we see the pole position of the Burg approach which says the system is stable. In d, we see the output of the filter and this is white noise.

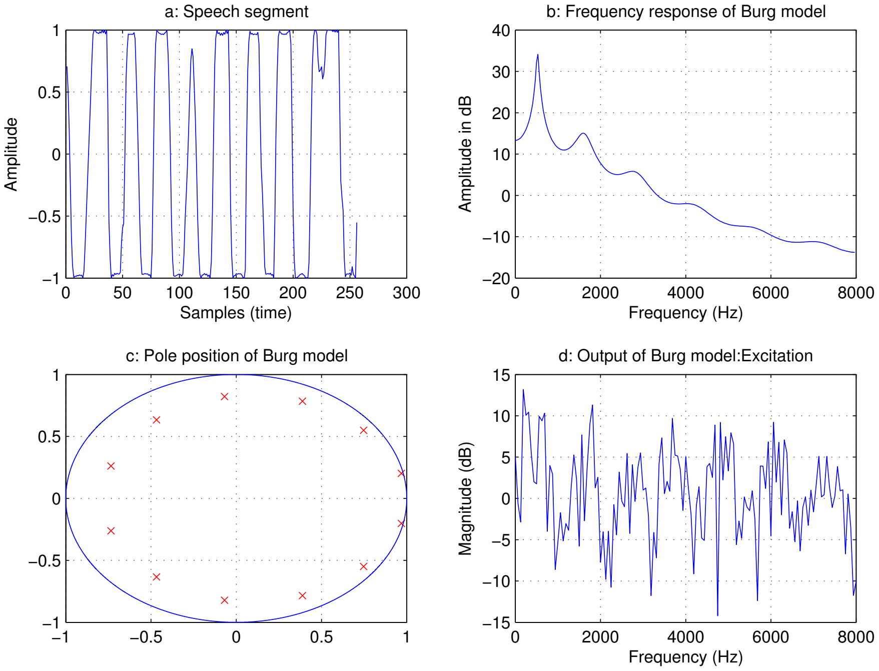


Figure 9.8: Signal model: Burg approach

## 9.4 ULS Approach

In this approach the error in equation (9.42) is minimized by computing the average of the sum of the squares of the estimated forward and backward linear prediction errors.The forward and backward prediction errors are computed in order to compute their combined error. In equation (9.17), we have seen the backward prediction coefficients are the reverse version of the forward prediction coefficients. The ULS approach is described in details [122], [121]. The ULS approach is modified covariance method. This is based on optimization with respect to all the prediction coefficients, whereas the Burg method performs a constrained least squares minimization with respect to only a single prediction coefficient.

**J** introduced in equation (9.23). This represents $p+1$ by $p+1$ dimensional reflection matrix and $'$ denotes transpose. Using the reflection matrix, we get a relation between forward linear prediction and backward linear prediction. This is shown in equation (9.38).

$$[\beta_1^p, \beta_2^p, \cdots, \beta_p^p]' = [\alpha_p^p, \alpha_{p-1}^p, \cdots, \alpha_1^p]' = \mathbf{J}\alpha \tag{9.38}$$

We get the forward prediction error $\mathbf{e}_p^f[n]$ and backward prediction error $\mathbf{e}_p^b[n]$ in equations (9.39) and (9.40). The total $N-p$ forward linear prediction error elements and the $N-p$ backward linear prediction error elements can be formed from $N$ data samples without searching through all the available data.

$$\text{FLP error:} \quad \mathbf{e}_p^f[n] = \mathbf{s}[n] - \hat{\mathbf{s}}[n] = \mathbf{s}_p^{'}[n]\alpha_p^{fb} \tag{9.39}$$

$$\text{BLP error:} \quad \mathbf{e}_p^b[n] = s[n-p] - \hat{s}[n-p] = \mathbf{s}_p^{'}[n]\mathbf{J}\alpha_p^{fb} \tag{9.40}$$

The vector notations of $\mathbf{s}_p$ and $\alpha_p^{fb}$ are formulated in equation(9.41).

$$\mathbf{s}_p[n] = \begin{bmatrix} s[n] \\ s[n-1] \\ \vdots \\ s[n-p+1] \\ s[n-p] \end{bmatrix}' ; \alpha_p^{fb} = \begin{bmatrix} \alpha_p[1] \\ \alpha_p[2] \\ \vdots \\ \alpha_p[p-1] \\ \alpha_p[p] \end{bmatrix}' ; \mathbf{J}\alpha_p = \begin{bmatrix} \alpha_p^p \\ \alpha_{p-1}^p \\ \vdots \\ \alpha_1^p \\ 1 \end{bmatrix} \tag{9.41}$$

The sum of the forward and backward linear prediction squared error $\eta^{fb}$ is written in equation (9.42). This generates $p+1$ set of linear equations shown in equation (9.42).

$$\eta_p^{fb} = \frac{1}{N}\sum_{n=p}^{N-1}\{[|\mathbf{e}_p^f[n]|^2 + |\mathbf{e}_p^b[n]|^2]\} \tag{9.42}$$

Substituting the values of the $\mathbf{e}^f[n]$ and $\mathbf{e}^b[n]$, we arrive at equation (9.43).

$$\eta_p^{fb} = \sum_{n=0}^{N-1}[(s[n] - \sum_{i=1}^{p}\alpha_p[i]s[n-i])^2 + (s[n-p] - \beta_p^{'}\mathbf{s}[n-p])^2] \tag{9.43}$$

The error minimization $\eta_p^{fb}$ with respect to the prediction coefficients yields a set of linear equations which can be formulated by equation (9.44) where $j = 1, 2, 3, \cdots, p$.

$$\sum_{i=1}^{p}\alpha_p[i]r[j,i] = r[j,0] \tag{9.44}$$

In equation (9.44), $r[i,j]$ is computed by equation (9.45).

$$r[i,j] = \sum_{n=p}^{N-1}\{\{s[n-i]s[n-j]\} + \{s[n-p+i]s[n-p+j]\}\} \tag{9.45}$$

Similar to equation (8.25) we have coefficient matrix $\mathbf{\Phi}$ obtained from $r[i,j]$ which is computed by equation (9.45). Thus we get matrix of the data vector and compute the inverse of the matrix using fast modified QR factorization. This is called an unconstrained model because the matrix $\mathbf{\Phi}$ of the data vector is not Toeplitz and the inverse can not be solved by the Levinson-Durbin approach what is the case in other standard signal models.

Equation (9.45) generates a set of $(p+1)$ times $(p+1)$ linear equations in a matrix $\mathbf{\Phi}$ similar to equation (8.28) discussed in chapter 8.

$$\mathbf{\Phi}\alpha = \begin{bmatrix} \eta_p^{fb} \\ \mathbf{0} \end{bmatrix} \tag{9.46}$$

$$\mathbf{\Phi} = \sum_{n=p+1}^{N}(\mathbf{s}_p[n]\mathbf{s}_p^{'}[n] + \mathbf{J}\mathbf{s}_p[n]\mathbf{s}_p^{'}[n]\mathbf{J}) \tag{9.47}$$

where

$$\mathbf{s}_p[n] = \begin{bmatrix} s[n] \\ s[n-1] \\ \vdots \\ s[n-p+1] \\ s[n-p] \end{bmatrix} \mathbf{s}_p^{'}[n] = \begin{bmatrix} s[n-p] \\ s[n-p+1] \\ \vdots \\ s[n-1] \\ s[n] \end{bmatrix}$$

We see the analysis of the signal model in figure 9.9 using ULS approach using the same signal segment used in figures 9.2, 9.3, and 9.8. Similar to these figures, in figure 9.9, we see the segment of the speech signal in a. In the same figure, b is the frequency response of the LP filter, figure c shows the pole position of the ULS approach and the output of the filter is white noise in figure d. Here the peak values in figure b are the coefficients of the AR parameter approximation and this are better shown in the figure than figures 9.2, 9.3, and 9.8. In figure b, we can clearly see the six peaks as representations of formants following the model order 12 as each two poles represents each formant. Thus using the ULS approach, we have six formants for 12 order LP model in b in figure 9.9.

In equation (9.46), $\mathbf{0}$ is a $p \times 1$ length zero vector and $\eta_p^{fb}$ is a $p \times 1$ length vector.

The problem to the coefficients is then solved by fast covariance QR factorization. For this we followed a reference [122]. The error $\eta$ is not solved by using Levinson-Durbin recursion, therefore it is called unconstrained [60].

Next we discuss a general analysis of different types of linear prediction solution approaches.

## 9.5 Analysis of the Signal Models

This analysis is collected from a number of literature that discusses the adaptive signal analysis for the signal model. Non-stationary speech is managed to follow the stationary in a mathematical sense for its analysis. The synonym of the stationary is approximation. Therefore a statistical model is necessary for its analysis. A stationary random process is not a realistic model for speech. As mentioned earlier an approximation, one assumes that speech signals keep their properties in intervals of about 20 ms duration. As a result, a prediction filter for this speech signal has to be updated according to this time frame. Therefore, an efficient algorithm for the inversion of the autocorrelation matrix is crucial for

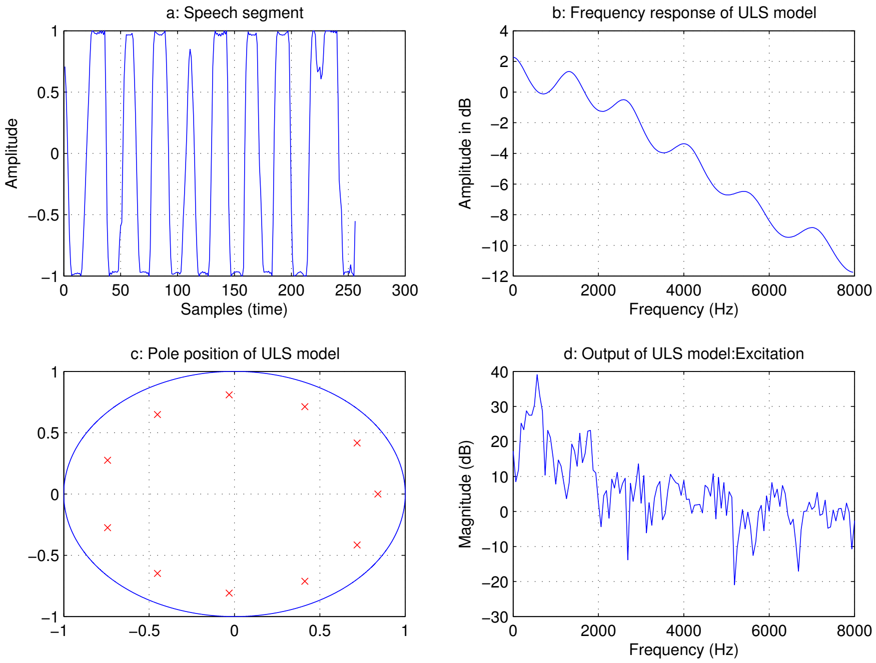


Figure 9.9: Signal model: ULS approach

the application of a predictor.

In our noise solution, we have the Yule-Walker approach for model analysis and the ULS approach for signal model. In figure 9.10, we have shown the analysis between these two models and true AR parameters. We see the ULS approach shows good AR parameters approximation than Yule-Walker approach. The implementation is based on the reference [98].

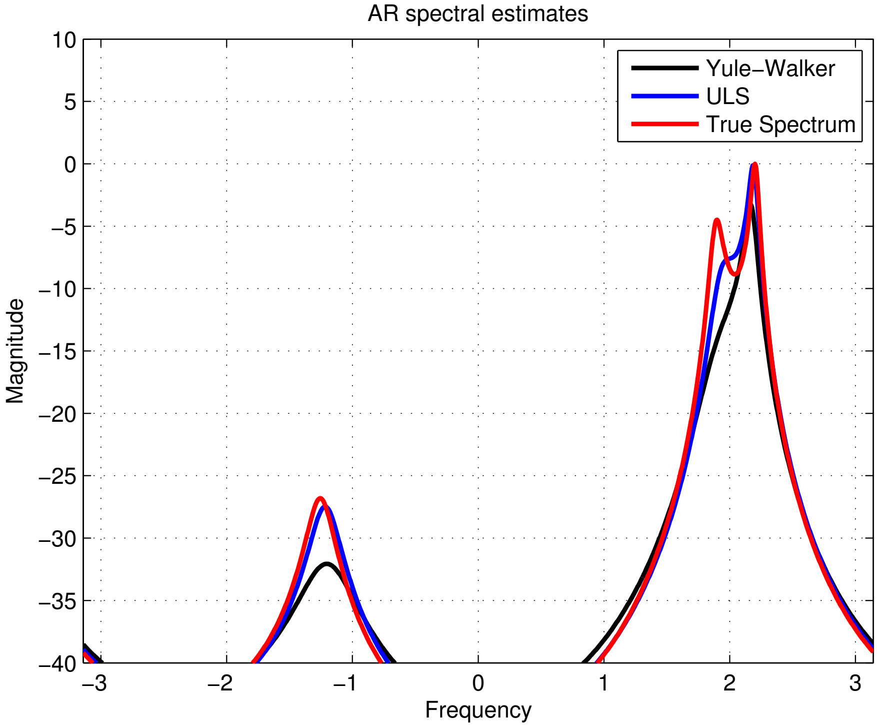


Figure 9.10: Signal model analysis: Yule-Walker approach and ULS approach

Both, the Burg approach and the modified covariance algorithm which is called here as ULS approach are based on the minimization of the forward and backward squared prediction errors. The ULS approach is based on the mini mization of the prediction coefficients. The Burg approach sets constraints on the LP coefficients so that this coefficients satisfy the Levinson recursion and obtain least squares optimization using reflection coefficients in order to solve AR parameters problems. Some problems such as spectral line splitting, bias of the frequency estimates are eliminated in ULS approach. The only problem applying the ULS approach is its weakened stability issue of the LP coefficients but mainly it does not appear when the ULS is structured following stable lattice filters which is used here. Next we summarize some important aspects.

- The ULS approach and Burg approach can be analyzed using the lattice structure. This is useful to capture the physical speech production process efficiently.
- The Yule-Walker introduces poor estimated parameters [89].
- The problems such as line splitting, frequency bias, and spurious or false peaks are observed in the Burg approach.
- The ULS approach may result in instability where the Yule-Walker approach and Burg approach may generate stable model analysis. In spectral estimation, a model stability is not a major concern.

# Chapter 10

# Steady-unsteady Noise Solution

**Outline** In this chapter we discuss how the steady-unsteady time-varying noisy signal is treated for the solution to our noise problem. This is similar to the noisy speech enhancement discussed in [34]. Here our signal model is based on the ULS approach, and the noise is modeled by applying the Yule-Walker equation. The noise is minimized in the sub-bands of the signal. The sub-bands are achieved by an M-band cosine modulated quadrature mirror filter bank (QMF) developed in [132] and the noise is minimized by the spectral minimization method proposed in [38]. Afterwards the Colored Noised Kalman filtering is applied in each sub-band in order to enhance the speech. The signal and the noise models for the Kalman filtering are discussed in chapters 8, 9. This chapter first gives a short overview about the structure of the sub-bands and how this is used, next the spectral minimization algorithm, finally the Kalman filtering operation as a treatment of the steady-unsteady time-varying noisy signal are discussed. We explain the existing algorithms for a complete analysis of our noise problem and its solution. Though the algorithm has already been applied in multimedia signal processing, the explanation of the problem definitions and the explanations of each subparts of the whole application are described in the chapter using our own terminology and applying our own concepts based on the literature review.

## 10.1 The Scenario

The scenario is described first only in some overview. Details will come later. We model the scenario in a natural way: We have noisy speech observations $y[n]$ at time indices $n$ which are mixed by clean speech $s[n]$ and background noise $b[n]$.

Therefore we have the observation $y[n]$ given in equation (10.1).

$$y[n] = s[n] + b[n] \tag{10.1}$$

The main probllem now is: To obtain the clean speech $s[n]$ itself.

The major difficulty is that it is directly unaccessible. At the current situation, one simple way we can obtain it is if we know $b[n]$ and subtract this from the observation but $b[n]$ is unkown too. For this reason the only way out is to estimate $s[n]$. Now the challenge is to provide at least a good or even the best estimate. In the sequel we will approach this challenge.

Our starting point is to estimate $y[n]$ and compare it with the obsevation.

$$\hat{y}[n]$$

For this we assume that we have a model of the situation that allows us to compute the output $y[n]$. For getting this our model tries to reflect the participating parts of the human body, in particular the vocal tract. While such a model is available in its principle structure but it contains unknown parameters. These parameters are approximately obtained in chapter 8 and 9 using the LPC. For this reason, the computed $\hat{y}[n]$ will not coincide with the observed value. It is only an approximation. The goal is to make the approximation as good as possible. This can be done by changing the parameters that are underlying the computation. There are different ways one can attain this unknown values.

First,we return to equation (10.1). Now suppose, we have two different estimated values $s_a[n]$ and $b_a[n]$ as well as $s_b[n]$ and $b_b[n]$ from some $a$ and $b$ obtained at some time index $n$. The best estimated value would be the one that would provide less differences between the observed value and estimated value.

We look at the signal $s$ and noise $b$ in the following two equations.

$$\begin{aligned} s[n] &= \sum_{i=1}^{p} \alpha_i s[n-i] + g_s u_s[n] \\ b[n] &= \sum_{k=1}^{q} \beta_k b[n-k] + g_b u_b[n] \end{aligned} \tag{10.2}$$

The first term on the right side of the first equation ( this is first introduced in chapter 7 in section 7.3, we maintain the same equation number for a convenience) is a linear combination of some past values of $s[n]$ with coefficients $\alpha_i$ for $i = 1, 2, \cdots, p$ that have to be determined. The second term in this equation describes a weighted white noise. We have a linear combination of some past values of $b[n]$ with coefficients $\beta_k$ for $k = 1, 2, \cdots, q$ that have to be determined.

Here we have equation (10.2) for the speech $s[n]$ and equation (10.1) for the noise $b[n]$. These equations holds if we have the coefficients and we do the arithmetic in the right way. As mentioned, initially the coefficients are not known and we have to determine them.

The speech $s[n]$ is modeled by the ULS approach discussed in chapter 9 in section 9.4 and the noise $b[n]$ is modeled by applying the standard Yule -Walker equation discussed in chapter 8 in section 8.3. Both of them use the minimizing the mean squared error (MSE) criteria of the least squares approach. In the MSE criteria the minimum mean squared error between the observed value and estimated value is investigated.

As said, a problem is that the computed $\hat{y}$ is not exactly the observed $y$. Thus we have an optimization problem which results when we make use of the equations (7.12), (10.2) in equation (10.1) in a recursive way.

Given the background of the noisy situation, the handling of the situation is taking place in three different steps that are described in [34].

- Sub-band decomposition
- Noise tracking in the sub-band by spectral noise minimization
- Colored Noise Kalman filtering

In the sub-band decomposition and synthesis stage, each sub-band is attached with a noise tracking by spectral minimization and colored noisy Kalman filtering operation as shown in figure 10.1. This says the signal is first decomposed into m-bands using analysis filter $h_m$ which z-transform is $H_m$. The sub-band signal is then used first for noise suppression by a spectral minima tracking algoithm,. Then an m-bands Kalman filters are used to enhanced the spectrally wighted sub-band signals. After the enhancement, the decomposed signal is synthesized to $x$ using synthesis filter $f_m$ which z-transform in $F_m$. Next, we describe first signal decompositions, then the noise tracking and finally the Kalman filtering operation.

## 10.2 Sub-band Analysis

A sub-band decomposition is a transformation that decomposes the signal into some sub-bands. Each sub-band has the frequency of each band. This is useful to

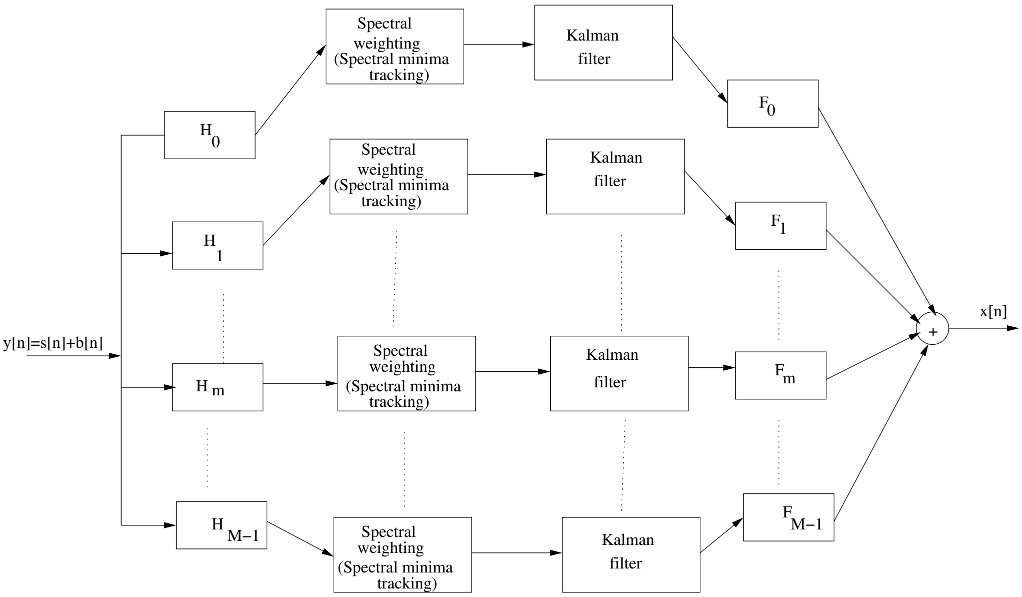

Spectral weighting (Spectral minima tracking)
Kalman filter
$F_0$
$H_0$
Spectral weighting (Spectral minima tracking)
Kalman filter
$F_1$
$H_1$
x[n]
+
y[n]=s[n]+b[n]
$H_m$
Spectral weighting (Spectral minima tracking)
Kalman filter
$F_m$
$H_{M-1}$
$F_{M-1}$
Spectral weighting (Spectral minima tracking)
Kalman filter

Figure 10.1: M-band Kalman filter for colored noise problem

manipulate the information of the signal and to analyze the signal in smaller sections rather than the whole signal. The sub-band decomposition is used in many applications. A common application of sub-band decomposition is a speech coding. A signal can be split into sub-bands in different ways as for instance by applying FFT based filter bank such as quadrature mirror filter (QMF) bank or a wavelet transformation. Some common concepts such as decimation, interpolation, sub-band decomposition, sub-band synthesis are used in the sub-band decomposition of a signal. The decimation is the process of decreasing the sampling rate and the interpolation is the process of increasing the sampling rate.

A basic sub-band decomposition system has an analyzer and a synthesizer. In the analysis, the sampling frequency $f_s$ of input signal $s[n]$ is divided into sub-bands via the analysis filter bank. For example, a two channel based sub-banded signal may have the signal bands $s_0[n]$ and $s_1[n]$. Each sub-band is also known as channel. Each sub-band is decimated at a decimated sampling rate for instance $\frac{f_s}{2}$. In the synthesis section, the decimated signal bands are interpolated via a synthesis filter bank [83], [84], [79].

We only investigate the M-band sub-band approach and do not discuss the other methods. For this purpose we start with the concept of M-band filter banks.

**M-band Filter Banks** We explain here how we apply the M-band quadratic mirror filter band (QMF) to split the signal in to sub-bands. As mentioned before, this is already applied in [34]. This QMF typed sub-band uses the cosine modulated low pass prototype filter in the polyphase FIR filter structure to realize the M-band filter banks with a nearly but not totally perfect reconstruction. Therefore it is called pseudo typed QMF sub-band analysis [7], [6]. The properties of this filter-bank are mentioned in [7] and a list of the main properties is given next.

- The FIR filter is designed using the window method which uses the Kaiser window function [119].
- The filter bank uses a polyphase FIR filter structure (see Appendix).
- The responses are uniform linear phases.
- This structure uses only a single prototype filter and a cosine modulation. Therefore it is simple to design.
- The sampling rate is critical; that means the number of sub-bands is equal to the decimation factor.

In figure 10.2 we see how the signal $s[n]$ is decomposed in the analysis section and regenerated in the sythesis section:

- In the analysis section, $s[n]$ is decomposed into sub-bands $v_m[n]$ for $m = 0, 1, \cdots, M-1$. An analysis filter $h_m[n]$ is used for signal decompositions. This is shown in equation (10.3) (the analysis filter is defined in equation (10.10)). The signal in each sub-band $v_m[n]$ is then down sampled by some factor $i$ and generates $u_m[n]$. This is shown in equation (10.4). Here $i = 32$. The QMF sub-band decomposition has a critical sampling rate. This means that the number sub-bands, the down sampling and the up sampling has same factor; if the sub-band

$$v_m[n] = h_m[n] \otimes s[n] \tag{10.3}$$

$$u_m[n] = \begin{cases} v_m[in] & \text{for} \quad n = 0, \pm i, \pm 2i, \cdots \\ 0 & \text{Otherwise} \end{cases} \tag{10.4}$$

- In the synthesis section we have first the up-sampled sub-banded signal $w_m[n]$ and the up-sample factor is $j$. Here $i$ and $j$ are the same integer valued numbers. This means $j = 32$. This is shown in equation (10.5). The up-sampled sub-banded signals are then synthesized into $x_m[n]$ by the synthesis filter $f_m[n]$ as it is shown in equation (10.6) (the synthesis filter is defined in equation (10.8)). The synthesis signals $x_m[n]$ are then summed up to $y[n]$ which is approximately equal to $s[n]$. This is shown in equation (10.7).

$$w_m[n] = \begin{cases} u_m[\frac{n}{j}] & \text{for} \quad n = 0, \pm j, \pm 2j, \cdots \\ 0 \quad \text{Otherwise} \end{cases} \tag{10.5}$$

$$x_m[n] = f_m[n] \otimes w_m[n] \tag{10.6}$$

$$y[n] = \sum_{m=0}^{M-1} x_m[n] \approx s[n] \tag{10.7}$$

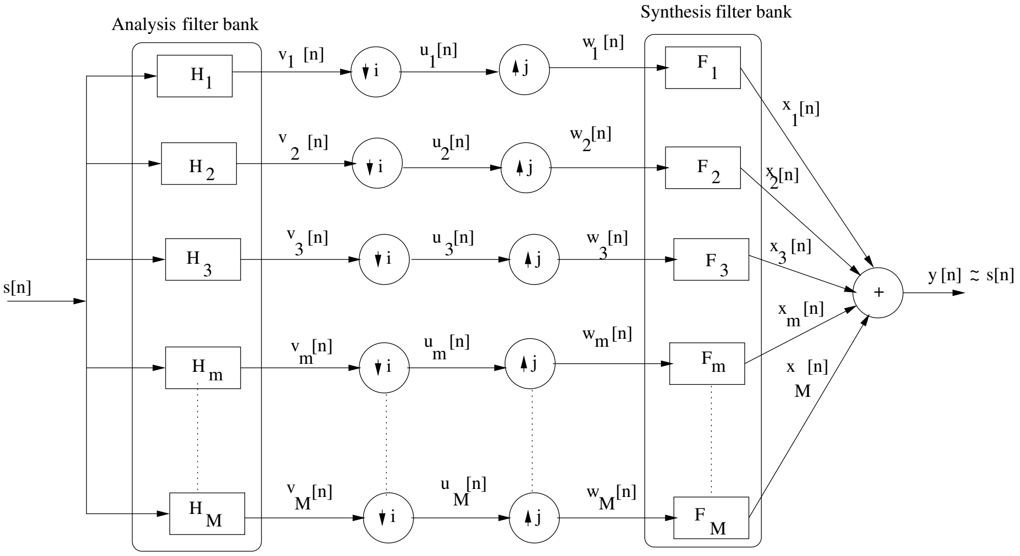


Figure 10.2: Pseudo Cosine Modulated M-Band QMF

The basic elements and properties of QMF are now:

- It uses only one prototype filter for the signal decomposition as the analysis filter and the synthesis filter are mirror images of each other.

- The sampling rate conversion in the analysis and synthesis case are equal to the number of decomposition bands. This is called critical sampling. In this case the up-sampling rate and the down-sampling rate are equal.

- The analysis and the synthesis filter are mirror images of each other. This means if $f_m[n]$ is a synthesis filter, then equation (10.8) is satisfied.

$$f_m[n] = h_m[L - 1 - n] \tag{10.8}$$

**Filter Coefficients** The analysis and synthesis filters are carefully designed to cancel aliasing and imaging distortions shown in equation (10.9). The coefficients of the filters are real. They are derived by a cosine modulation instead of an exponential modulation that happens to be in the discrete time Fourier transform (DFT) based filter banks. The adjacent sub-band aliasing is cancelled by establishing precise relationships between the analysis and synthesis filters $h_m[n]$ and $f_m[n]$. These conditions are given in equations (10.10) and (10.11).

$$\hat{s}[n] = \frac{1}{M} \sum_{l=0}^{L-1} \sum_{m=0}^{M-1} s[m] h_m[lM - n] f_m[l - Mn] \tag{10.9}$$

In equations (10.10) and (10.11), $\Omega = (-1)^m \frac{\pi}{4}$ and $w[n]$ corresponds to the $L$ sample length of the Kaiser window function.

$$h_m[n] = 2w[n] \cos\{\frac{\pi}{M}(m + 0.5)(n - \frac{L-1}{2}) + \Omega_m\} \tag{10.10}$$

$$f_m[n] = 2w[n] \cos\{\frac{\pi}{M}[(m + 0.5)(n - \frac{L-1}{2}) - \Omega_m\} \tag{10.11}$$

At the analysis stage, the input signal $s[n]$ is processed by a $(L-1)^{th}$ order FIR filter. The input is divided into $M$ sub-bands in the analysis stages. In the synthesis stage, $y[n]$ is a combination of the sub-bands. The synthesis filter is the mirror image of the analysis filter. Both the analysis and synthesis section uses FIR filter based on Kaiser window function.

## 10.3 Spectral Minima Tracking in Sub-bands

Now the noise level in each band in the M-band signals is tracked down to its minimum following the spectral minimum tracking algorithm described in [38]. The algorithm is based on the spectral amplitude estimation in the sub-bands. This estimation is based on the minimum mean squared error estimation (MMSE). The goal is here to track the local minima of the noisy measurement signal by computing the local minima of the speech signal by using some constants. These are determined experimentally.

The spectral minima tracking is a type of spectral weighting. This is a spectral amplitude estimator. This is a typed of minima tracking in each sub-band. This attenuates different spectral regions of the mixed noisy speech signal and the noise with different constant known factors. An aim of this is to obtain less noisy signal. In this approach the noise estimate is updated continuously tracking the minima of the noisy speech in each sub band. The concept is that ehe noise estimate increases when ever the noisy speech power increases.

We use the power spectrum and first introduce the basic notions. In equation (10.12), $P_{s_m}(k)$ denotes the power spectrum of the $m^{th}$ band signal $S_m(k)$. It is the $k^{th}$ spectral component of $m^{th}$ sub-band of the noisy speech signal. This has some how similarity the spectral subtraction method. In the equations we see first the smoothed power spectral density of the observation using smoothing factor rho and the spectrum of the observation. The noise spectral density is updated until it equals to power spectral density of the observation.

$P_{b_m}(k)$ is the noise power spectrum of the $B_m(k)$. Here B is the noise vector. It is the $k^{th}$ spectral component of the $m$ sub-band. The short time noise power spectrum is needed to estimate the spectral amplitude of the noisy signal. The noise spectrum estimation performs some type of temporal minima tracking of $P_{s_m}(k)$. This spectral estimator has a build-in minimum tracking $\tau$ term in equation (10.14).

In equation (10.12), $P_{s_m}(k)$ is smooth noisy signal power spectral density of $k$ component at $m$ band. This is also the local minima of the noisy speech signal. $|Y_m(k)|$ is noisy signal spectrum of $k$ component at $m$ band. In equation (10.14), $P_{b_m}(k)$ is noise power ppectral density of $k$ component at $m$ band $\rho$ is the smoothing factor and typically it is from 0.7 to 0.9 which is selected experimentally. $\tau$ is the look a head factor which controls the adaptation time of the local minima $P_{s_m}(k)$. Typical parameter selections for $\rho = 0.7$, $\tau = 0.96$, and $\upsilon = 0.998$ in order to adapt the noise in each sub-bands.

- Smoothed sub-band power spectrum is given in equation (10.12):

$$P_{s_m}(k) = \rho P_{s_{m-1}}(k) + (1-\rho)|Y_m(k)|^2 \tag{10.12}$$

- If noise power spectral density $P_{b_m}(k)$ at $m$ band which has frequency component $k$ is less than than the smoothen signal as shown in equation (10.13), then it is updated by equation (10.14).

$$P_{b_{m-1}}(k) < P_{s_m}(k) \tag{10.13}$$

- If $P_{b_m}(k)$ is equal to smoothed noisy speech power density spectrum $P_{s_m}(k)$ i.e. eq (10.15), then it stops updating.

$$P_{b_m}(k) = \upsilon P_{b_{m-1}}(k) + \frac{1-\upsilon}{1-\tau}(P_{s_m}(k) - \tau P_{s_{m-1}}(k)) \tag{10.14}$$

- Else

$$P_{b_m}(k) = P_{s_m}(k) \tag{10.15}$$

## 10.4 Kalman Filter

One can estimate something before an event has happened (priori) or after an event has happened (posteriori). In our context this leads to the next concepts. The detailed of this derivations can be found in [21], [127].

### 10.4.1 State space derivation

First we reformulate the equations formulated for $y[n]$, $s[n]$ and $b[n]$ in terms of vectors and matrices where we use bold face letters. The state information is written in the state space form in equation (10.16). In the equation, $n$ is the time varying index and the state vector $\mathbf{s}$ at time $n+1$ is a $p$ length vector of the linear combination of $p$ previous vectors using a $p \times p$ dimensional matrix $\mathbf{A}$ and some additional disturbance vector $\mathbf{u}$ of length $p$ that is modeled as zero mean random white noise which is perpetrated by a $p \times 1$ dimensional matrix $\mathbf{g}_s$ at time $n$.

$$\mathbf{s}[n+1] = \mathbf{A}[n]\mathbf{s}[n] + \mathbf{g}_s[n]\mathbf{u}[n] \tag{10.16}$$

Equation (10.16) is rewritten in equation (10.23).

$$\mathbf{s}[n+1] = \gamma^T[n]\mathbf{s}[n] \tag{10.17}$$

The steady background noise defined by an AR process is written in equation (10.18). In the equation, $n$ is again the time index. The equation says if we look at the noise vector $\mathbf{b}$ at time $n+1$, then it is a linear combination of a $q \times q$ dimensional matrix $\mathbf{D}[n]$ and noise vector $\mathbf{b}$ at time $n$ with an additional disturbance $\nu$ which is perpetrated by a $q \times 1$ dimensional matrix $\mathbf{g}_b$ at time $n$. $\nu$ is modeled as zero mean random white noise.

$$\mathbf{b}[n+1] = \mathbf{D}[n]\mathbf{b}[n] + \mathbf{g}_b[n]\nu[n] \tag{10.18}$$

Using an abbreviation $\psi$, equation (10.18) is rewritten in equation (10.19). $'$ denotes a transpose symbol.

$$\mathbf{b}[n+1] = \psi^{'}[n]\mathbf{b}[n] \tag{10.19}$$

The definitions of $\mathbf{s}[n]$, $\mathbf{b}[n]$, $\mathbf{g}_s$ , $\mathbf{g}_b$, $\gamma$, $\psi$ as well as of the matrices $\mathbf{A}[n]$ and $\mathbf{D}[n]$ are given below:

$$\begin{aligned}
\mathbf{s}^{'}[n] &= \big[s[n-1], s[n-2], s[n-3], \cdots, s[n-p+1], s[n-p]\big]_{p\times 1}\\
\mathbf{b}^{'}[n] &= \big[b[n-1], b[n-2], b[n-3], \cdots, b[n-q+1], b[n-q]\big]_{q\times 1}\\
\mathbf{g}_s^{'} &= \big[0,\ 0, \cdots, 0, 1\big]_{p\times 1}; \quad \gamma^{'} = \big[0,\ 0, \cdots, 0, 1\big]_{p\times 1}\\
\mathbf{g}_b^{'} &= \big[0,\ 0, \cdots, 0, 1\big]_{q\times 1}; \quad \psi^{'} = \big[0,\ 0, \cdots, 0, 1\big]_{q\times 1}
\end{aligned}$$

$$\mathbf{A}[n] = \begin{bmatrix} 0 & 1 & \dots & \cdots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \\ \alpha_1 & \alpha_2 & a_3 & \dots & \alpha_p \end{bmatrix}_{p\times p} \quad \mathbf{D}[n] = \begin{bmatrix} 0 & 1 & \dots & \cdots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \\ \beta_1 & \beta_2 & \beta_3 & \dots & \beta_q \end{bmatrix}_{q\times q} \tag{10.20}$$

The $\alpha_i$ and $\beta_i$ are coefficients to be determined. How these coefficients are determined using the ULS approach and the Yule-Walker equation are explained in chapter 8 and chapter 9.

The observation equation is given in equation (10.21). In the equation, $n$ is the varying time index. The equation says if we look at the observation vector $\mathbf{y}$ at time $n$, then we see that it is a linear combination using a $q \times p$ dimensional matrix $\mathbf{C}[n]$ at time $n$ from the previous $q$ states of the state $\mathbf{s}[n]$ and disturbance which is modeled as a colored noise $b$. In this equation, a measurement disturbance matrix $\mathbf{G}[n]$ times measurement disturbance $\mathbf{w}[n]$. The noisy observation vector given by $\mathbf{y}$ at time $n+1$ is shown in a state space form in equation (10.21).

$$\mathbf{y}[n+1] = \mathbf{C}[n]\mathbf{y}[n] + \mathbf{G}[n]\mathbf{w}[n] \tag{10.21}$$

The detailed of equation (10.21) is:

$$\mathbf{C}[n] = \begin{bmatrix} \mathbf{A}[n] & \mathbf{0} \\ \mathbf{0} & \mathbf{D}[n] \end{bmatrix}, \mathbf{y}[n] = \begin{bmatrix} \mathbf{s}[n] \\ \mathbf{b}[n] \end{bmatrix}, \mathbf{G}[n] = \begin{bmatrix} \mathbf{g}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{g}_b \end{bmatrix}, \mathbf{w}_n = \begin{bmatrix} \mathbf{u}[n] \\ \mathbf{v}[n] \end{bmatrix}$$

$\mathbf{u}_n$ and $\nu_n$ are zero-mean white noise sequences. The covariance of $\mathbf{u}_n$ is $\sigma_u^2[n]$ and the covariance of $\nu_n$ is $\sigma_b^2[n]$ such that equation (10.22) hold and summarized $\mathbf{W}[n]$ is covariance matrix of $\mathbf{w}[n]$.

$$\mathbf{W}[n] = \begin{bmatrix} \sigma_u^2[n] & \mathbf{0} \\ \mathbf{0} & \sigma_b^2[n] \end{bmatrix} \tag{10.22}$$

Now the The state $\mathbf{s}$ and the noise $\mathbf{b}$ are not correlated. The noisy observation vector $y$ at time $n$ is a sum of speech s and noise b. Here we introduce a matrix $\mathbf{H}[n]$ to rewrite the state space definition of the observation $\mathbf{y}$. Now our observation vector is written in equation (10.23).

$$\text{Observation :} y[n] = \mathbf{H}[n]^T \mathbf{y}[n] = \begin{bmatrix} \gamma[n] & \psi[n] \end{bmatrix} \begin{bmatrix} \mathbf{s}[n] \\ \mathbf{b}[n] \end{bmatrix} \tag{10.23}$$

Given the model parameters, the computations of the M-band colored noisy Kalman filtering operation is taking place in two steps. For this we read the literature [31], [46], [127], [21], [10]. The computational steps are:

- Prediction of the measurement.
- Update the predicted estimation.

The computational steps are discussed next.

### 10.4.2 Prediction Estimates

The prediction estimate of $y[n]$ at time $n$ given the value for $n-1$ is $y[n|n-1]$. $\hat{y}[n|n-i]$ is the predicted value of $y[n]$ based on the observation samples up to time $[n-i]$. In order to estimate the error, we have to consider the development of the error over time.

The innovation or the error signal $e[n|n]$ is given in equation (10.24).

$$e[n|n] = y[n] - \hat{y}[n|n] \tag{10.24}$$

Similar to equation (10.24), we have equation (10.25).

$$e[n|n-1] = y[n] - \hat{y}[n|n-1] \tag{10.25}$$

The prediction equation is shown in equation (10.26).

$$\hat{\mathbf{y}}[n|n-1] = \mathbf{C}[n-1]\hat{\mathbf{y}}[n-1] \tag{10.26}$$

By using equation (10.26), equation (10.25) can be derived as shown in equation (10.27).

$$\begin{aligned} \mathbf{e}[n|n-1] &= y[n] - \hat{y}[n|n-1] \\ \mathbf{e}[n|n-1] &= \mathbf{C}[n-1]\hat{\mathbf{y}}[n-1] + \mathbf{G}[n]\mathbf{w}[n] - \mathbf{C}[n-1]\hat{\mathbf{y}}[[n-1]|[n-1] \\ \mathbf{e}[n|n-1] &= \mathbf{C}[n-1]\{[\hat{\mathbf{y}}[n-1] - \hat{\mathbf{y}}[[n-1]|[n-1]]\} + \mathbf{G}[n]\mathbf{w}[n] \end{aligned} \tag{10.27}$$

Equation (10.27) can be re-written as shown in equation (10.28).

$$e[n|n-1] = \mathbf{C}[n-1]\mathbf{e}[(n-1|n-1)] + \mathbf{G}[n-1]\mathbf{w}[n-1] \tag{10.28}$$

The covariance matrix of the prediction error $e[n|n-1]$ is shown in equation (10.29). In the equations $'$ denotes the transpose.

$$\mathbf{P}[n|n-1] = E\{\mathbf{e}[n|n-1]\mathbf{e}[n|n-1]^{'}\} \tag{10.29}$$

Similar to equation (10.29), we have equation (10.30).

$$\mathbf{P}[n|n] = E\{\mathbf{e}[n|n]\mathbf{e}^{'}[n|n]\} \tag{10.30}$$

Substituting the value of the prediction error shown in equation (10.29), the prediction error covariance matrix $\mathbf{P}[n|n-1]$ is shown in equation (10.31) [21].

$$\mathbf{P}[n|n-1] = \mathbf{C}[n-1]\mathbf{P}[n-1|n-1]\mathbf{C}^{'}[n-1] + \mathbf{G}W[n]\mathbf{G}^{'} \tag{10.31}$$

### 10.4.3 Update Predicted Estimation by Correction

The next step is then to estimate the current estimate $\hat{y}[n|n]$ from $\hat{y}[n|n-1]$. This leads to the new state update estimation equation shown in equation (10.32).

$$\hat{\mathbf{y}}[n|n] = \hat{\mathbf{y}}[n|n-1] + \mathbf{K}[n](\mathbf{y}[n] - \mathbf{H}^{'}[n-1]\hat{\mathbf{y}}[n|n-1]) \tag{10.32}$$

In equation (10.32), the Kalman gain matrix $\mathbf{K}[n]$ has to be computed and how this is computed is shown in equation (10.33).

$$\mathbf{K}[n] = \mathbf{P}[n|n-1]\mathbf{H}[n-1](\mathbf{H}^{'}[n-1]\mathbf{P}[n|n-1]\mathbf{H}[n-1])^{-1} \tag{10.33}$$

The state variable estimation error, also called the innovation signal, is now shown in equation (10.34). The innovation is a mixture of the signal and the noise [127].

$$e[n|n] = (\mathbf{y}[n] - \mathbf{H}^{'}[n])\hat{\mathbf{y}}[n|n-1] \tag{10.34}$$

For the covariance matrix of estimation error we have equation (10.35) and $\mathbf{I}$ is the identity matrix.

$$\mathbf{P}[n|n] = (\mathbf{I} - \mathbf{K}[n]\mathbf{H}^{'}[n])\mathbf{P}[n|n-1] \tag{10.35}$$

Using the Kalman gain for the estimation of the prediction, we can estimate the clean signal denoted as $\hat{s}$ by equation (10.36).

$$\hat{s}[n] = \mathbf{H}^{'}\hat{\mathbf{y}}[n|n] \tag{10.36}$$

In figure 10.3, we see the signal flow diagram of the Kalman prediction and estimation for the color noisy speech signal. Here, we see the observation consists of $s$ and noise $b$. These are predicted first and then estimated using the Kalman gain $K$ in order to generate estimated $s$. In this diagram, we see the observation consists of mixed speech and noise. Both of them are modeled by AR approaches. The observation is estimated, updated and corrected by Kalman gain matrix $K$.

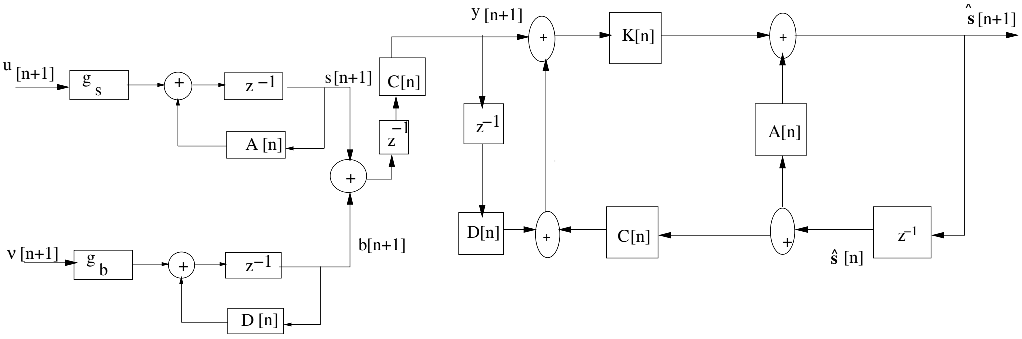


Figure 10.3: Signal Flow in the Kalman filtering

In figure 10.4, we see that time varying steady-unsteady noisy signal is enhanced. For this we used 32 sub-bands where each band is 1024 length. The noise is minimized and the Kalman filter is applied in each sub-band. In this figure, a is noisy spoken command Offne die Tuer and b is an enhanced version of this in the time domain. The amplitude and sound level in c and d of the noisy and enhanced signal are then measured using A-weighting filter (discussed in chapter 3). This filter is used in practice for instance in the acoustic control. The spectrum of the noisy and enhanced signal are shown in e and f. We computed the SNR of this experiment and it is 21.

## 10.5 Analysis and Evaluations

In this section, we analyze, investigate and compare some speech enhancement approaches to evaluate our time varying noise treatment approach. These experiments and comparisons are done using our own data. In the figures, the standard enhanced speech means the application of standard pre-emphasis filter and the corresponding technique, the specially enhanced signal means the applications of the redundancy removal approach, the pre-emphasis filter, the mached filter and the corresponding technique.

### 10.5.1 Wiener Filter

The Wiener fillter is a kind of competition to the present appoach and therefore we consider it. In the Wiener filter, $y[n]$ in equat (10.1) is mixed with the desired signal $s[n]$ and an additive noise $b[n]$. A common choice of $b[n]$ is an additive white Gaussian noise. Then $y[n]$ is estimated by the coefficients $h_i$. The signal $s[n]$ is a short time signal. This means the signal is windowed by a Hamming windowing (see chapter 8). This generates an estimate of $y[n]$ which is denoted by $\hat{y}[n]$ and the difference between $y[n]$ and $\hat{y}[n]$ is an error $e[n]$. The error is minimized by using mean squared error (MSE) shown in equation (10.39). The accepted signal is the one which give the minimum mean squared error (10.39). More on the Wiener filter can be found in [126]. (10.37).

$$\hat{y}[n] = \sum_{i=0}^{p-1} h_i y[n-i] \tag{10.37}$$

$$e[n] = y[n] - \hat{y}[n] \tag{10.38}$$

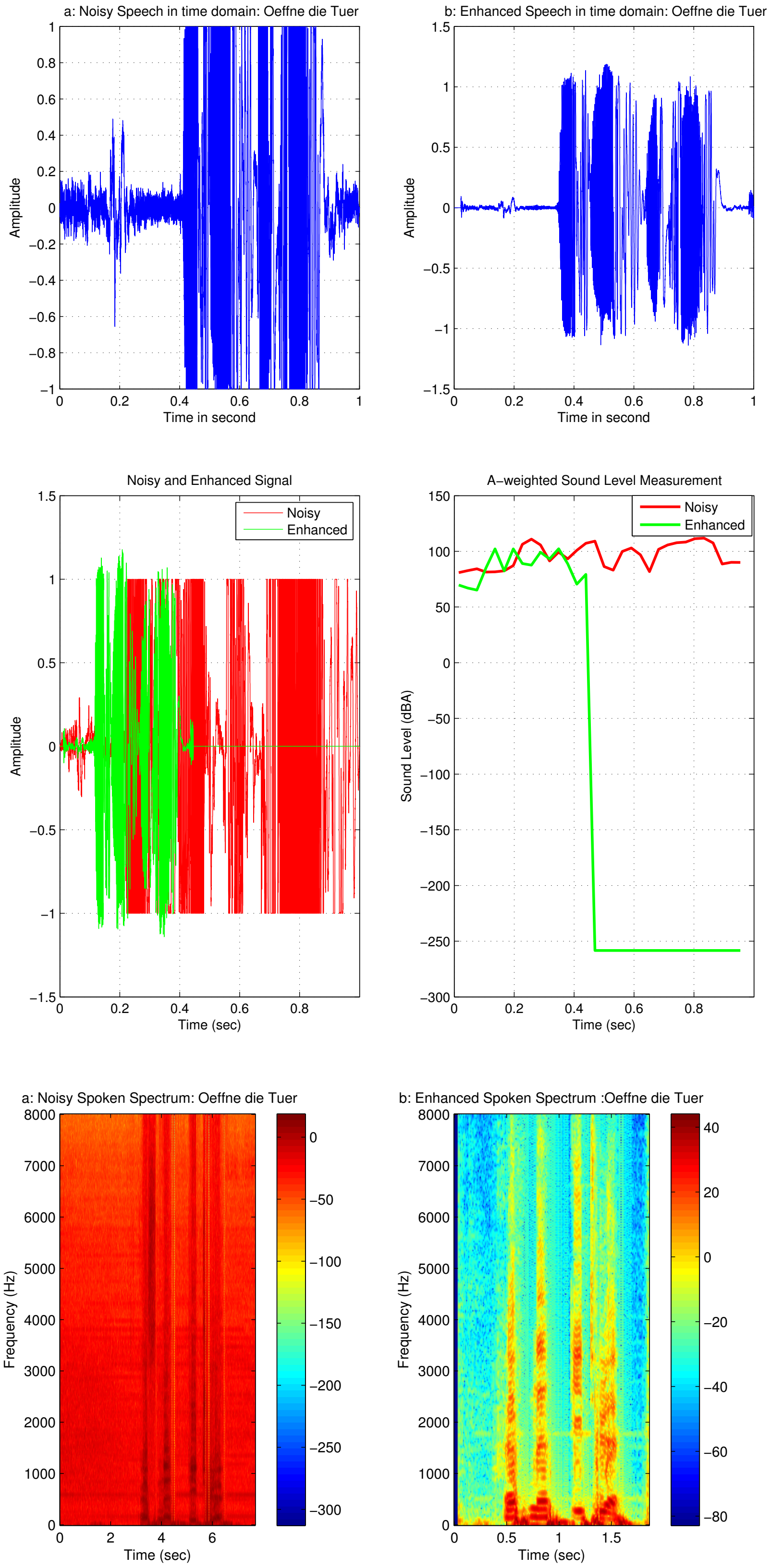


Figure 10.4: Hybrid noisy speech and M-band Kalman filter

$$E\{e^2\} = E\{(y - \hat{y})^2\} \tag{10.39}$$

In figure 10.5 we see the result of applying the Wiener filter. In the figure, in a we see the speech signal which is enhanced by using a standard pre-emphasis filter (see chapter 4). In the same figure, in b, we see our approach that is the redundancy removed signal is used for strong noise removal by applying the matched filter and then this signal is enhanced by using the Wiener filter. The enhanced signal created an additional noisy rhythmic sound. The Wiener filter is not effective for our hybrid noisy signal. The SNR in this case is approximately -1.04 (explained in chapter 3).

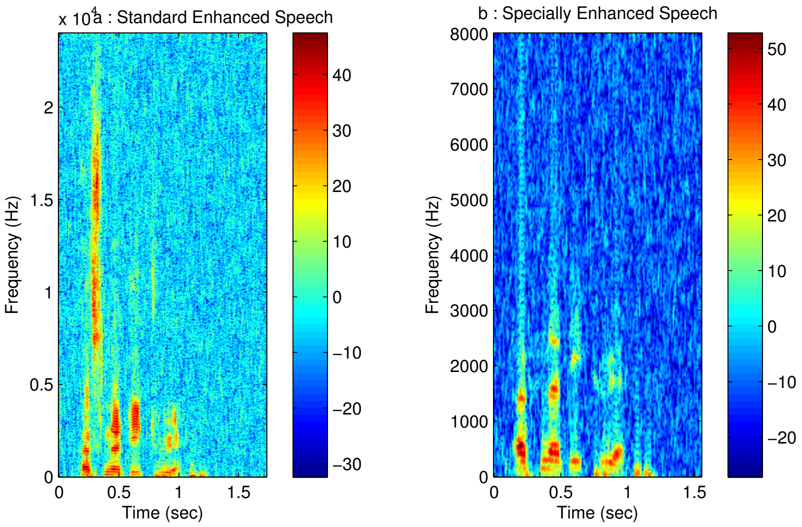


Figure 10.5: Evaluation of Wiener filter and its output

### 10.5.2 Spectral Subtraction

One of the most commonly standard noise reduction methods is the spectral subtraction. In this method the magnitude spectrum of the noise is subtracted from the magnitude spectrum of the noisy speech signal. In equation (10.40), $Y(k)$, $S(k)$ and $B(k)$ are the Fourier transforms of $y[n]$, $s[n]$ and $b[n]$.

$$Y(k) = S(k) + B(k) \tag{10.40}$$

An estimate of the enhanced magnitude spectrum of the signal $\hat{S}(f)$ can then be found subtracting the magnitude spectrum of the noise spectrum from the magnitude spectrum of the observation signal $Y(k)$. The constants are $\zeta = 1$ and $\theta = 1$.

$$|\hat{S}(k)|^{\zeta} = |Y(k)|^{\zeta} - \theta|B(k)|^{\zeta} \tag{10.41}$$

In figure 10.6, in a we see the noisy signal enhanced by the spectral subtraction method and in b we see the noisy signal is specially enhanced. The spectral subtraction method did not enhance the speech signal rather this added an additional noisy sound. The SNR is approximately -3.47.

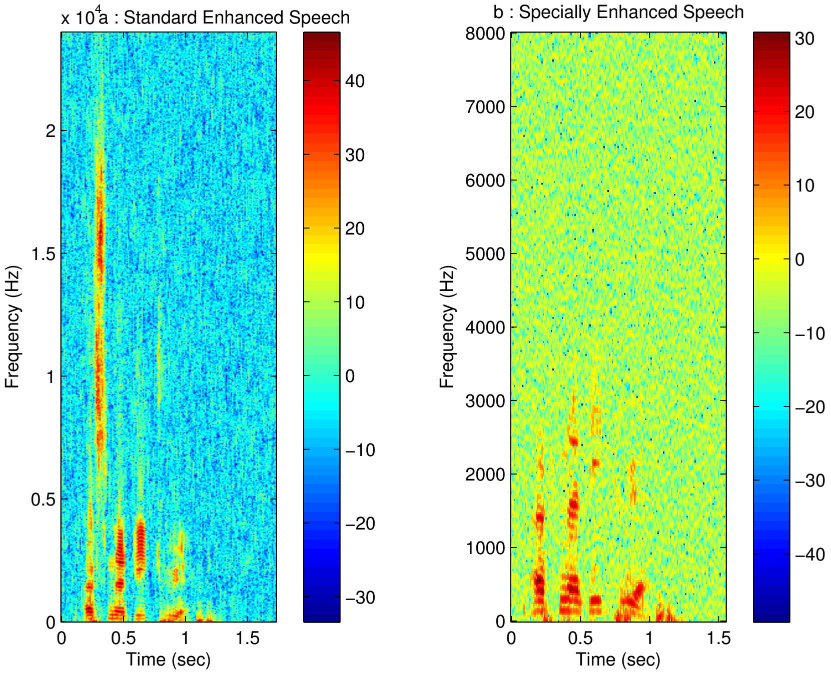


Figure 10.6: Evaluation of spectral subtraction and its output

### 10.5.3 White Noise Kalman Filtering

The Kalman filter is first applied to the speech signal assuming the speech is corrupted by the white noise [72]. The model is shown in equation (10.42). The system noise $\mathbf{w}[n]$ is a white Gaussian noise. This has zero mean and unit variance. The measurement noise $\mathbf{v}[n]$ is an additive noise which is also zero mean and has known variances. Here the signal model is based on the Yule-Walker equation and the transition matrix **A** is a $p$ times $p$ dimensional coefficient matrix, system matrix units **B**, **C** is a $p$ times 1 dimensional matrix and **s** is $p$ times 1 dimensional vector.

$$\begin{aligned} \mathbf{x}[n+1] &= \mathbf{A}[n]\mathbf{s}[n] + \mathbf{B}[n]\mathbf{w}[n] \\ \mathbf{y}[n] &= \mathbf{C}[n]\mathbf{x}[n] + \mathbf{v}[n] \end{aligned} \tag{10.42}$$

Applying this Kalman filter to our hybrid noisy speech, we obtained a degraded speech signal. We observed this when using the standard noise reduction technique and also using our approach which is a redundancy removal, pre-emphasizing and strong noise removal and then applying Kalman filter. In figure 10.7, we see the result of this type of Kalman filter in the hybrid noisy our speech. This filter is not useful for our application. The SNR is approximately -1.187.

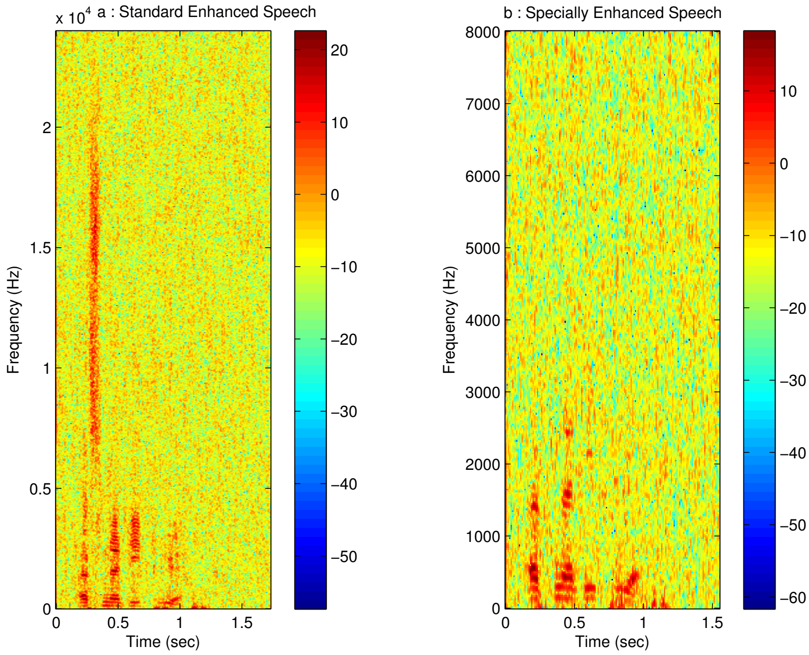


Figure 10.7: Evaluation of white noise of the Kalman filter and its output

### 10.5.4 KEM Filtering using White Noise

In equations (10.16) the system noise $u[n]$ is white noise and in equation (10.23), the measurement noise $w[n]$ is a known additive noise. The signal model is based on Yule-Walker equation. The states and parameters which are the coefficients of the signal model are optimized by expectation-maximization approach in the E-step and M-step. This is a similar approach that is described in [116] (introduced next) except that the noise is white and additive. We see the application of this approach to our signal. This is an inefficient approach for the hybrid noisy speech. More over the computation time for a single speech signal which has about 26000 samples, this approach takes 2 minutes. The SNR is approximately -1.36.

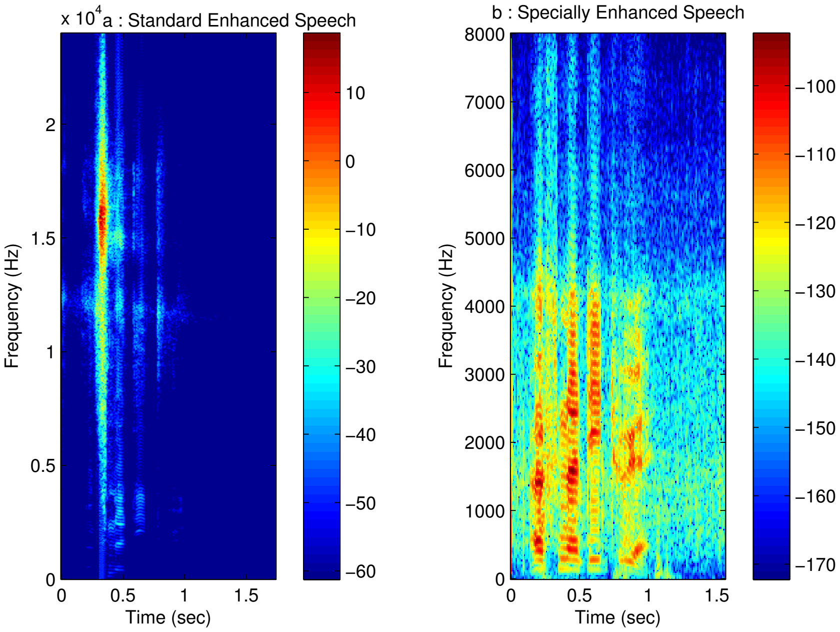


Figure 10.8: Evaluation of white noise of the Kalman filter using EM approach and its output

### 10.5.5 KEM Approach for Colored Noise

In [116], the colored noisy speech is enhanced by a Kalman filter using an expectation-maximization (EM) approach. Considering our state space derivations in equations (10.16) and (10.23), the signal model is based on the Yule-Walker equation, the system noise $u[n]$ is the colored noise.This is also modeled by the Yule-Walker equation. The enhanced speech is then recognized by an HMM based speech recognition system. Typical noise is computer fan noise, noise in the lab, typical office environment. The type of the ASR system is single word speech recognition system. The noisy speech is enhanced by the Kalman filter. The speech signal modeled by using the standard Yule-Walker approach and the noise is considered as colored. The noise is modeled by the AR approach which parameters are obtained by using the Yule-Walker approach. This enhances the speech using EM approach iteratively and by estimating the state expectation and covariance matrix in the E-step and parameters of the signal and noise model are estimated in the M-step. This approach is also known as KEM. In figure 10.9, we see the result of applying the KEM. This is an inefficient approach for our hybrid noisy signal. More over it has a high computational time which is about 3 to 5 minutes or more for a single speech signal which has a length of 25000 to 60000 samples.

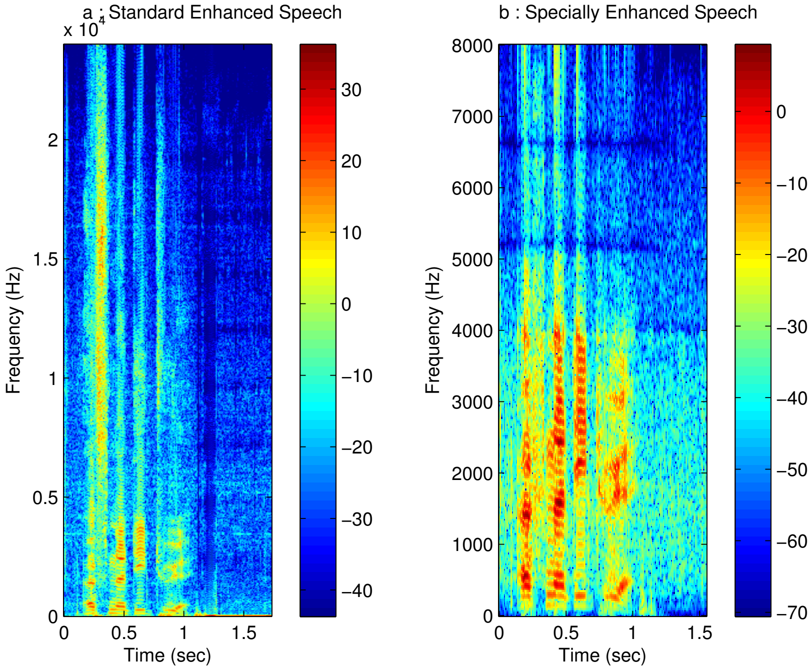


Figure 10.9: Evaluation of color noise of the Kalman filter using EM approach and its output

### 10.5.6 FFT based Suband Decomposition and Kalman Filtering

In this approach with respect to our state space derivations in equations (10.16) and (10.23), the signal model $s[n]$ is based on the Burg approach and the noise model $b[n]$ is based on the Yule-Walker approach which is estimated in the subband. The signal sub-bands are obtained by an FFT based filter bank. This approach is described [46]. How the FFT based filterbank is used to decompose the signal can be found in [140]. This enhances our hybrid noisy speech to some extent. We see the result of its application to our noisy data in figure 10.10 where a shows the enhanced speech applying the pre-emphasized FFT based sub-band decomposition and Kalman filter at each band and b shows our noise reduction approach by redundancy removal, pre-emphasis, matched filter and then the Kalman filtering on sub-band decomposed signals. The SNR is 6.09.

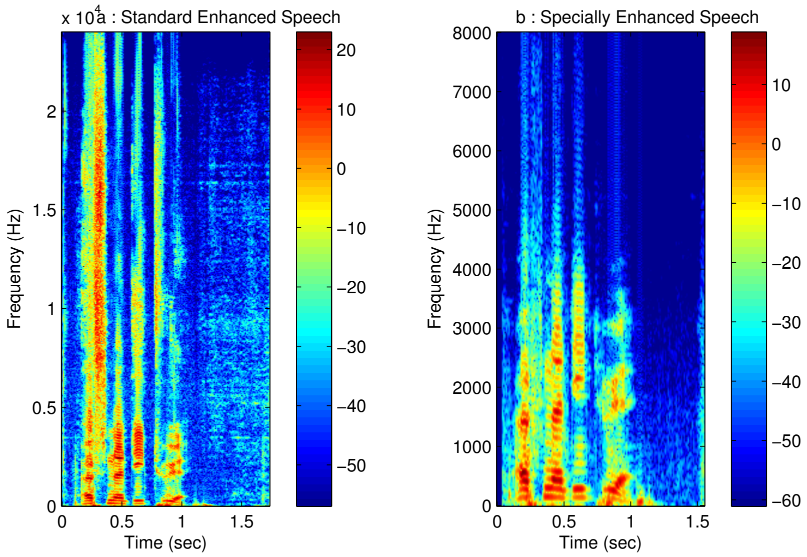


Figure 10.10: Evaluation of color noise in the sub-band Kalman filter and its output

### 10.5.7 Mband Colored Noise and Kalman Filtering

Here we see the result of spectral minimization of the noise and the Kalman filter on each sub-band in the M-band signal in figure 10.11. This enhances the speech and we use this enhanced speech for the feature extraction. The SNR is 10.49.

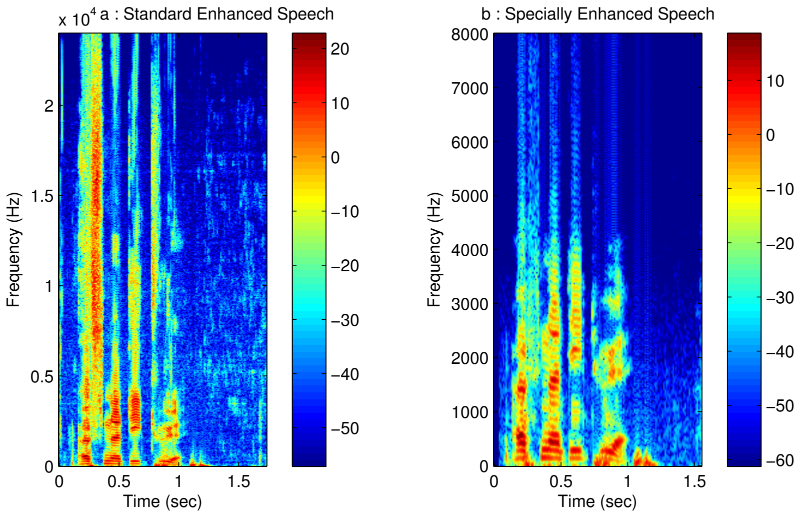


Figure 10.11: Evaluation of color noise of the Kalman filter using color noise Mband filter bank and spectral minimization and its output

### 10.5.8 Principle Component Analysis (PCA) Approach

The de-noising operation of our noisy speech is done by applying PCA. The PCA is an eigen value analysis tool. This extracts the meaningful basis of the noisy redundant signal by searching for the principle components in a coordinate system using the coordinates in order to simplify a complex expression to a simpler one. The principal components (PC) are the linear combination of the basis vectors. The detailed of this is described in [124]. Here the noisy signal is de-noised but the signal loses its useful information. This is not an efficient noise reduction approach for our hybrid noisy speech while feature extraction is used for the recognition. There is an option to apply this PCA de-noised signal for the classification and recognition. But we have not investigated this further. The result of this PCA application is shown in figure 10.12.

In the Wiener filtering, spectral subtraction method, different versions of Kalman filter such as an iterative Kalman filter for the colored noise or a Kalman filter for the white noise or for an assumption of stationary color noise did not solve our noise problem and rather degraded the spoken command.

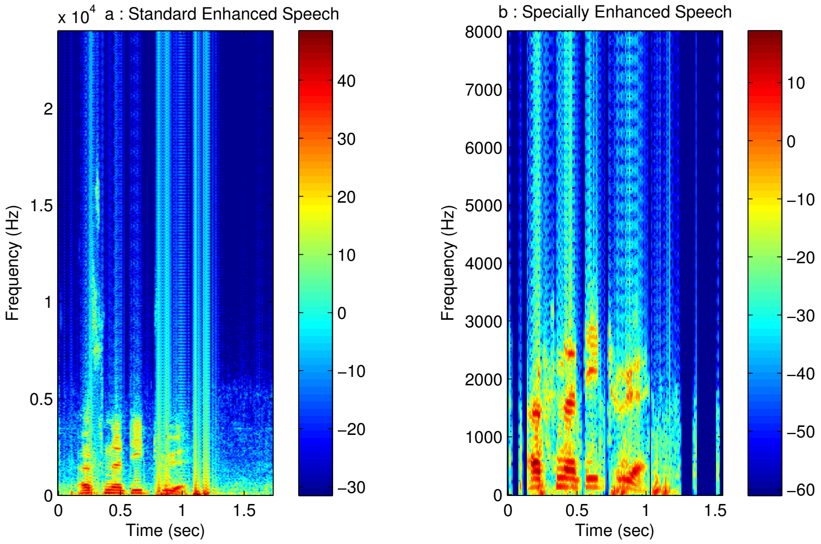


Figure 10.12: Evaluation of white noise applying PCA and its output

# Chapter 11

# Psychoacoustics and DANSR System

**Outline of the Chapter** We provide some essential psychoacoustics information and basic definitions in this chapter. We explain their necessity and inclusion in the feature extraction in order to extract perceptual features. We introduce the psychoacoustics quantities that are adapted to the DANSR's feature extraction technique. We are not concerned with the human interpretation of psychoacoustic elements and we restricted ourselves to the basic quantities on top of which the interpretation has to be built.

Speech contains more information then expressed in the meaning of the words. Humans are often able to recognize this information when they obtain the sound data. Our task is to identify these data quantities so that a machine is able to process them further. For this purpose we discuss properties of the human ear. This is a very complex device and we make the discussion as short as possible. We mention only aspects where we have taken advantage of. Nevertheless, there are quite many of them.

A major reason to adapt the psychoacoustic quantities to take the speech signals that are perceptually relevant only. The collected data have redundancies. In the percpetual feature extraction discussed in the following chapters, the redundancies are avoided following the human speech perception by the ear. By this we simply mean that speech we hear is analyzed, extracted and compressed naturally in the human hear where the redundant irrelevant information are filtered out and the speech is perceptually relevant and meaningfully audible. In the perceptual feature extraction, the goal is to mimic the perception process of human hearing removing the redundant information but keep only the perceptually meaningful relevant compact information for recognition.

## 11.1 Psychoacoustics for DANSR

The speech sounds arrive in a random process and it shows variations both in the spectral and temporal analysis. The human ear and brain together analyse the frequency of the speech. The outer ears accept speech sound pressure waves and send this through the middle ear to the inner ear. This transforms finally the sound to the brain. Thus we hear and recognize the speech by their collaborative work.

Spoken language contains more information than a written text. This information is hidden in the speaking style but contained in the wave forms obtained by the receiver. This information is not discovered by the speech recgnition discussed so far. It is the topic of psychoacoustics that is a bigger area in itself in which we will not introduce and give only some general remarks. Instead we will rather describe the quantities that need to be extracted. In this chapter we mention a number equations from psychoacoustics area. We took them from the literature without explanations. From the principle point of our approach, the corresponding derivations, values of specific constants and explanations do not play a role.

In the inner ear, the basilar membrane is working as a spectrum analyzer. By responding to the temporal variation of the sound pressure wave and its localizations, the human ear responds to temporal variations of pressure and localizes the sound. The frequency, timing, amplitude, loudness and phase information at different frequency ranges and the localization of the sound sources are determined by the brain.

The study of the psychoacoustics is related to the perception of the sound and related phenomena. The speech signal in the temporal and spectral intervals i.e. between 100 to 1000 ms is generally analyzed by going through the audio sensation and its variations as well as the loudness-time function of the psychoacoustics [113]. In chapter 2 in section 2.2, we have introduced into the role of the human ears in recognizing speech.

The purpose of using the psychoacoustics quantities is to approximate the mapping of the signal in the auditory system. This type of auditory or perceptual modeling approximates the use of the masking threshold, loudness scales, sound pressure level etc. The masking threshold is the limit that makes one signal more audible than the others. The perceptual measurement needs to be connected to the feature extraction. To adopt some basic psychoacoustics quantities, we first reviewed the human auditory system and its functioning, then the hearing model that is used in [12]. The temporal time-frequency resolution and the threshold masking are common adaptations for the perceptual spectral analysis [42], [110]. The perceptual entropy is used for speech coding and multimedia application for speech data compression in [90], [6], [105], [58], [130]. Based on the review of some psychoacoustics literature such as [145], [105], [82], [28], [74], [143], [3], [139], we have selected some basic psychoacoustics quantities such as frequency analysis and masking properties, perception of loudness and perceptual entropy. Some of them are also commonly used in the perceptual feature extraction techniques discussed in the next chapter. Thus we have developed a special feature extraction technique and it is discussed in chapter 13.

For perceptual adaptation we incorporated the auditory filter-bank to perceive frequencies along the critical bands. This is defined later in the chapter. The behaviors of the basilar membrane in the inner ear of the auditory system is similar to the overlapping passbands of a bank of bandpass filters. This is called an auditory filter. This influences the adaptation of some fundamental properties such as frequency masking or the scaling of the loudness of the human auditory system. An adaptation to the critical bandwidth is related to the bandwidth of an auditory filter which is incorporated in masking, loudness, absolute threshold, and phase sensitivity [12]. The processing time of the samples of the signal is notified most often by the loudness. This is one of the basic information we at first perceive. The human ear perceives sounds following its temporal pressure variations [113]. The characteristics of the sound perception process of the human ear and brain are non-linear. The response of the human brain and ears can be quadratic or cubic or quantic. For example, two loud pure tones at corresponding frequencies $f_1$ and $f_2$ are simultaneously sounded together to generate a third difference tone $|f_2 - f_1|$ to be heard.

The difference between a standard sub-band analysis and a critical sub-band analysis is that the standard sub-band is of equal width and the width of the standard sub-band does not reflect the human auditory behavior where in the critical band analysis, it works according to the function of frequency to approximate the human auditory behavior. The critical band based sub-band uses some scales to follow the distance of the basilar membrane in the cochlea in the inner ear. The critical band analysis is mapped to the critical band frequency scale such as Mel, Bark or Erb scales that we discuss in section 11.6.

In order to model the human auditory system, the perceptual quantities such as absolute threshold of hearing (ATH), sound pressure level (SPL), sensation level (SL), masking frequency, temporal masking, the mapping of the non-linear frequency scale are considered. In the frequency analysis the signal is transformed to a non-uniform logarithmic scale following some special frequency scale such as Bark, Mel or ERB. The mapping process and the transformed non-uniform new scale is called a critical band.

The sound pressure is measured in Pascal. In psychoacoustics, the values of sound pressure lies between $10^{-5}$ Pascals to $10^2$ Pascals. It is measured by a hearing threshold given by SPL. Then again in the perception stage, there is a sensation level (SL) which indicates an intensity level of an acoustic events to be heard by a listener. The SL may be used sometimes to determine which sound to be heard regardless of the loudness of the sound. It is not the same as ATH. The SPL and ATH are defined next.

### 11.1.1 Sound Pressure level (SPL)

The ratio between the reference sound pressure in Pascals and the threshold of hearing in Pascals is the sound pressure level (SPL) [12], [145]. The SPL is measured in dB. The absolute threshold of hearing is estimated as $p_0 = 2 \times 10^{-5}$ Pascals which is about $20\mu Pa$.

The intensity of the sound pressure in decibels relative to a given reference level is computed by equation (11.1). In this equation, $L_{spl}$ is the sound pressure level, $p$ is sound pressure of an event in pascal.

$$L_{spl} = 20 \log_{10} \frac{p}{p_0} \tag{11.1}$$

The relation between dB and Pascal is shown below in equation (11.2) and dB to Pascal transformation is shown in equation (11.3) where $Pa$ stands for Pascal. dBSPL means SPL is measured in dB.

$$L_p(\text{dBSPL}) = 20 \log_{10} \frac{p}{p_0} \tag{11.2}$$

$$p(Pa) = p_0^{\frac{L_p(\text{dBSPL})}{20}} \tag{11.3}$$

The SPL is also measured with respect to sound intensity level(SIL). This is equal to the sound power level(PWL) i.e. $PWL = 10 \log_{10} \frac{P}{P_0}$ and sound intensity level(IL) i.e. $IL = 10 \log_{10} \frac{I}{I_0}$ [145]. The PWL is used to measure the perception of loud mixed tonal sound.

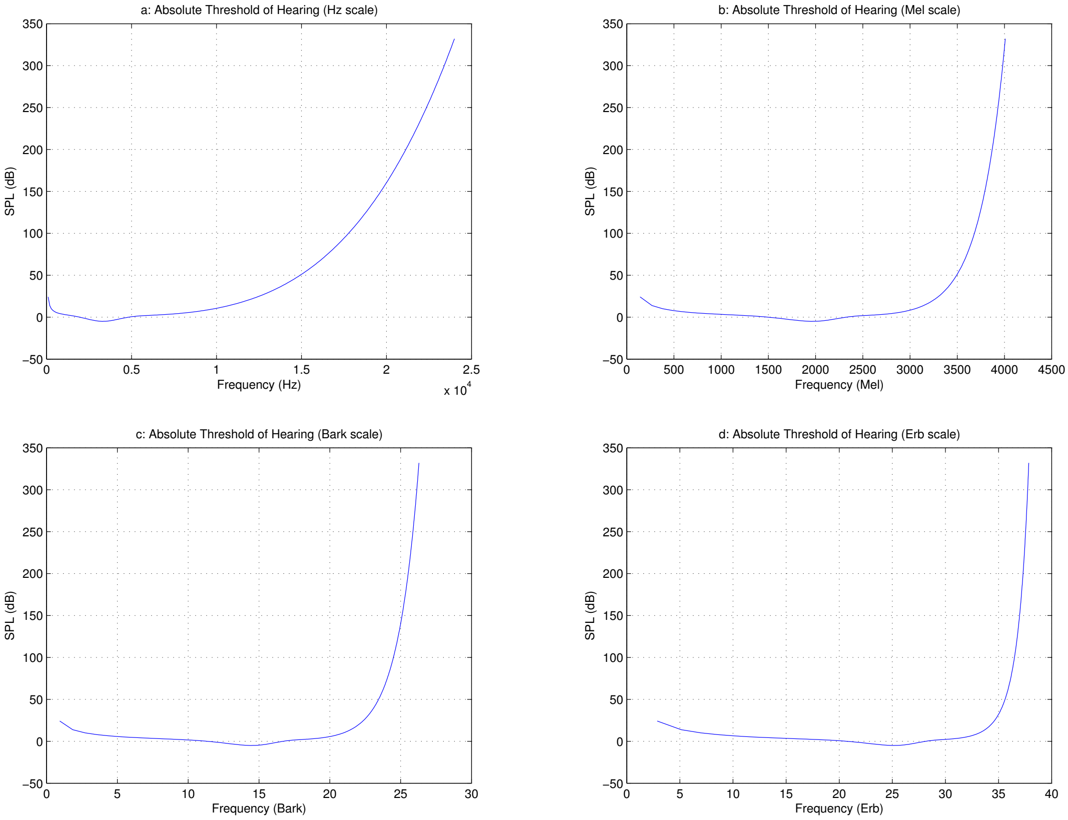


Figure 11.1: ATH in linear frequency scale in Hz, Bark, mel, and erb Scale

### 11.1.2 Absolute Threshold of Hearing (ATH)

The threshold of hearing is a listener's ability to recognize a sound in a noise free environment. It is expressed by a sound pressure level(SPL) and computed by equation (11.4) where $f$ is the frequency in hertz and $T_q(f)$ is expressed in dB, see [105]. This is a standard formula used in psychoacoustics studies. We have used the ATH in chapter 13 to compute the perceptual entropy (PE).

$$T_q(f) = 3.64(\frac{f}{1000})^{-0.8} - 6.5e^{-0.6(\frac{f}{1000}-3.3)^2} + 10^{-3}(\frac{f}{1000})^4 \tag{11.4}$$

In figure 11.1, we see the threshold of hearing is measured in SPL in dB in a quiet environment as a function of frequency. The threshold is measured in SPL in dB as a frequency of the linear frequency Hz in a, in b we see the threshold as a function of Mel frequency scale, in c we see that it is measured a function of Bark frequency scale and in d we see the threshold measured as a function of Erb frequency sale. In figure 11.1, the frequency $f$ is replaced with the Bark, Mel and Erb frequency scale in equation (11.4). For this implementation we mainly used the matlab toolbox given in [85]. In figure 11.1, the ATH measured by perceptual scales namely the Bark, Mel and Erb in b,c and d have the similar looking but the ATH shown in a some what different than the one shown in b,c and d. From this figure, we can imagine that the standard frequency that is measured in Hertz (Hz) may misinterpret the spectral analysis. The scales are defined in section 11.6. Therefore it is important for our purpose that the features are perceptually transformed using the perceptual scale in the auditory filter.

## 11.2 Concepts of Perceptual Adaptation

The purpose of a perceptional model is to hear, interprete and understand the sounds of spoken language. We concentrate on the first and partially on the second issue. The speech sound signal contains a number of acoustic elements that are used in the speech perception. These representations can then be combined to be used in the word recognition and other language processing activities. This is done particularly in the cochlea and in the basilar membrane.

The perceptual adaptation is managed by adopting several perceptual quantities such as the critical bandwidth transformation, the intensity-loudness power law transformation which is also the hearing law.

Next we introduce the hearing process of the human ear that we experience in our daily life in order to recognize speech.

## 11.3 Auditory System and Hearing Model

The sound enters into the human ears as a pressure wave and the human ears perceive the sound by its vibration. The human ears are also known as an auditory system because this acts as a sensor for the human hearing. For this, the human ears are the principle organ. Here first we explain how the human ear interprets sounds for its perception. Then we introduce how the human ear is used in the literature in order to model it.

Below this is illustrated by two figures. Figure 11.3 shows the human ear which has three main parts. Figure 11.4 shows how it is modeled to capture its perception processing.

### 11.3.1 Human Auditory System

Here we first outline the anatomy of the human ears, then we describe how these organs are used to perceive human speech. The human auditory system is explained here in order to understand the human speech perception process and to adopt some essential perceptional quantities to extract perceptional features for our DANSR. If we look at figure 11.3 that is collected from [65], we see that the human ear or the auditory system consists of major three components. These are introduced below and for this we have followed the description of the auditory system given by [13]. Figure 11.2 is a simple explanation of the components of the human ears and the interactions of the components of human ears in order to perceive speech.

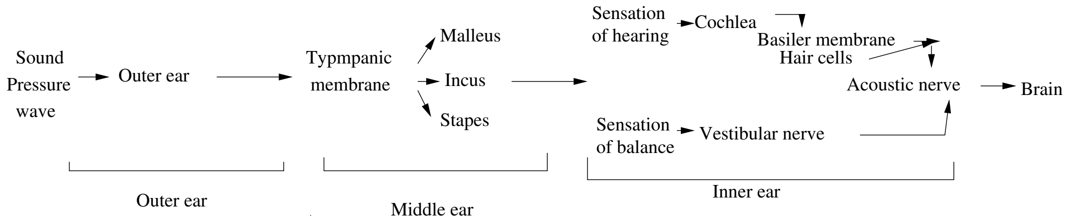


Figure 11.2: Simple View: Human ear and the interactions among the components

- Outer ear: This is connected to the middle ear through an auditory or ear channel via the tympanic membrane to the way to middle ear. The channel has many small glands so that the canal is lubricated from the secretion of the glands and the area is protected. The outer ear is also called pina.
- Middle ear: Three small parts, malleus, incus, stapes on the back side of the tympanic membrane are the components of the middle ear. These three bones can vibrate. The middle ear is again connected to the throat and nasal system through the eustachian tube. There is a nerve connected in the malleus and incus to the tongue. The middle ear is connected to the inner ear through stapes.
- Inner ear: This is a cavity which has bones inside it. The inner ear has two regions : i) Sensation of hearing which is the snail shell shaped cochlea and ii) Sensation of balance which is semicircular shaped vestibular nerve. The vestibular nerve is associated with many ducts and the middle ear is

connected to the inner ear through stapes to vestibular. The ducts belong to the vestibular nerve and has many fluid filled passage ways. The cochlea is connected to the brain with different 12 pairs of auditory or acoustic nerves. There are many fluid filled cavities in the vestibule as well as in the cochlea. This is the place where frequency analysis is taking place.

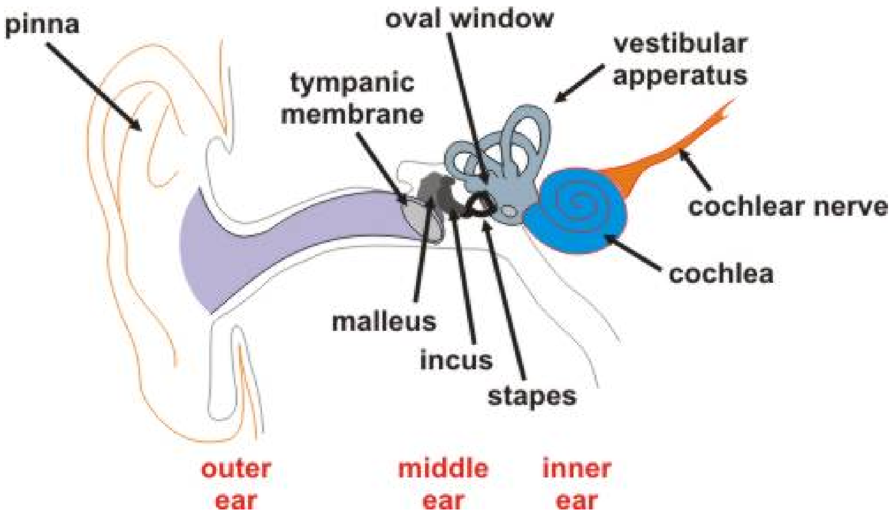


Figure 11.3: Human Auditory System [65]

Next we have introduced how we have used these organs to perceive sounds for its recognition.

### 11.3.2 Human Hearing Process

The sound wave travels from the outer ear through the ear channel to the tympanic membrane. The vibrations are transmitted to the hair cells connected to a fluid filled passage in the inner ear. The vibrations generate signals which are carried out to the brain for the sound interpretation through the auditory or acoustic nerve. There are about 16000 to 20000 hair cells along the length of the cochlea in 4 different rows. There is only one row in which the inner hair cells are attached to nerves, and the rest third rows are outer hair cells. These hair cells in the cochlea play a significant role in the properties of the sound i.e. pitch, loudness and how these properties stimulate the hair cells and send a signal to the brain.

The idea of the fluid filled cavities is that a movement of the bones caused a vibration wave in the fluid which stimulates the microscopic hair cells connected to the nerve. Moving back and forth, the hair cells connected to the cavities in the cochlea fire electrical signals or impulses that are carried out to the brain through the auditory or acoustic nerves to the brain as an interpretation of the sound. The vibration or firing of the hair cells at different rate helps the brain to interpret the sound frequency one from the other.

The human ear is subjective to its response to different frequencies. This characteristics is technically achieved by using the different types of scales namely the Bark scale, Mel scale and the Erb scale. The peripheral auditory system acts as a frequency analyzer. The basilar membrane of the inner ear plays a significant role to locate and characterize the frequency. An auditory filter can measure the neural tuning curve and neural impulse responses. The auditory filter is mapped on the critical band to represent the frequency resolution of the auditory system. The critical bandwidth is measured by comparing the masking and loudness [142].

### 11.3.3 Hearing Model

Now if we compare the human auditory system given in figure 11.3 with figure 11.4, we can see its approximation in figure 11.4. In this figure, we see in the hearing model the capture of the sound wave, its entrance to the inner ears through the outer and middle ear are estimated by the spectral shaping (this is explained chapter 12). In the same figure the frequencies analysis of the sound wave and their distinction by the different parts of the auditory organ shown in figure 11.3 as well as transportation of these to the brain through the auditory nerve are estimated in figure 11.4 by spectral analysis and their parametric representations (discussed in chapter 12). In the human hearing model approximated by the auditory filter bank, a sound of a definite frequency does not actually vibrate the membranes in the cochlea at one point. Rather, there is one point where the vibration is biggest, and a range around that value where we have that the vibration is big enough to be important. This range on the cochlea is called the critical band excited by a sound. It covers frequencies about 10% to 15% higher and lower than the tone which is played. The flexing which is the bending of the cochlea falls away as the tone moves from the center of the band. Intensity is the power per area and the power is the rate at which the energy is distributed. The pitch is determined by a periodic sound's frequency of one period of the wave [37]. The resolution of a finite length signal is the minimum number of samples

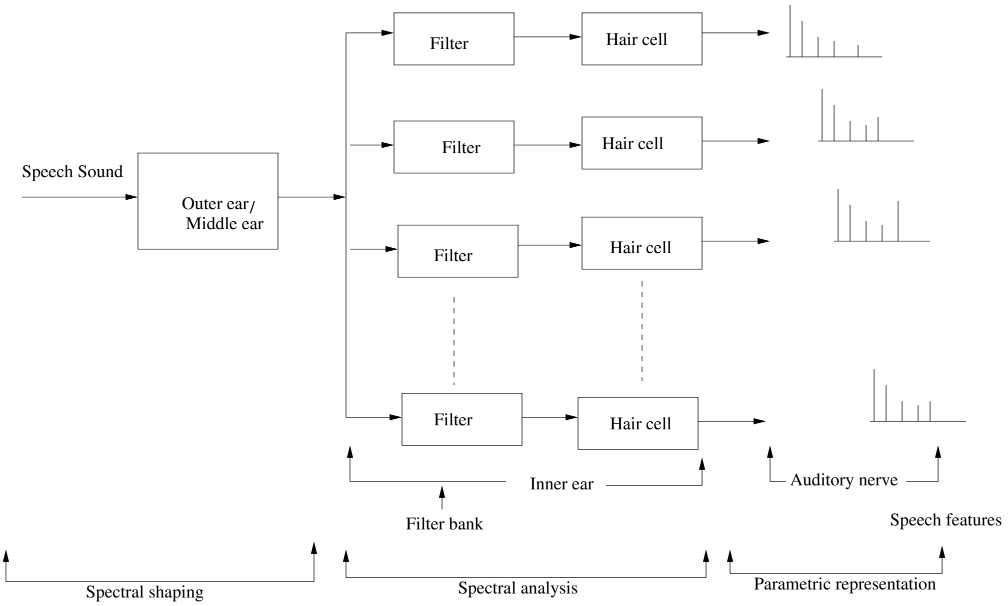


Figure 11.4: Approximated Human Auditory Filter bank [12]

that is required to represent it. Thus the resolution can be seen as information container of the signal. This is approximated in the auditory filter-bank via each filter used in the filter-bank. For these special perceptual frequency scale is used for the spectral analysis discussed in section 11.5.

## 11.4 Auditory Masking and Masking Frequency

Auditory masking is a psychoacoustic effect that determines the mapping of the frequency in the critical band. Frequency masking makes one sound inaudible due to a presence of another sound. This gives the threshold of the audibility at where one sound is raised by the presence of another sound [76]. The frequency of the later sound may be higher [147]. The inaudible frequency of the sound is called a masked frequency and the frequency of the sound which presence makes masked frequency is called masker frequency. Two common masking types are i) Frequency masking or simultaneous masking excites multiple tones at the same time and ii) Temporal masking excites a particular frequency zone in the cochlea along the basilar membrane. Both types of maskings are carried over to the human brain by the auditory nerve [145]. The auditory masking is related to the SPL and the sensation level (SL) which is an intensity level of an acoustic event to be heard by a listener. The masking effect is the same when the power of the tone and the power of noise spectrum is near that tone. However, the masking effect outside this area of the tone does not interfere to that described area. Here, if the characteristic frequency band has the same acoustic power for the tone and the noise spectrum within that band, the tone is masked and this is the concept of the critical bands defined in [41]. Further to this explanation, an assumption is that the human hearing system processes sounds in relatively narrow frequency bands. The hearing system produces masked threshold frequencies independent of the frequency. The unmasked threshold is the quietest level of the signal which can be perceived without masking the signal. In some literature, the total masking threshold is approximated by a summation of the threshold produced by an individual signal components following the power law.

The human sound perception i.e. the speech hearing is affected by masking properties. The intensity the acoustic stimulation is measured by the standard sound pressure level (SPL). The loudness remains constant for a narrow band noise source at a constant SPL even as the noise bandwidth is increased up to the critical bandwidth tends to remain constant about 100 Hz to 500 Hz and increases approximately 20% of the center frequency above 500 Hz. The width of the critical band is commonly referred to as one Bark scale which is a non-linear function. It is often used to convert the frequency from the Hertz to the Bark scale.

## 11.5 Frequency Analysis and Critical Bands

Here we refer to the parts of the ear described in figure 11.2 and describe its functioning.The critical bands are some sub-divisions of a frequency domain. The sub-divisions are some non-uniform sub-bands. These are used to understand the frequency analysis of the human auditory system. The critical band introduced by the scientist Harvey Fletcher in the 1940s, is the frequency bandwidth of the ”auditory filter” created by the cochlea which is a sense organ of hearing within the inner ear. The critical bank is roughly the band of audio frequencies within which a second tone will interfere with the perception of a first tone by auditory masking [12]. The critical band denotes a constant distance on the cochlea and the bandwidth where the signal intensities are added to decide whether the combined signal exceeds a masked threshold [139]. The critical bands are continuous and the audible frequency in each of the band has a tone in its centered position.

The critical band relates to the perception properties such as loudness, pitch, and tone. The auditory system performs a frequency analysis of sounds into their component frequencies. The cochlea acts as a spectrum analyzer of the sounds in the inner ear. The high frequency bands are wider than the low frequency bands. A summation of a collection of the critical band responds is assumed to be the loudness [61]. The human hearing system processes sounds in narrow frequency bands. It produces a masked thresholds frequency below 500 Hz and the range of the masked threshold is independent of the frequency. The critical bandwidth has a constant width below 500 Hz but it increases 10 dB per decade for the frequency bands. The bandwidth of the bands increases by a factor of 10 as the frequency increases by the same factor [41]. The critical bandwidth shows a constant bandwidth at about 100 Hz, but the range of the critical bandwidth increases proportional to the frequency above the frequencies of 500 Hz and the increase rate is 20% of the center frequency.

The masking threshold is measured on the critical-band. Therefore the power spectrum of the signal is partitioned into critical bands. The architecture of the cochlear filter pass band is non-uniform which changes as a function of the frequency non-linearly. There are many different formulations that have been developed to calculate the critical bandwidth. One approach is to calculate the critical bandwidth as shown in equation (11.5) where $f$ is the linear frequency in Hz and $BW_c(f)$ is the frequency of the critical band in $f$. This critical is later transformed into perceptual spectral band when it is multiplied by a perceptional feature scale such as the Mel, the Bark or the Erb scale.

$$BW_c(f) = 25 + 75(1 + 1.4(f/1000)^2)^{0.69} \tag{11.5}$$

The psychoacoustic model needs a delay in the time-frequency decomposition in order to center the data in the audio frame within the psychoacoustic analysis window. Critical bands and their band width distribution are shown in the next table.

### 11.5.1 Perception of Loudness

The loudness is a physiological aspect and it considers the intensity of the sound. Its sensation to an environment for a particular subject is difficult to measure. The loudness is measured in sone or phone and the unit of the loudness level is phone. The loudness is subjective and environmental. The definition of the loudness in some literature paraphrased here is an intensive attribute of the auditory sensation where the sound can be distinguished as loud and soft for being pro-

Table 11.1: Critical bands and their band width distribution [143]

| Band No | Lower Band(Hz) | Center(Hz) | Upper Band(Hz) | Band Width(Hz) |
|---:|---:|---:|---:|---:|
| 1 | 20 | 50 | 100 | |
| 2 | 100 | 150 | 200 | 100 |
| 3 | 200 | 250 | 300 | 100 |
| 4 | 300 | 350 | 400 | 100 |
| 5 | 400 | 450 | 510 | 110 |
| 6 | 510 | 570 | 630 | 120 |
| 7 | 630 | 700 | 770 | 140 |
| 8 | 770 | 840 | 920 | 150 |
| 9 | 920 | 1000 | 1080 | 160 |
| 10 | 1080 | 1170 | 1270 | 190 |
| 11 | 1270 | 1370 | 1480 | 210 |
| 12 | 1480 | 1600 | 1720 | 240 |
| 13 | 1720 | 1850 | 2000 | 280 |
| 14 | 2000 | 2150 | 2320 | 320 |
| 15 | 2320 | 2500 | 2700 | 380 |
| 16 | 2700 | 2900 | 3150 | 450 |
| 17 | 3150 | 3400 | 3700 | 550 |
| 18 | 3700 | 4000 | 4400 | 700 |
| 19 | 4400 | 4800 | 5300 | 900 |
| 20 | 5300 | 5800 | 6400 | 1100 |
| 21 | 6400 | 7000 | 7700 | 1300 |
| 22 | 7700 | 8500 | 9500 | 1800 |
| 23 | 9500 | 10500 | 12000 | 2500 |
| 24 | 12000 | 13500 | 15500 | 3500 |
| 25 | 15500 | 18775 | 22050 | 6550 |

cessed and recognized by the human hear and the brain. The perceived loudness is not proportional to the intensity of the sound. It is more nearly proportional to the logarithm of the intensity. This is what makes decibels such a useful measure. The relationship between the loudness of a sound and a perceived loudness of the human ear is captured by an estimation of the magnitude of the loudness and the estimation of the production of the loudness of the sound [12].

Hearing of a sound has several effects. The environmental phenomena affect the sound of hearing. For instance, if the environment is quiet or noisy, then the perceived sound can be different but still there is a threshold at which level the subject can perceive the sound regardless of an environmental effect. According to my knowledge the environmental impact is not getting here a priority so far in the literature on the perceived sound research.

About 150 dB SPL spans the dynamic range of the auditory system; an SPL reference of a quiet environment is around 0 dB SPL while a stimulus of 140 dB SPL approaches the threshold of a pain.

Here the human auditory system can select from loud or weak or soft sound components. Without this ability all sounds would be the same. The loudness in this sense may be a part of the frequency warping. Thus we realize the importance of loudness and distinguish the sound according to this. Based on [129], [142], the loudness is denoted as a power function of physical sound intensity.

Equation (11.6) is a non-linear equation where the loudness of the sound is proportional to the intensity raised to the power of 0.3 [12]. In equation (11.6), $L$ is the perception of the magnitudes of the loudness, $k$ is an arbitrary constant determining the frequency scale unit, $I$ is the stimulus intensity, $a$ is a power exponent which is generally 0.3.

$$L = kI^a = kI^{0.3} \tag{11.6}$$

**Equal Loudness Curves** The human ear can distinct the frequencies well and this is of interest for a machine too. The audibility of the human is variable according to the frequency. The same SPL at different frequencies may not be perceived as equally loud. This distinction is captured by a set of loudness curve that compares SPL (sound pressure in db) with the phone scale. In the phone scale, a sound with equal phone values is perceived equally loud at any frequency. The value of SPL and loudness level is equal at $f = 1$KHz on the equal loudness curves. This has a lower sensitivity at lower frequencies. These aspects are sometimes termed as equal loudness pre-emphasis [42].

## 11.6 Analysis: Perceptual Scales

One purpose of psychoacoustic scales is to provide steps that correspond to equal perceptual intervals [35]. In different contexts the same numerical distance can be perceived in different ways what are reflected in the scales. We will discuss the Mel, Bark and Erb scale in this section. These are measured in logarithmic scales. These scales give different intervals for the linear frequencies. These scales influence the way in which the speech is recognized and therefore present information for the listener.

### 11.6.1 Mel Scale

The Mel scale (measured in Mel) is a non-linear scaling that is used to perceive frequency characteristics of the human ear. In the Mel scale, the frequency ranges are divided into four equal intervals. The frequency is adjusted there in such a way that one half of this frequency scale is equivalent to a given linear frequency. The pitch is a psychoacoustic variable is characterized by the frequency, loudness, intensity or amplitude of the acoustic sound. The Mel scale shows a good performance while discriminating the speech segments. In the mel scaled band, the mel scale is used to represent the frequency in the critical band. One mel is defined as one thousands of the pitch of a 1 kHz tone. The filter bank is a set of triangular filter banks based on critical band scales at frequencies i.e organized in the 2nd column in table 11.1. The frequency $f$ is in Hertz. The spacing of the critical bands is non-linear. The Mel scale is first used in ASR system for perceptual speech feature extraction by [110].

Next we discuss the relation between Mel scale and linear frequency scale measured by Hz. There are several different formulations of the Mel scale. Each of them is used differently in the literature. Below we present a definition in equation (11.7) that we see frequently used. In equation (11.7), $f$ is in Hz linear scale and $f_{mel}(f)$ is in the Mel scale. According to [78], the cochlear position $x$ from 0 to the frequency is $f$ in Hz where $f = 165.4 \times (10^{2.1x} - 1)$ and thus the scaling at $f = 1000$ gives in the Mel scale at $m = 512.18 \times \ln(\frac{f}{165.4} + 1)$. Here the break frequency is 165.4 Hz that separates the log-like high-frequency region from the linear-like low-frequency region.

$$f_{mel}(f) = 2595 \log_{10}(2 + \frac{f}{700}) \tag{11.7}$$

The corresponding inverse expression from mel scale to the linear frequency scale Hz is shown in equation (11.8).

$$f = 700(10^{(f^{-1}(m)/2595)} - 1) \tag{11.8}$$

### 11.6.2 Bark Scale

In the Bark frequency scale an equal or uniform distance represents perceptually equal distances. The Bark scale is linear below 500 Hz and non-linear above 500 Hz [15]. This non-linear spectral distance is measured using the logarithmic frequency axis. There are several definitions of this concept. As mentioned in section 11.5 they all give the same information about the relation between the Hz and the Bark scale [25] , [56].

We state the formula that we have followed for the Bark scale and its inverse transformation. We followed in this regard Bark scale's recent formula. Among the different Bark scale formulations we use the formula shown in equation (11.9).

$$B(f) = 6\sinh^{-1}(\frac{f}{600}) \tag{11.9}$$

The Bark scale to hertz scale is converted using equation (11.10).

$$\text{hz} = 600 \times \sinh\frac{B(f)}{6} \tag{11.10}$$

### 11.6.3 Erb Scale

The other commonly used perceptual frequency scale is the Erb scale. The Erb scale is formulated in equation (11.11) where $f$ is the center-frequency in Hz, normally in the range 100 Hz to 10kHz. The Erb scale is generally narrower than the classical critical bandwidth (CB) such as Bark or Mel scale and $f$ is in Hz [67].

$$\text{Erb}(f) = 21.4\log_{10}(0.00437f + 1) \tag{11.11}$$

The Erb warping is determined by scaling the inverse of eq (11.11), evaluated along a uniform frequency rangning from zero to the number of Erbs at half of the sampling rate, so that direct current (DC) maps to zero and half the sampling rate maps to $\pi$ [67].

### 11.6.4 Comparison

We see in figure 11.5, the Mel scale in a, the Bark scale in c and the Erb scale in e. The basilar membrane which is situated in the cochlea is thin close to the stapes but wider at the end. The thin section of this membrane responds to high frequency and the wider section responds to low frequency. Following this anatomical structure of the basilar membrane, the critical band is closely spaced at the low frequency but widely spaced close to high frequency in figure 11.5. In this figure, the critical band is mapped to the Mel scale in b, the Bark scale in d and Erb scale in e. Each of these cases, we see the critical band is densely spaced at low frequency and sparsely spaced at high frequency.

In figure 11.6, we see the comparison among the Bark, Erb and Mel perceptual scales. The curves indicating the scales are generated using the equations (11.9), (11.11) and (11.7). They all show the similar behaviors against linear scale Hz. They are expanded at low frequencies below 1000 Hz and above this frequency they are all compressed. For this implementation we mainly relied on the matlab toolbox given in [85].

## 11.7 Analysis: Auditory Filter-bank

To capture the functions of the human auditory system and its hearing, a bank of filters is arranged in such a way that the passbands of the filer-bank are overlapped. This is used to model the human auditory system, it is called the auditory filter bank. The filter shape can be different such as triangular, trapezoidal. The filter bank can be uniform or non-uniform. A set of transfer functions $H_m(z)$ in the analysis filter bank splits the input into $M$ subband signals in the synthesis filterbank. The shape of the filter bank can be triangular or trapezoidal.

For the speech feature extraction usually non-uniform spaced filter-bank is used. In such a case, the part of the spectrum below 1kHz is processed by more filters in the bank because it is assumed that the 1st formant lies in the lower frequency range and there exists more vocal tract information.

The frequency resolution of the auditory filter bank largely determines which portions of a signal are perceptually irrelevant. The auditory time-frequency analysis that occurs in the critical band filter bank induces simultaneous and non-simultaneous masking phenomena that are assumed to be the shape of the distortion spectrum. A perceptual model exploits the masking thresholds for a complex sound. The loudness scale is related to the sound level depending on the

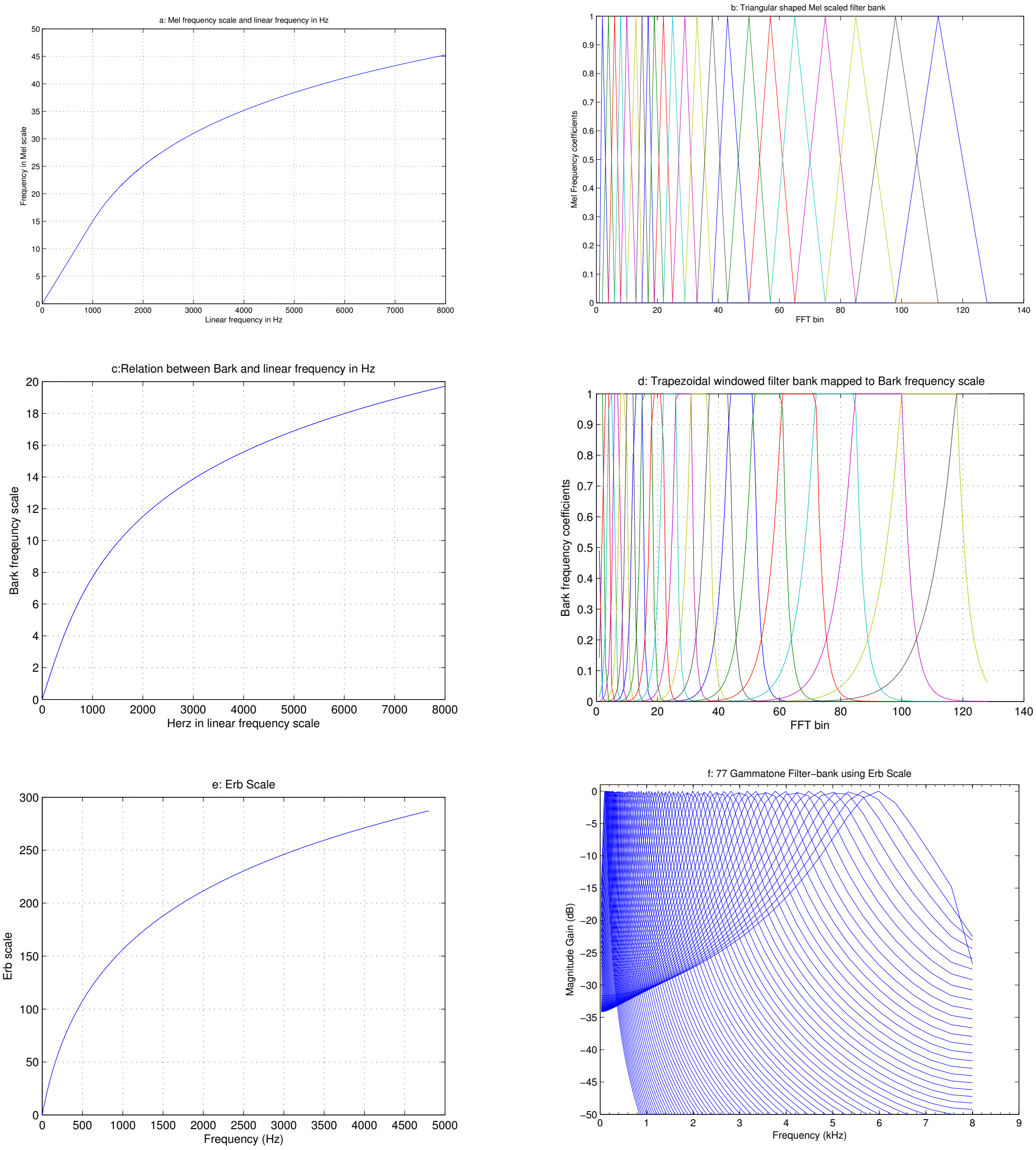


Figure 11.5: Critical band in trigonometric, trapezoidal and rectangular filter-bank mapped to Mel, Bark, and Erb frequency scale

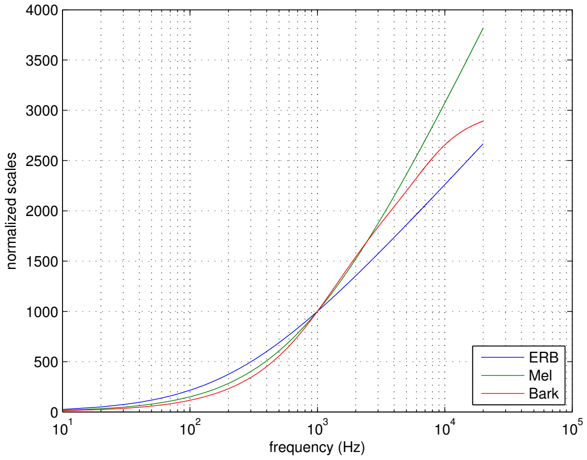


Figure 11.6: Perceptual Scales: Erb, Bark, and Mel frequency scales and linear frequency scale in Herz

duration and frequency of the sound. It is variable with respect to the perceived sound level.

In the following discussion of the filterbanks we refer to the scales introduced above.

### 11.7.1 Mel Filterbank

In Mel frequency wrapping, the signal power is the input to the bank of filters which has a bandwidth of the triangular band pass filters and gives the frequency resolution at different frequency bands. The bandwidth of the triangular band pass filter is positioned in the psychoacoustic frequency scale in particular in the Mel scale. This is done by integrating the area of the bandwidth over this frequency scale. The filters are overlapped in the filter bank such that the lower boundary of one filter is at the center frequency of the previous filter and the upper boundary of the filter is the center frequency of the next filter. The maximum response of the filter is the top vertex of the triangular filter which is the center frequency and is normalized to unity.

The frequency is given by Mel scale, then the corresponding frequency is

$$f_f^{-1}(mel) = [700e^{m/1127} - 700]\text{Hz}$$

Suppose the number of filters is $M$ and $m = 1, 2, 3, \cdots, M$, where $m$ denotes the triangular filter given by equation (11.12) and $f$ is the frequency in each bin which size is equal to the FFT size. Now the filter-bank $H_m(k)$ is given in equation (11.12). $f(m-1)$, $f(m)$, $f(m+1)$ are the left, middle and right boundary of the $m^{th}$ filter. $H_m(k)$ is the weight of energy at frequency $k$ for $m^{th}$ filter.

$$H_m(k) = \begin{cases} 0 \text{ for } k < f(m-1) \\ \frac{2(k-f(m-1))}{f(m+1)-f(m-1))f(m)-f(m-1))} \text{ for } f(m-1) \leq k \leq f(m) \\ \frac{2(f(m+1)-k)}{(f(m+1)-f(m-1))(f(m+1)-f(m))} \text{ for } f(m) \leq k \leq f(m+1) \\ 0 \text{ for } k > f(m+1) \end{cases} \tag{11.12}$$

Here $f_l$ and $f_h$ are the lowest and highest frequencies of the filter-bank in Hz, and $f_s$ is the sampling frequency in Hz, $M$ is the number of filters, and $N$ is the size of the FFT. Then the boundary points $B$ are non-uniformly spaced in the mel-scale. One bin has the same length as the size of the FFT.

$$f(m) = \frac{N}{f_s} B^{-1}\big(B(f_l) + m\frac{B(f_h) - B(f_l)}{M+1}\big) \tag{11.13}$$

In equation (11.13), $B$ is the acoustic scale which can be either Mel, or Bark or Erb scale. We then compute the log-energy for $m = 0, 1, \cdots, M$.

### 11.7.2 Bark Critical-band

The relation between the Bark scale and the linear frequency scale is shown in equation (11.14). There $f$ is the frequency in Hz and $f_b$ is the related Bark frequency.

$$f_b = 6 \ln\big(\frac{f}{600} + \big(\{\frac{f}{600}\}^2 + 1\big)^{0.5}\big) \tag{11.14}$$

In the critical band analysis shown in equation (11.15), the center frequencies of the filters in the filter bank is spaced in the Bark scale and the distance between the center frequencies of the filters in the critical band filter-bank is 1 Bark scale. The first filter is placed at zero frequency and the last filter is placed at Nyquist frequency. The lowest and highest frequencies in the filter bank are 0 and the Nyquist frequency. This spacing is similar to the Mel scaled filter bank. In the Bark scaled filter-bank given in equaation (11.15) $f_b$ is the bark frequency and

$f_{cb}$ is the center frequency of the filter in Bark scale [100].

$$\Phi = \begin{cases} 0, & \text{if } (f_b - f_{cb}) < -2.5 \\ 10^{(f_b - f_{cb} + 0.5)}, & \text{if } -2.5 \leq (f_b - f_{cb}) \leq -0.5 \\ 1 & \text{if } -0.5 \leq (f_b - f_{cb}) \leq 0.5 \\ 10^{-2.5((f_b - f_{cb}) - 0.5)}, & \text{if } -0.5 \leq (f_b - f_{cb}) \leq 1.3 \\ 0, & \text{if } (f_b - f_{cb}) > 1.3 \end{cases} \tag{11.15}$$

Again, we took over the equation from the literature [?].

## 11.8 Perceptual Adaptation in DANSR

In summary, the main adapted psychoacoustic quantities in this research are:

- Bark scaled Critical Band
- Pre-emphasis by Loudness Scale
- Perceptual Entropy

For the extension we have applied the similar formulation as given in section 11.7.2. Pre-emphasis by the loudness scale (12.20) is discussed in chapter 12 and the perceptual entropy is discussed in section 13.3.1 in chapter 13. The purpose of the perceptual adaptation is as mentioned before to keep the relevant perceptually meaningful information by removing the redundancy. In this line, the perception of human auditory system is attempted to mimic by adapting some basic perceptual quantities listed above in this section.

## 11.9 Psycho-acoustical Analysis of MP3

Here we would like to mention that the MP3 was developed by Karlheinz Brandenburg and his group and some explanation that can be found in [90] used the spectral analysis and the compression done by the human ear. A reality is that what the sound pressure waveform we perceive has redundancy and our ears analyze the frequency of the perceived the waveform and process this. Only the essential parts of this is that we need to perceive the semantic of the wave and the rest is done by the brain. The purpose of the feature extraction discussed in chapter 12 is to reduce the redundancy of the captured waveform and used this for recognition. The aim for the MP3 development by Brandenburg and his group is to closely achieve the audio and psychoacoustics properties in order to compress the real world signals. But we have not investigated these aspects of the audio and psychoacoustics properties in details. The perception of human hearing and the role of human ear is still an ongoing research and it is not yet clear how human ears analyse the captured waveform.

# Chapter 12

# Standard Features and Feature Extraction Techniques

**Outline of the chapter** We discuss the feature extraction techniques in our context. We start with a brief overview of the standard feature extraction techniques, their effectivity and limitations. This includes in particular feature types such as cepstrum, MFCC, LPC, LPCC, PLP, RASTA and SILTT. In case where different notations and conceptual variations exist we make precise what we use. This will also be used to take advantage for computer implementations when studying the human ear. On these foundations we present our extraction process in the next chapter. This is an essential part of our system. Although several of the details are somehow known, the selection and integration of the individual methods into DANSR are innovative.

## 12.1 Fundamentals: Feature Extraction

In the speech processing, the speech waveform is recorded by using some sensors or transducers. They are not used directly for recognition purposes, they are rather transformed into some lower dimensional vectors. These are the feature vectors. These represent the input speech i.e. some acoustic phonetic information. From this point on the information of the features can be analyzed on an acoustic, a phonetic, or a linguistic, word or the language level. Distinctly the role of a feature extraction technique is vital in order to perform the transformation of the speech signals into feature vectors, then to authenticate the information of these feature vectors in the classification and recognition stage.

The feature extraction has the following aspects that we will discuss next:

- Feature extraction: This is a description of the approach that is used to extract the features from the data collection.
- Feature subset generation: This describes how the features are distributed.
- Definition of the evaluation criteria: The criteria of the feature selection i.e. entropy or energy or spectral envelope.
- Assessment of the evaluation criteria: This is to confirm the validity of the evaluations in the classification and in the recognition stage.

## 12.2 Features and their Purpose

One of the main goals of the signal analysis used in the feature extraction is to reveal information they contain. This type of signal analysis is also in the context of the machine learning. In this context the signal analysis is mainly concerned about the information of the signal and how that can be represented efficiently in a compact form so that one can represent the original input on the machine. This type of analysis transforms a large dimensional signal into some small finite vectors of relatively smaller length than the original signal. These finite vectors hold the essential signal information and can now be used as a representation of the original signal in a compact manner. How the features are extracted and what are they, can be intuited in figure 12.1. There discrete speech signal $s$ is windowed by a window function, then processed spectrally. then these are used for feature extraction. The features are generated in vector. In figure 12.1, $\mathbf{o}_1, \mathbf{o}_2, \cdots$, are feature vectors and features are the elements in the each feature vector. These will be detailed in this chapter.

Often, features that are not individually relevant may become relevant in certain contexts. One approach to feature selection is to use the rank of the features according to their individual relevance. An example of this is the MFCC feature extraction technique. In this technique, in order to capture the dynamic behavior, there exist first the feature transformation, then derivative of the features, then again second derivatives of the features. But these options are application and user dependent. The MFCC feature extraction technique is discussed in the text.

The next sections have mainly an overview character. Later we will be precise when we discuss the methods that we use.

Now some common feature extraction methods are listed:

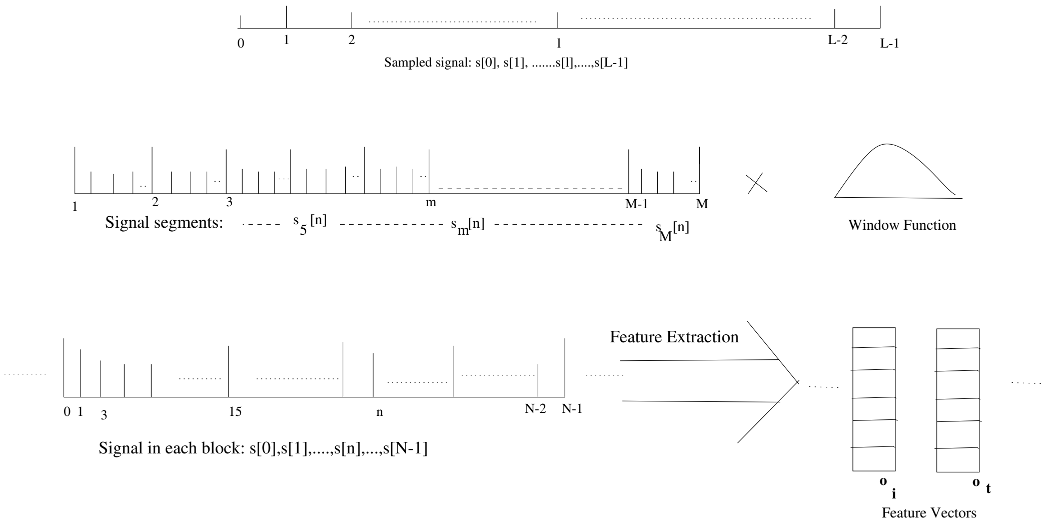


Figure 12.1: Speech Features in Picture

- Non-parametric Fourier transform based speech features extraction: This commonly uses spectral envelops of the speech in the transformation for feature extraction.

- Non-parametric wavelet or local trigonometric transformation (LTT) types of speech feature extraction: These extraction processes may be categorized as non-parametric because they do not use a model. They rather decompose the signal in a special manner and feature analysis is generally based on a discrete cosine transformation.

- Parametric Fourier Transform based speech feature extraction: This uses some parametric model such as linear prediction or a linear combination of spectral envelops in the transformation to extract features.

Some feature extraction techniques incorporate perceptual properties of human hearing and human speech production. The requirements and the representations can be different for different applications. One can not say in general what is the best feature extraction method is. But an analysis of different types of feature extraction techniques is given in section 12.12. For example, speech feature extraction in the speech recognition application distinguishes between different phonemes of a language. Such representations vary according to the type of the demand. In such cases, it raises a compromise between what to keep or delete and therefore one or the other information can be selected or ignored according to the priority of the speech recognition aspect. Given the above feature extraction methods, we used the features that use the LTT types of features using DCT typed spectral analysis. However, we extended this feature extraction technique to perceptional feature extractions importing some essential perception quantities. This is discussed in the next chapter 13.

Below we provide some different measurements that are commonly used as features in the speech recognition research.

### 12.2.1 Conventional Feature Parameters

There are different measurements and distribution properties to be used as features. Below we provide first a list of conventional feature parameters. We will see the use of the computations of these parametric feature transformations in the description of standard feature extraction techniques in sections 12.5, 12.6, 12.9.

**Frame energy** This is a measure of the energy of the short time speech signal.

**Spectral envelope** A spectral envelope is a piecewise spectral information. It is used parametrically and also non-parametrically using FFT to characterize the signal.

In the non-parametric method, generally the windowing or low pass filter based log magnitude spectrum is computed to extract the piecewise spectral information of the system. In the parametric method, most commonly the amplitude response of the all-pole filter is analyzed to obtain the piecewise spectral information of the system.

**Log energy** These are logarithmic computations of the short term energy of the speech signal.

**Delta cepstrum** These are the derivatives of the cepstrum features introduced in section 12.5. They are used to capture the dynamic behavior or the underlying information of the speech process by taking the derivatives of the primarily extracted feature parameters. We will discuss them below. Particular examples are the derivatives of the energy and the velocity and the acceleration of the features as an indication to get a realization of the time variation of the signal.

**Spectrogram** The spectrogram is a graphical representation of the energy density as a function of the frequency. Spectrograms of the speech signals often analyze the phonemes and their transitions. In a linguistics sense, phoneme can be defined as some phonetically distinct articulations.

**Entropy** Entropy characterizes the behavior of the random variables. It is quite often useful to estimate the probabilities of events to find the hypothesis of the smallest error [53].

## 12.3 Steps involved in Feature Extraction

The perceptual feature extraction has four fundamental steps: Spectral shaping, spectral analysis, perceptual representation, and parametric transformation. In a standard situation, the feature extraction of the speech starts by pre-emphasizing the signal. Then the signal is blocked into frames in the spectral shaping following the spectral analysis and the perceptual feature transformation. A common approach to the feature extraction methods is that the features are computed on the speech frames. The concepts are or have been defined in detailing the past steps.

**Spectral shaping** This involves the transformation of the analog speech waveform into discrete time signals, pre-emphasizing, pre-filtering, signal segmentation into blocks (uually in the range of 10 to 30 ms) for their spectral analysis in the next step.

**Spectral analysis** This is an analysis of the frequency information enquired in the frames by some spectral analysis methods such as Fourier transform, discrete cosine transform (DCT) or wavelet transform. This transformation can be perceptual when it uses the perception scales such as Bark scale or Mel scale.

**Perceptual feature representation** Here the aim is to approximate the perception of the human ear. This is discussed in chapter 11. The procedure is in general taking place in the human ear and is the subject of the spectral analysis in the non-uniform band pass filter bank. This type of analysis is called as perceptual spectral analysis. The analysis is done in the critical band (see chapter 11) which uses some non-uniform scales as for example Mel, Erb or Bark scale. The frequency analysis in the critical band is an auditory filtering (see chapter 11).

The perceptual feature representation uses auditory filtering that applies some psychoacoustic quantities such as masking properties in the critical band, perception of loudness scaling, equal loudness contour.

**Parametric feature transformation** This is used to extract the speech features which can be if the perceptual spectrally analyzed speech frames after the perceptual feature representation ahave been considered. These features are then used for the classification and the recognition discussed in chapter 14.

## 12.4 Analysis of Standard Feature Extraction Techniques

First we list in an overview of the most commonly used feature extraction methods that incorporate human hearing and then we list the commonly used feature extraction methods that do not have links with the human hearing or the perception process.

- Standard Perceptual Feature extraction techniques : They are concerned with the human speech perception.
  - Mel frequency cepstral coefficients(MFCC)
  - Perceptual linear predictive (PLP) cepstral coefficients
- Feature extraction techniques : They do not consider the human perception in the architecture.
  - Linear predictive coefficients (LPC)
  - Linear predictive cepstral coefficients (LPCC)
  - Cepstrum
  - Shift invariant local trigonometric transformed (SILTT) features

Some of the feature extraction methods are described in brief next. We repeat some elements for being clear.

The common processings used in MFCC and PLP are the following:

- Spectral Shaping

- Perceptual Spectral Analysis
- Perceptual Feature Representation
- Parametric Feature Transformation

The other standard feature extraction techniques such as cepstrum, LPC, LPCC do not include the perceptual feature representations. The MFCC and PLP are in fact an extended version of cepstrum and LPC techniques.

## 12.5 Cepstral Feature Extraction Technique

The cepstral features are derived mainly from cepstrum analysis. Here we come to more details. There are two types of cepstral features: i) LPC cepstrum analysis and ii) FFT cepstrum analysis. In the LPC cepstrum analysis, the speech signal $s[n]$ is first seen as an output of the convolution of the excitation $u[n]$ and the impulse response $h[n]$ as given in equation (12.1). This is just characterizing the signal in terms of the model parameters using the deconvolution approach discussed in chapter 7 and 8.

$$s[n] = u[n] \otimes h[n] \tag{12.1}$$

The equation (12.1) is represented in the frequency domain using FFT and the corresponding equation is (12.2). $S(k)$, $U(k)$ and $H(k)$ are the Fourier transforms of $s[n]$, $u[n]$ and $h[n]$. $n = 0, 1, 2, \cdots, N-1$ and $k = n = 0, 1, 2, \cdots, N-1$. Then the logarithm of this equation is taken in equation (12.3) in order to separate the source excitation and the impulse response. Then the cepstrum is computed by the homomorphic filtering i.e. taking the inverse Fourier transform of the logarithmic computation of the magnitude of the speech spectrum. The homomorphic filtering is defined by equations (12.2) and (12.3). The details of this explanation can be found in [9], [69].

$$S(k) = U(k)H(k) \tag{12.2}$$

$$\log(S(k)) = \log(U(k)) + \log(H(k)) \tag{12.3}$$

The logarithmic operation on the $S(k)$ gives equation (12.3). This is now an additive operation. The logarithmic view allows an additive operation. It is assumed that the excitation is related to high frequencies and the response of the vocal tract filter has a relatively lower frequency than the excitation source. Based on this assumption, the source is separated. The cepstrum features of the speech are then obtained by homomorphic signal processing what is discussed in [9]. This involves taking the inverse FFT of the logarithmic spectrum of the short term spectrum of the signal. These steps are shown in equation (12.9). The FFT based cepstrum features are the inverse Fourier transforms of the logarithm of the magnitude of its FT of the finite length data sets.

**Spectral shaping by windowing the signals** The letters here have a different meaning than in the earlier notation. The signal is framed into $m$ blocks by applying a window function of length $N$ in such a way that each signal of the $m$ blocks has $N$ many samples where $m = 1, 2, \cdots, M$ and $n = 0, 1, \cdots, N-1$.

This segmented signal is called $s_m[n]$: this signal is a frame. The formulation of the framed signal by the windowing is shown in equation (12.4). The length of the window is equal to the signal length $N$. Thus the framed or the windowed signal $s_m[n]$ is the multiplication of the signal $s[n]$ by the window function $w[n]$. Multiplying the signal $s[n]$ by the window function $w[n]$, we attain the signal frames denoted by $s_m[n]$ for $m = 1, 2, \cdots, M$ and $n = 0, 1, \cdots, N-1$, shift $r$ which is the shift samples between adjacent segments and $r \leq N$.This shifting is sometimes known as overlapping. This can take place at each one half or two third of the length of the signal where the length of the signal is assumed to be the same as the length of the signal block. This is shown in equation (12.4).

$$s_m[n] = s[n + mrw[n] \tag{12.4}$$

There are different types of window functions available. Some commonly used window functions in the speech research are the rectangular window function given in equation (12.5), the Hamming window function given in equation (12.7), and the Hanning window function given in equation (12.6). One purpose of the windowing is to improve the leakage problem and results in a spectral bias in the frequency analysis. The leakage problem arises when signal is truncated into blocks. The formulation of the rectangular window is :

$$w[n] = 1 \quad \text{for} \quad 0 \leq n \leq N-1 \text{ and } \quad 0 \text{ otherwise} \tag{12.5}$$

The formulation of the Hanning window is :

$$w[n] = 0.5 - 0.5\cos(\frac{2\pi n}{N-1})) \quad \text{for } n = 0, 1, 2, \cdots, N-1 \tag{12.6}$$

The formulation of the Hamming window is :

$$w[n] = 0.54 - 0.46\cos(\frac{2\pi n}{N-1}) \quad \text{for } n = 0, 1, 2, \cdots, N-1 \tag{12.7}$$

We will not discuss the derivations of the window functions and the constants used in equations (12.5), (12.7) and (12.6). These are some commonly used window functions in digital signal processing (DSP).

**Spectral analysis and feature transformation** The cepstrum is computed by taking the inverse Fourier transform (IDFT) of the logarithm of the squared magnitude in the DFT for the signal given in equations (12.8) and (12.9) [14].

**Logarithmic power spectrum** In this step, first the logarithm of the square of the spectrum of the windowed signal $S_m(k)$ of the windowed signal $s_m[n]$ is computed by taking the logarithmic computation of the square of the absolute value of the magnitude of the DFT for the windowed signal. This is shown in equation (12.8).

$$S_m(k) = 10\log_{10}(|\sum_{n=0}^{N-1} s_m[n]e^{-j2\frac{\pi}{N}kn}|^2) \tag{12.8}$$

**Parametric feature transformation** The cepstrum parameters for the windowed signal are denoted by $\beth_m[n]$ and are computed by equation (12.9) which is the inverse DFT of equation (12.8).

$$\beth_m[k] = \frac{1}{N}\sum_{k=0}^{N-1} S_m(k)e^{2\frac{\pi}{N}kn} \tag{12.9}$$

The whole cepstral operation in order to extract cepstral features is shown in figure 12.2. There we see the signal is first segmented into framed signals and the discrete Fourier transform (DFT) of the framed signals is computed, then the inverse DFT is taken on the logarithm of the DFT transformed signal.

In figure 12.3, we see the features extracted from speech command Öffne die Tür. Extracted features do not show good structure of the features. By good, we mean the difference between the features are not treated well by the cepstral feature extraction technique.

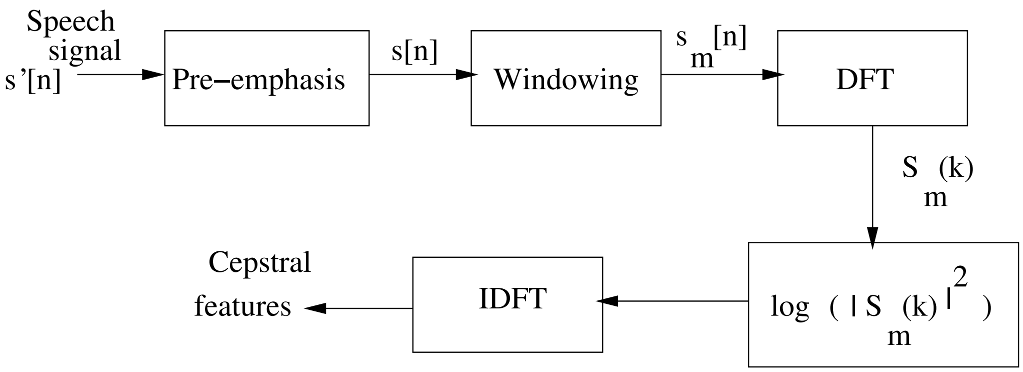


Figure 12.2: Cepstral feature extraction technique

The MFCC, LPCC, PLP feature extraction techniques are more or less an extension of the cepstral feature extraction technique. This is our next topic.

## 12.6 MFCC Feature Extraction Technique

The MFCC extracts perceptual speech features for the speech recognition. It is probably the first perceptual speech feature extraction technique for the speech recognition system and is examined in [110]. This is one of most commonly used feature extraction techniques in the ASR technology.

The MFCC extraction process is shown in figure 12.4.

The relations among the steps in MFCC are shown in figure 12.4 and are then discussed. Next we give a short descriptions of the components shown in figure 12.4.

**Pre-emphasis** First the input signal denoted by $\acute{s}[n]$ is pre-emhasized by using a pre-emphasis filter. The pre-emphasized signal is $s[n]$ for $n \in \mathcal{Z}$ (integers) and $a_{em} \approx 0.97$. The pre-emphasis filter repeated here is already shown in equation (4.3) in chapter 4 in section 4.3. There $\acute{s}[n]$ is denoted by $s^{'}[n]$ (observe $'$ does not denote here a transpose or $'$ does not denote a transposein in chapter 4).

$$s[n] = \acute{s}[n] - a_{em}\acute{s}[n-1]$$

The specific value is a kind of standard resulting from experience and will not be discussed here.

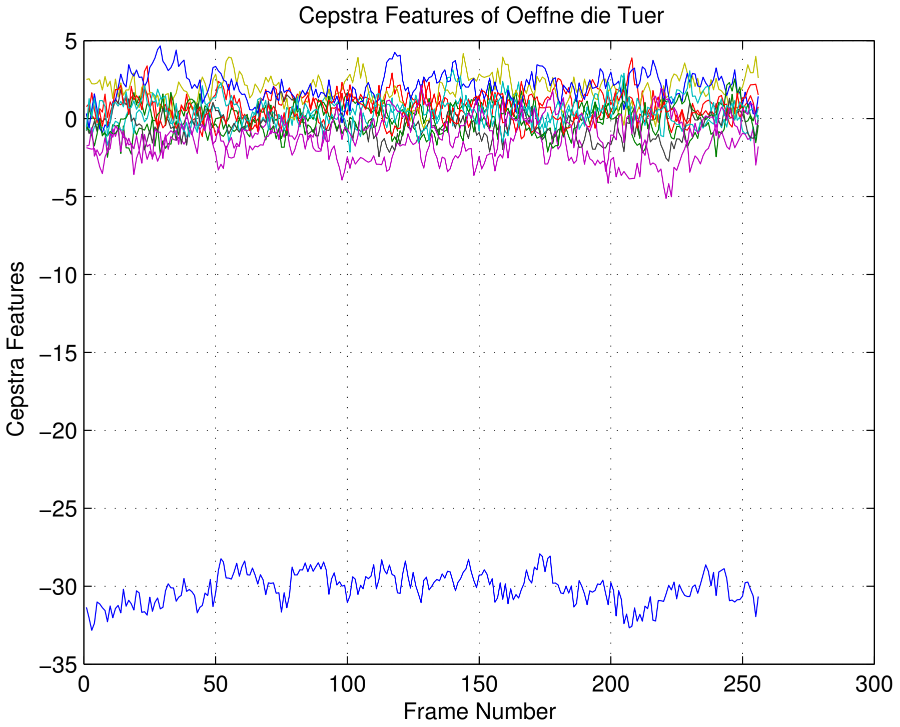


Figure 12.3: Cepstra features

**Windowed signal** The signal is windowed into $s_m[n]$ following the same procedure as described in section 12.5 and the same formulation as given in equation (12.4). Then $s_m[n]$ is used for the power spectrum computation in the next stage. Windowing is descrbed in detail in section 12.5.

**Logarithmic power spectrum** The windowed signal is used to compute the logarithm of the square of the spectrum of the windowed signal using equation (12.8) given in section 12.5.

**Perceptual spectral analysis** The human ear has a high frequency resolution in the low frequencies and a low frequency resolution in the high frequencies. In order to reflect the frequency resolution property of the human ear, the power spectrum obtained by equation (12.8) is multiplied by the Mel scaled filter banks. The Mel scaled filter banks given in equation (11.12) in chapter 11 is also known as Mel filter bank. This is a set of triangular shaped filter banks which is scaled by a Mel scale. The Mel scale is given in equation (11.7) in chapter 11. This

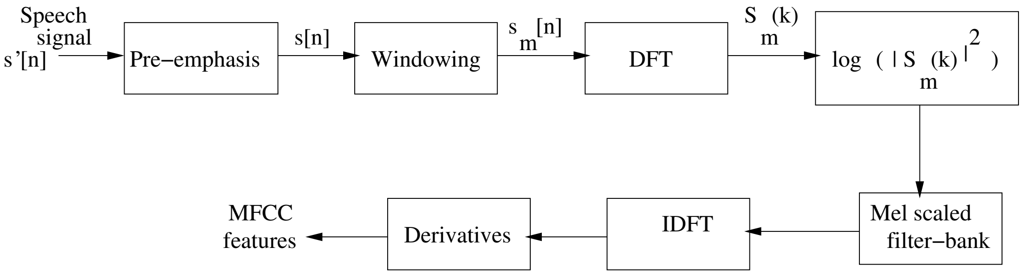


Figure 12.4: MFCC feature extraction technique

scale is linearly spacing up to 1000 Hz and logarithmic spacing above 1000 Hz.

In equation (12.10) (given in chapter 11), $H_l(k)$ is the Mel filter bank coefficients for $l = 0, 1, 2, \cdots, L$ where $L$ (observe $l$ and $L$ should not be confused with the notations in chapter 8) denotes the number of filters that are mapped to the Mel scale and $k = 0, 1, \cdots, \frac{N}{2}$. Due to the FFT symmetry property, the computation is taken for $0 \leq n \leq \frac{N}{2}$ which is one half of total lengh $N$. In equation (12.10), $\aleph_m(l)$ is the perceptual power spectrum of $l$ th critical band in the Mel scale and the subscript $m$ denotes the power spectrum of the $m$ th frame. $H_l(k)$ is defined in equation (11.12) in chapter 11.

$$\aleph_m(l) = \sum_{k=0}^{\frac{N}{2}} H_l(k) S_m(k) \tag{12.10}$$

**Logarithmic perceptual power spectrum** In this step, the logarithm of the squared magnitude of the output of the Mel filter bank is computed. The logarithm of the spectrum of each filter-bank is obtained on each frame using equation (12.11). The reason for computing the logarithm on the mel power spectrum of the speech frames is to compress the wide-ranging varieties to follow the dynamic range compression characteristics of the human hearing system. In equation (12.11), the ouput of the filter is $P_m[l]$ for $l = 1, 2, \cdots, L$ many triangular band pass filters and $m = 1, 2, \cdots, M$.

$$P_m(l) = 10 \log_{10} | [\aleph_m(l)] | \tag{12.11}$$

**Feature transformation applying DCT** Since the log power spectrum is real and symmetric the inverse DFT is equivalent to the discrete cosine transformation (DCT). The detailed information about this computation and the related transformation can be found in [103]. Windowing and DCT is considered in section 12.10 and chapter 13. The DCT has the property to produce uncorrelated features and thus the features variations can be set using a diagonal assumption. This property is used to model the speech features using the Gaussian mixture model (GMM) for the classification in the HMM recognition discussed in chapter 14. Now the DCT of $P_m(l)$ is computed by the formulation given in equation (12.12) where $l = 1, 2, \cdots, L$ and $n$ the DCT coefficients where $n = 0, 1, 2, \cdots, J$.

$$c_m[n] = \sum_{l=1}^{L} \big(P_m[l] \cos(\pi n(l - 1/2)/L\big) \tag{12.12}$$

The $c_m(n)$ are the standard coefficients of the MFCC features. But more dynamic information of the speech can be captured by taking derivatives of the speech. This is discussed below.

**Derivatives of the feature transformation** Since the speech signal is a random process that changes over time, the speech signal is analyzed on the frames. The dynamics of the changes of the features can be further captured by taking first and second derivatives of the features obtained at the previous step. The purpose of these derivatives is to capture any changes that may take place during the feature transformation from the time domain to the feature domain at their perception stage. The first derivative of the MFCC standard feature is called delta feature. How this is computed is shown in equation (12.13). Here $m$ is used for the speech frames and $m = 1, 2, \cdots, M$. $n$ denotes the number of features from each frame and $n = 1, 2, \cdots, J$. The derivative is computed with respect to twice the signal window.

The second derivatives of the MFCC standard feature is called delta-delta features. It is computed by equation (12.14). These formulations are collected from [115], [54].

$$\Delta c_m[n] = c_{m+2}[n] - c_{m-2}[n] \tag{12.13}$$

$$\Delta^2 c_m[n] = \Delta c_{m+1}[n] - \Delta c_{m-1}[n] \tag{12.14}$$

**Signal energy** Now an additional feature in the MFCC analysis is the signal energy. It can be computed separately and then this can be included in the feature parameters. This is computed by equation (12.15). The energy MFCC feature coefficients are denoted by $e_m$ where $m$ denotes the frame number. The notation $e$ should not be confused with the error denoted by $e$ in chapters 6,7, 8,9 and 10. Each frame has only one frame energy and this is normally the first feature in the feature vector where each vector is representing each speech frame.

$$e_m = \sum_{n=0}^{N-1} s_m^2[n] \tag{12.15}$$

We see that the MFCC feature parameters are energy of the frames, cepstral coefficients which is extracted using DCT, delta, and delta-delta cepstrum coefficients. Such MFCC features $c_m$ are shown in equation (12.16).

$$c_m = [e_m, c_m[n], \Delta c_m[n], \Delta^2 c_m[n]] \tag{12.16}$$

In figure 12.5, we see the features extracted from speech command Öffne die Tür. These features in each frame are better managed than cepstral features.

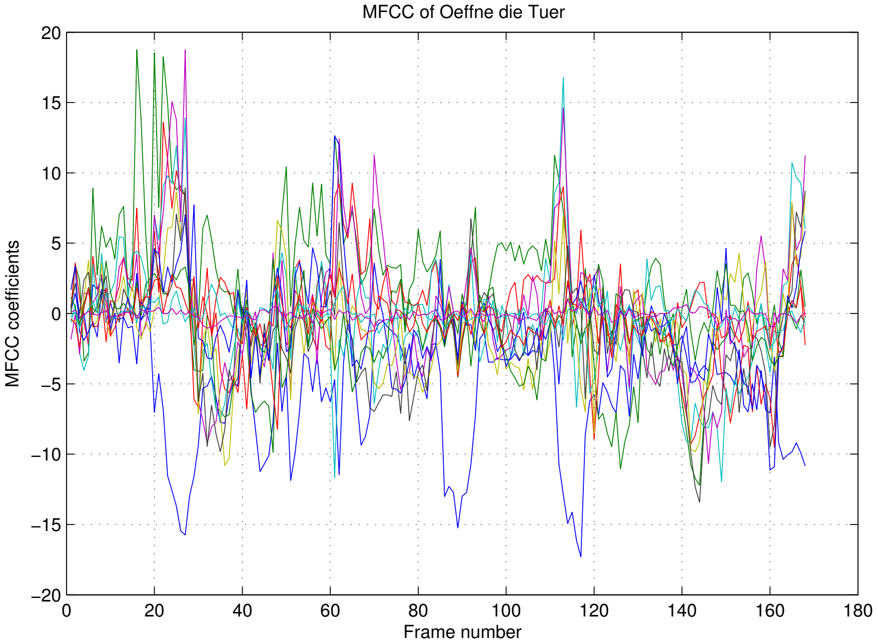


Figure 12.5: MFCC features

## 12.7 LPC Feature Extraction Technique

The basic idea of this technique is to the obtain the auditory spectrum and then to obtain the features. In the LPC feature extraction the signal is segmented and windowed by using equation (12.4). The linear prediction (LP) analysis described in chapter 8 are used to obtain LPC coefficients. These LPC coefficients are the LPC speech features used for the classification and the recognition. An extended version of this, LPCC, is discussed next. The LPC feature extraction technique is shown in figure 12.6.

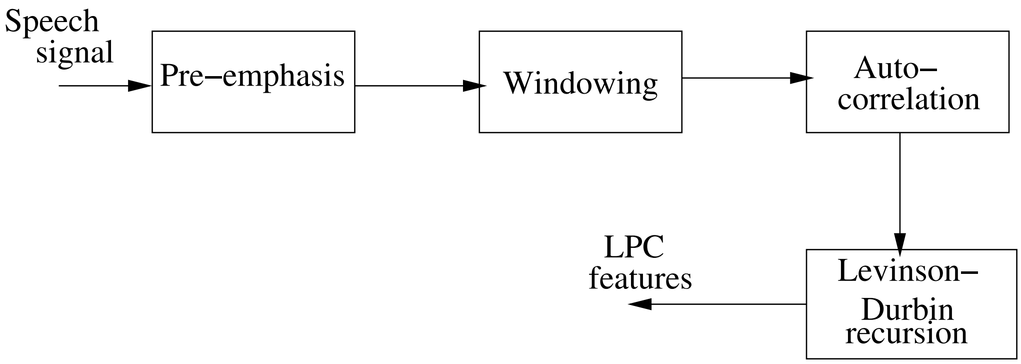


Figure 12.6: LPC feature extraction technique

## 12.8 LPCC Feature Extraction Technique

The LPCC feature extraction technique is a combination of the LPC analysis and the cepstrum analysis discussed in section 12.5. The basic idea of this technique is the same as in the LPC analysis. In this technique, the windowed signal is used for computing the autocorrelation and the power spectrum and the inverse DFT of the logarithm of the power spectrum is computed for the LPCC features. A block diagram of the LPCC feature extraction technique is shown in figure 12.8.

In figure 12.9, we see the features extracted from speech command Öffne die Tür. These features in each frame are good managed as it is done in MFCC features.

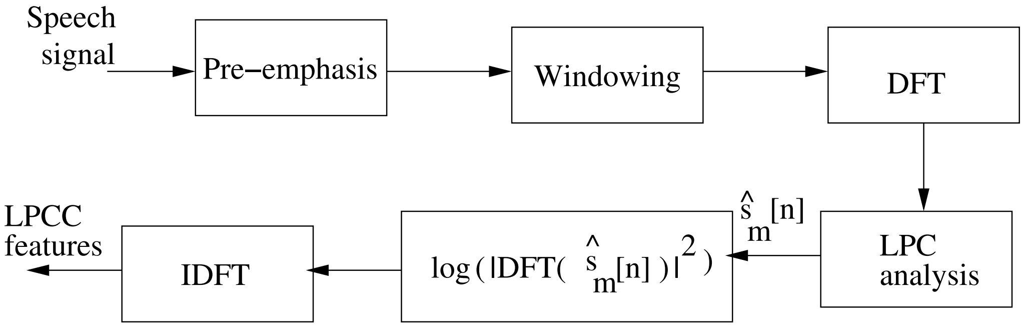


Figure 12.7: LPCC feature extraction technique

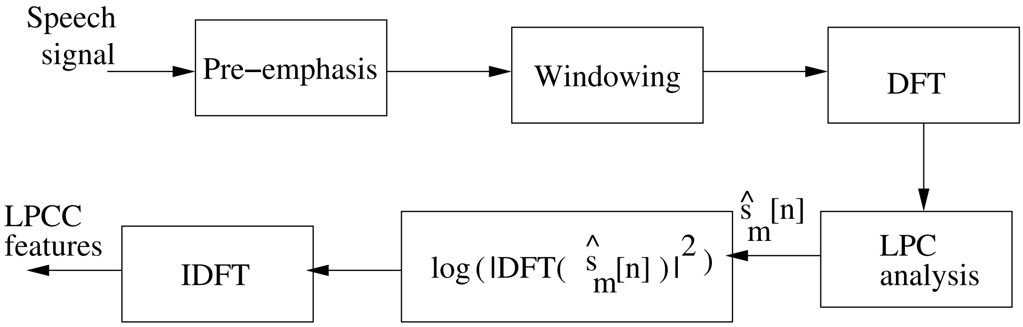


Figure 12.8: LPCC feature extraction technique

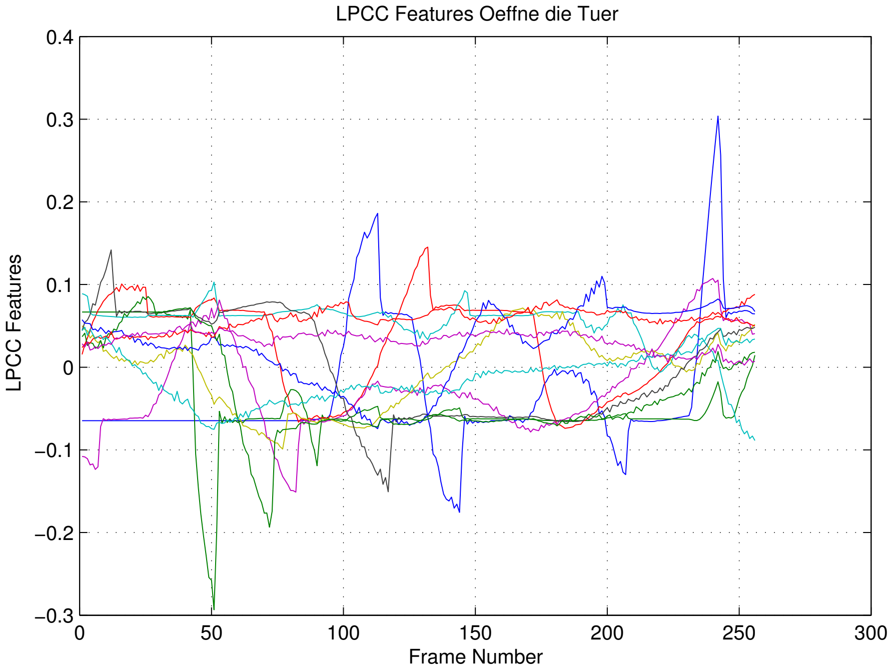


Figure 12.9: LPCC features

## 12.9 PLP Feature Extraction Technique

This Perceptual Linear Predictive (PLP) technique relates the psychoacoustics studies to the auditory spectrum in order to create perceptual PLP features . It uses the LP analysis, the cepstrum analysis and the insights about human perception. This technique is described in [42]. The steps are:

- Spectral Shaping: This is the same as in the cepstral analysis and the MFCC analysis.

- Perceptual Spectral Analysis: This first uses human speech perception with critical band analysis, equal loudness pre-emphasis and intensity loudness conversion and adapt these to the speech which are already spectrally shaped.

- Perceptual Feature Transformation: This is the same as the cepstral analysis and the MFCC analysis.

- Feature Transformation: This uses the vocal tract model of the perceptual feature transformed speech and then solves the problem of finding the model parameters using the LPC analysis. These parameters are the PLP features.

### 12.9.1 Perceptual Spectral Features

The spectrally shaped speech is now mapped to psycho-acoustical quantities in order to approximate the human speech perception. The psycho-acoustical quantities are : i) Critical band analysis, ii) Equal loudness pre-emphasis and iii) Intensity loudness. These are introduced below.

**Critical band analysis** This uses the Bark scale in order to approximate human hearing perception. The formulation of the Bark critical band analysis is given in section 11.7.2 in chapter 11. Equation (11.15) and equation (11.14) explained in chapter 11 in section 11.7.2 is mainly used by [42] for PLP perceptual spectral analysis.

In this explanation, each of the filter's output is the sum of the product of the windowed FFT speech signals and the $\Phi_m(k)$ weight given in eq (12.17). There $m = 2, 3, \cdots, (M-1)$ where $m$ is the running index of the filters in the filter-bank and $k$ denotes the frequency of certain $m$ th filter. The first filter $m = 1$ is at 0 Barks and the last filter at $m = M$. The output of the first and last filters are calculated in such a way that the resultant output of these two filters is equal to their closest filter output [100].

$$\mathbf{X}_m(k) = \sum_{k=0}^{\frac{N}{2}-1} \mid S(k) \mid^2 \mid \Phi_m(k) \mid \tag{12.17}$$

Here $X_m$ is the filter output of the $m$th filter and $\mid S(k) \mid^2$ is the $N$th power spectrum of the windowed speech frame and $\mid \Phi_m(k) \mid$ is the filter weights of the $m$th Bark filter corresponding to the linear frequency scale [100].

**Equal loudness pre-emphasis** This re-emphasizes the spectrum to approximate the unequal sensitivity of human versus technical frequency to approximate the equal loudness curve. This is known as the psychophysical equal-loudness pre-emphasis which reduces the spectral amplitude variation between the critical band spectrum and the linear frequency band spectrum.

In equation (12.18), $E_m$ is the weighted equal loudness obtained by equation

(12.19).

$$\eta_m(k) = \sum_{k=0}^{\frac{N}{2}} E_m(k) X_m(k) \tag{12.18}$$

In equation (12.19), $\omega$ is an angular frequency and $\omega = 2\pi f$ and $m$ is the signal segment and $f$ is the linear frequency. $\centerdot$ denotes the multiplication.

$$E_m(\omega) = \frac{(\omega^2 + 56.8 \centerdot 10^6)\omega^4}{(\omega^2 + 6.3 \centerdot 10^6)^2(\omega^2 + 0.38 \centerdot 10^9)} \tag{12.19}$$

**Intensity loudness** Given the signal intensity, equation (12.20) approximates the human loudness perception. Eq (12.20) is a cubic-root amplitude compression which relates the sound intensity and perceived loudness [100].

$$\gimel_m(k) = \sum_{k=0}^{\frac{N}{2}} (\eta_m(k))^{\frac{1}{3}} \tag{12.20}$$

**Perceptual Feature Transformation** It uses the cepstral coefficients of the perceptually spectral speech features spectrum by taking the inverse discrete transform of the perceptually spectral speech features for its perceptual feature transformation using equation (12.9) given in section 12.5.

**Feature Transformation** Here this uses the autoregressive approach for the perceptually transformed features for a vocal tract modeling and the solution to the parameters are obtained by using LPC analysis given in chapter 8. These LPC parameters are the PLP features.

Figure 12.10 shows the PLP feature extraction technique. It contains the components that have been already introduced in this technique.

More about the intensity and loudness can be found in psychoacoustics studies [41]. We have not done a detailed investigation of these studies. Our literature survey on the perceptual feature extraction is given in chapter 11. In figure 12.11, we see the features extracted from speech command Öffne die Tür. These features in each frame are good managed but it shows some constant value. This might be the energy extracted from each feature vector.

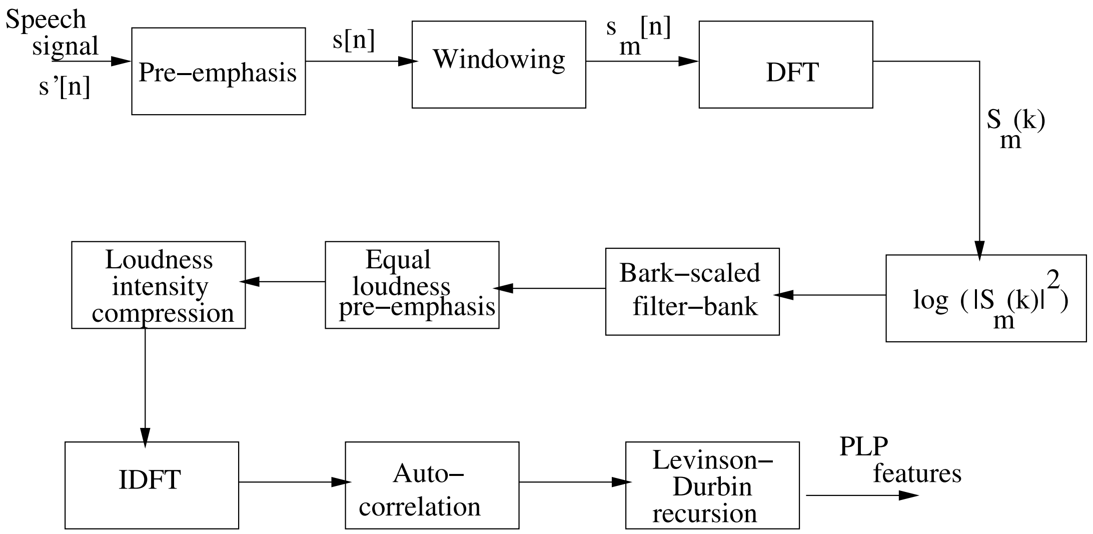


Figure 12.10: PLP feature extraction technique

## 12.10 SILTT Feature Extraction Technique

As said, the local trigonometric transformation (LTT) is the basis to extract shift invariant local trigonometric transform (SILTT) features. The idea of this technique is to build a library of functions of an orthonormal basis that compared to the given signal or collection of signals has the lowest information cost.There are several possibilities to compute the cost function. Examples are Shannon entropy, Neumann entropy, wavelet entropy etc. The Shannon spectral entropy is a common such cost function.

One first selects the window width. Then the procedure is repeated to construct the best basis from all possible local cosine and sine bases using the cost functional. For the search for the local cosine and sine basis the best matching to the signal in terms of the cost function defined on the entropy of the decomposed signal l is selected. The entropy then gives the SILTT features.

In figure 12.12, we can see that the signal is first segmented, windowed and folded in order to extract spectral coefficients using DCT IV. These are then used to extract spectral entropy. These spectral features are then given to the inverse DCT IV. The results are unfolded in order to extract SILTT features. We extended the SILTT for our feature extraction and a detailed formulation of this technique and its extended version in our approach is discussed in the next chapter.

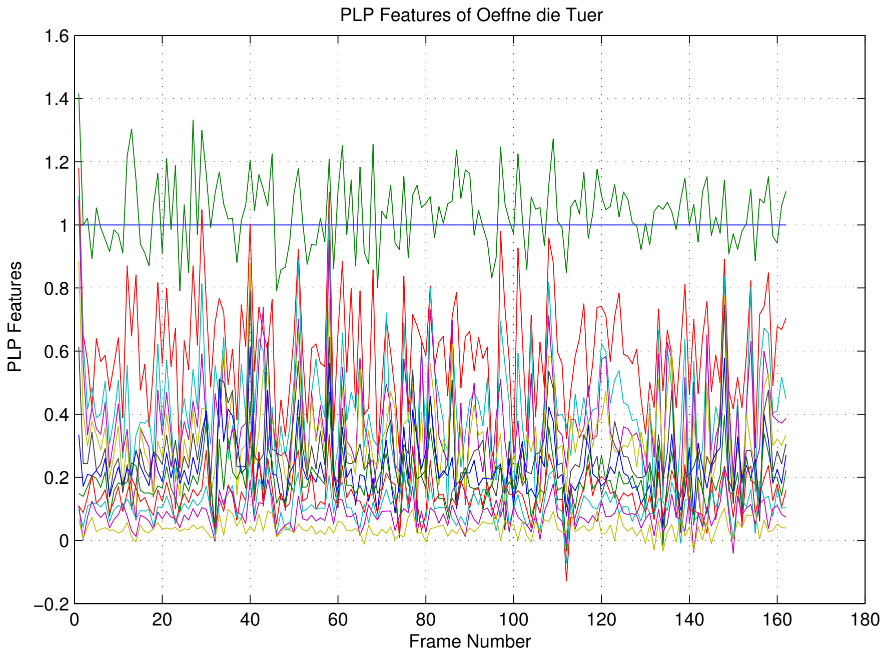


Figure 12.11: PLP features

## 12.11 Additional Features and their Extractions

The cepstrum, MFCC, LPC and PLP feature extraction techniques have been used for both the clean and noisy speech recognitions. Over the years there are several other feature extraction techniques developed by modifying the cepstrum, MFCC, LPC and PLP feature extraction techniques. Examples are the cepstral mean normalization (CMN). The CMN is described in [33]. The multilevel CMN is the sub-band based CMN. This is described in [40]. Next we list several systems just for completenes without analysis. Relative spectral processing known as RASTA extending PLP feature extraction technique is described in [43]. The RASTA processing suppresses the noise in the overlap-add analysis-synthesis section of the signal. A detailed description of the overlap-add analysis-synthesis process can be found in [60], [9]. This RASTA spectral processing is done on the cube root of the power spectrum by setting of all negative spectral values to a small positive constant [44]. Details on the RASTA processing can be found in [1]. An extension of the RASTA feature extraction technique for additive and convolution noise has been used for the speech feature in [99]. These are all essentially an extension of PLP technique and a slight modification of the RASTA

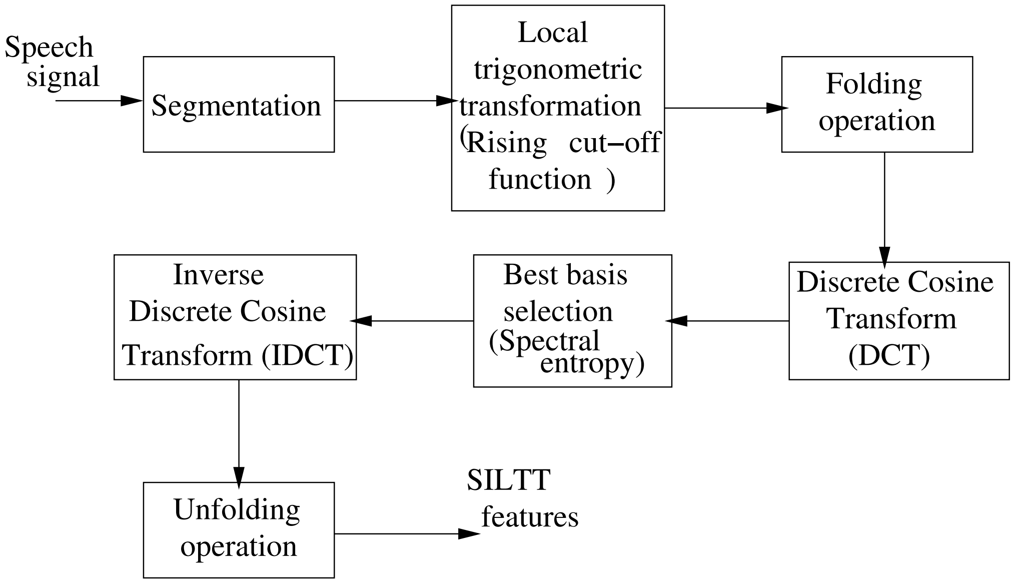

Speech signal
Segmentation
Local trigonometric transformation (Rising cut−off function )
Folding operation
Discrete Cosine Transform (DCT)
Best basis selection (Spectral entropy)
Inverse Discrete Cosine Transform (IDCT)
Unfolding operation
SILTT features

Figure 12.12: SILTT feature extraction technique

using the spectral analysis in sub-bands. The reconstruction of missing features extending the MFCC feature extraction technique is done in [16]. Relating the psycho-acoustics to multi-channel processing using RASTA in the framework of HMM and artificial neural network (ANN) is done in [40]. There are also some other feature extraction techniques e.g. warped or perceptual minimum variance distortionless response (MVDR) which is based on the unbiased minimum variance estimate of the spectral components [136]. The SILTT has been used for speech processing, the wheezing lung sounds analysis, seismic data processing, speech recognition in [111], [101], [32], [95], [50], [88]. These techniques are mainly taking the cepstral mean normalization or by applying some additional digital filtering using full band or in the sub-band. They are used for the clean speech recognition and noisy speech recognition. We have not investigated further details of these features at this stage.

## 12.12 Analysis of Feature Extractions

Here we evaluate and compare the standard feature extraction techniques. We rely either on the results in this thesis or the cited literature. In the standard feature extraction methods such as LPC, PLP, MFCC, the time-axis of the speech signal is divided into a series of some time intervals and then some functions are used to smoothen the intervals in an overlapping or non-overlapping manner. It can happen that the speech signal is non-stationary even though it is analyzed in most situations as a quasi-stationary typed. The problem in the standard methods is that the division of the speech signal may be processed as unchanged in each 10 to 30 ms and the change may not be fixed in a consecutive manner. Here the situation can be improved in some ways if there is a flexibility in the speech signal processing. This flexibility is managed in the SILTT feature extraction method.

The difference between the Cepstrum and the simple Yule-Walker approach for the LP technique is the Fourier spectrum of the segment and computation of the autocorrelation process of the segment and then computing the spectrum using the FFT. In both cases, there is a problem to determine the true spectrum. It is observed that the MFCC using Fourier transform (FT) for the Mel filter-bank cepstral coefficients gives better recognition results for recognizing clean speech and speaker [68]. The PLP based on FT using Bark-filter bank cepstral coefficients and LP technique performs well in recognizing noisy speech [1]. The basic difference between MFCC, LPCC, PLP using the FFT and SILTT applying the DCT is in the approximation of the human auditory perception. In MFCC, the frequency bands are positioned logarithmically using the Mel frequency scale. In PLP, the frequency bands are positioned logarithmically using the Bark frequency scale following the critical bandwidth and equal loudness approximation approach [144].

In SILTT, the perceptual feature transformation is not used. In the LPC, the pre-emaphsis is done at the front end, and on the contrary, the PLP uses equal-loudness filtering. In LPC, the linear spectral analysis is followed, where in PLP analysis, it is compressed to the critical band spectrum. In LPC, the cepstrally smoothed spectrum is the transformed features. In PLP, the LPC based smoothing of the spectrum is considered for the feature transformation [62].

In MFCC and PLP the signal decomposition and spectral analysis is followed by the process of the lapped transformation where the FFT is applied. The problem of abrupt discontinuity is it reduced because of the lapped transformation. We handled this by applying a local trigonometric transformation followed by the lapped transformation and extra care is taken to the edge applying the folding operation. The discontinuity is smoothened here better than in the traditional MFCC and PLP. The other problem in the LTT transformation is that it is not transformed to the perceptual conversion. That we have done in this feature extraction technique.

Even the existing standard feature extraction methods are quite successful in speech recognition. However, the success has a limited and restricted domain. In some situations they work well at some scenario, but sometimes they do not. However, in our analysis and evaluation, the MFCC features shows better structured than other commonly used feature extraction techniques namely ceptral features, LPCC features and PLP features.

# Chapter 13

# APLTT Feature Extraction

**Outline of the chapter** In this study we apply the adaptive perceptual local trigonometric transformation (APLTT) to extract features for the DANSR. This chapter provides the architecture and definition of the APLTT feature extraction technique. The APLTT is an extension of the shift invariant local trigonometric transformation (SILTT) introduced in the previous chapter. This is a new perceptual feature extraction technique because we extend the SILTT by adopting some psychoacoustic quantities into it. The adopted psychoacoustic quantities are: Erb scaled critical band spectral analyzer, loudness and masking properties and perceptual entropy. We explain here how this is done.

The feature extraction steps are in general:

- Spectral Shaping: The signal is decomposed into blocks using a lapped transformation followed by a folding operator.
- Spectral Analysis: This is done by computing discrete cosine transform IV (DCT - IV).
- Perceptual Mapping: This is done by computing perceptual spectral information using a Bark scale. This is an extension of the SILTT.
- Perceptual Feature Transformation : The perceptual features are the perceptual entropy (PE). This is a modification of the SILTT.
- Parametric Feature Transformation : This is done by computing the inverse of the discrete cosine transform IV (IDCT - IV) followed by an unfolding operation.

Below each of the steps is described independently.

## 13.1 Spectral Shaping

In spectral shaping the signal is prepared for the spectral analysis. The spectral shaping has several sub steps:

- Decompose the signal in a dyadic manner
- Windowing the signal
- Apply a folding operator

These steps are now discussed below:

### 13.1.1 Signal Decomposition

The signal $s(n)$ is decomposed in such a way that the length of each block (segment) is a power of 2. The length of the decomposed signal is in the range of 10 to 30 milli seconds (ms). Each block has $N$ many samples such that $n = 0, 1, 2, \cdots, 2^{\mathbb{N}} - 1$. Each block is denoted by $I_j$ where $j = 0, 1, 2, \cdots, 2^{\mathbb{N}} - 1$.

### 13.1.2 Windowing the signal

Windowing is in principle an extension of segmentation. The purpose of it is to overcome generated discontinuities.

The starting point are segments that are intervals $I_j = [a_j, a_{j+1})$. Next a number $r$ is chosen such that $a_j + r, < a_{j+1} - r$, $\forall j$. The intervals are taken as $[a_j - r, a_j + r)$. The number $r$ describes the overlapping. This is used for smoothly combining the intervals. The smooth windowed signal is obtained by applying the trigonometric cutoff functions what is discussed next. It starts from taking a function $\beta(t)$. The function $\beta(t)$ is such that $\beta(t) \in \Re^d$ with $0 \leq d$ satisfying the following condition:

$$\beta(t) = \begin{cases} 0 & \text{if } t \leq -1 \\ 1 & \text{if } t \geq 1 \end{cases} \tag{13.1}$$

This condition will now be extended to a full definition in the interval -1, 1.

The function $\beta(t)$ is called a rising cutoff function because $\beta(t)$ rises from being identically zero to being identically one as $t$ goes from $-\infty$ to $+\infty$. In equation

(13.2), $b_j(t)$ is defined for any real number and it is a continuous function.

$$b_j(t) = \begin{cases} \beta(\frac{t-a_j}{r}) & t \in [a_j - r, a_j + r) \\ 1 & t \in [a_j + r, a_{j+1} - r) \\ \beta(\frac{a_{j+1}-t}{r}) & t \in [a_{j+1} - r, a_{j+1} + r) \\ 0 & t \in (-\infty, a_j - r] \cup [a_{j+r}, \infty) \end{cases} \tag{13.2}$$

To ensure smoothness we take a number $\epsilon$ such that $a_{j+1} - a_j \geq \epsilon > 0$ for each $j \in Z$ The whole complex process is shown in figure 13.1. In the sequel we will define the steps. The framed signal is already discussed in sections 13.1.1 and 13.1.2.

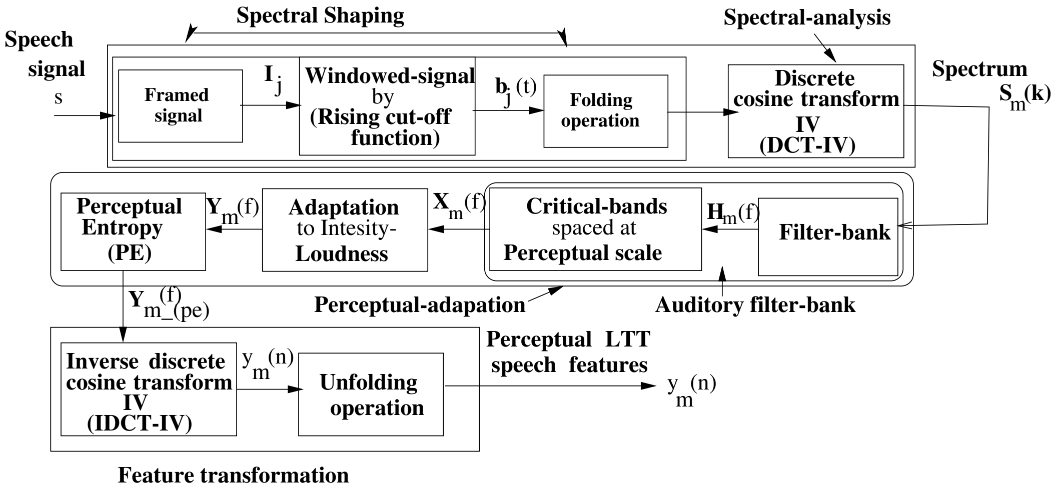


Figure 13.1: DANSR feature extraction: Adaptive Perceptual LTT (APLTT)

### 13.1.3 Rising Cut-off Functions

In this section we define some operators that are used for windowing in a more general way. The main element of these transformations is the unitary operator which transforms a sharp cut off function into a smooth orthogonal projection. This operator is called folding operator. The folding operator depends on the real variable and must satisfy the symmetry property. The motivation of the rising cut-off function: We introduce the rising cut-off function in order to provide a smooth integration of the segments. The smoothness results in a projection depending on the parameter function's smoothness [32], [95].

For this we want to specify $\beta(t)$ more in the inner interval $[-1, 1]$ and two more functions $\rho$ and $\theta$ are introduced. In equation (13.3), one has to choose the numers $m, n \in Z$ in equation (13.3).

$$\rho(t) = \begin{cases} 2n\pi & \text{if } t \leq -1 \\ 2m\pi & \text{if } t \geq 1 \end{cases} \tag{13.3}$$

We take $r_0(t) = r\sin(t)$ and $r_1(t) = r(\sin\{\sin(\frac{\pi}{2}t)\})$. Thus we can say $r_1(t) = \sin(\frac{\pi}{4}(1 + \sin\frac{\pi}{2}t))$ and define in equation (13.4):

$$\theta(t) = \begin{cases} 0 & \text{if } \ t < -1 \\ \frac{\pi}{2} & \text{if } t > 1 \end{cases} \tag{13.4}$$

By choosing $\theta(t) + \theta(-t) = \frac{\pi}{2}$, we can specify the rising cut-off function $\beta(t)$ as shown in equation (13.5) where $i$ is used in $\exp[i\rho(t)]$ to indicate a complex exponential term.

$$\beta(t) = \exp[i\rho(t)]\sin(\theta(t)) \tag{13.5}$$

Since $|\beta(-t)| = |\sin[\frac{\pi}{2} - \theta(t)]| = |\cos(\theta)|$, we obtain $|\beta(t)|^2 + |\beta(-t)|^2 = 1$. The $\beta$ function is now defined as $\beta(t)$:

$$\beta(t) = \begin{cases} \sin\frac{\pi}{4}(1 + \sin\frac{\pi}{2}t) & \text{for } -1 \leq t \leq 1 \\ 0 & \text{for } t < -1 \\ 1 & \text{if } t > 1 \end{cases} \tag{13.6}$$

This functions satisfies the needed conditions.

### 13.1.4 Folding Operation

A basic operation is now to split each interval into two overlapping intervals and construct a basis for each one. The integration is obtained by a rising cut-off function. The local cosine transform reduces the blocking effects and smoothens the signal. The principle of overlapping between the adjacent blocks is used in the local cosine trigonometric transformation. There we use smooth cut-off functions to split the signal and to fold the overlapping sections back into segments in such a way that the orthogonality is preserved. To obtain a better frequency localization, the signal is multiplied by a smooth window function which uses the local sine and cosine bases consisting of sine or cosine multiplied by a smooth compactly supported bell functions. These localized sine and cosine functions are orthogonal. The basis element is characterized by the position $\alpha$, the interval $I$, and frequency index $k$. This generates the smooth local trigonometric basis. The (DCT-IV(discussed below) is used for folding the overlapping sections back into the interval $I = [a_j, a_{j+1})$ using the bell function $b_j(t)$.

Equation (13.7) shows an inner product of the local cosine basis using the folding operator and DCT-IV. There the $s_j[n]$ start with the multiplication of the window $b_j(t)$ and the signal $s[n]$. The multiplication is folded at the edges using the folding operator and then DCT IV is applied on the folded result giving the disjoint window segments $s_j[n]$.

$$s_j[n] = \begin{cases} b_j[n]s[n] + b_j(2a_j - n)s(2a_j - n) & \text{if } a_j \leq n \leq (a_j + r) \\ s[n] \text{ if } a_j + r \leq n \leq [a_{j+1} - r \\ b_j[n]s[n] + b_j[2a_{j+1} - n]s[2a_{j+1} - n] & \text{if } a_{j+1} - r \leq n \leq a_{j+1} \end{cases} \tag{13.7}$$

In short, we denote this as $s_j[n] = b_j[n]s[n]$. Observe that $s_j[n]$ is 0 outside of the interval.

In figure 13.2, we see the folding operation is explained in figure a. In this figure, we see the folding operation is done symmetrically in the middle of the bell function where left side shows -0.5 and right side shows 0.5. Thus the windowing proceeds taking the middle of the left window and middle of the right window. The both mid points of the windows are folded. This is shown in figure b in the same figure.

## 13.2 Spectral Analysis

Up to now we still have been in the time domain. Now we go to the z-domain. We will use a kind of the Fourier transformation. After segmentation we would like to do this for each segment. The problem is now coming from the discontinuities at the boundaries of the segments. For dealing with it we introduced overlapping intervals and the rising cut-off functions. The transformation that we will use now is the Discrete Cosine Transform IV (DCT-IV). The spectral analysis is ilustrated in figure 13.2.

### 13.2.1 Discrete Cosine Transform IV (DCT-IV)

The DCT IV as a kernel function is shown in equation (13.8). There $n = 0, 1, 2, \cdots, (N/2 - 1)$ and $k = 0, 1, 2, \cdots, (N/2 - 1)$. A detailed derivation of

DCT family can be found in [39], [103]. For describing properties of DCT-IV, we define:

$$F_k = \sum_{n=0}^{N-1} \frac{\pi}{N} \cos\left([n+\frac{1}{2}][k+\frac{1}{2}]\right) \tag{13.8}$$

Here we represent the notations for the local trigonometric symmetric case where the signal is windowed and folded prior to its spectral representations. It has also the property of recovering the original representation of the signal. We now introduce (13.9), $\chi_{I_j}[n]$ is the characteristic function of the interval which is 1 for $n \in I_j$ and 0 otherwise.

$$\phi_k^j[n] = \frac{\sqrt{2}}{\sqrt{|I_j|}} \cos \frac{\pi}{|I_j|}[k+\frac{1}{2}][n-a_j]\chi_{I_j}[n] \tag{13.9}$$

Folding has the property of recovering the original representation of the signal

$$c_k^j = \left\langle S_j[n], \phi_k^j(n) \right\rangle \tag{13.10}$$

This allows the representation shown in equation (13.11). The inner products of the $c_k^j$ and $\phi_k^j[n]$ are the DCT coefficients.

$$S_j[n] = \sum_{k\in\mathcal{Z}} c_k^j \phi_k^j[n] \tag{13.11}$$

In figure 13.2, we see the spectral analysis In figure 13.2. There we see the DCT-IV is applied on the windows shown figure b. The APLTT spectral analysis of speech signal is shown in figure d. We see the signal is a detailed represented by the APLTT spectral analysis.

In figure 13.3, we see the spectral analysis of the speech signal applying DCT-IV in figure a and applying the Fourier transform in figure b in the same figure. The APLTT spectral analysis captures more detailed of the signal than the Fourier transform based spectral analysis.

**Logarithmic Power Spectrum** In this step, first the logarithmic power spectrum of $S_j[n]$ is computed by equation (13.12) where $k = 0, 1, 2, \cdots, (N/2-1)$.

$$\vartheta_m(k) = 10\log_{10}(|\sum_{n=0}^{N-1} S_j[n]|^2) \tag{13.12}$$

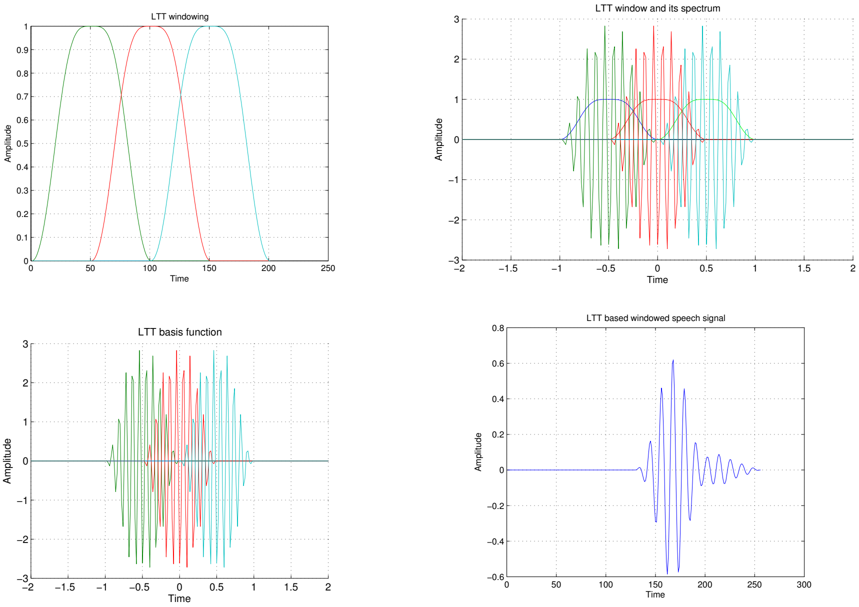


Figure 13.2: Spectral shaping and spectral analysis: LTT and FT

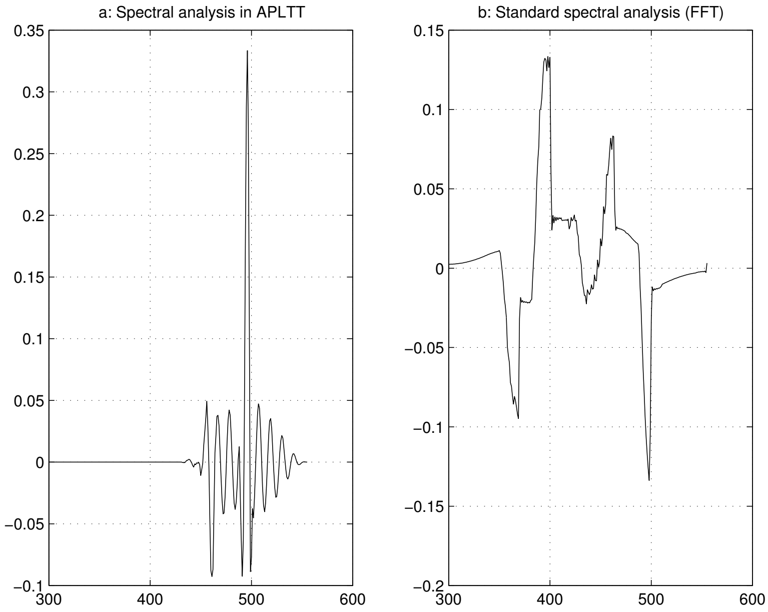


Figure 13.3: Windowed signal in APLTT and in standard feature extraction technique

### 13.2.2 Perceptual Feature Transformation

In this LTT formulation we have embedded some basic perceptual quantities such as critical bands, the scale of loudness which influences the frequency masking and the perceptual entropy. The adaptation is new to this technique.

For critical band adaptation, we first use the concept of auditory filters where the critical band is realized.

#### 13.2.2.1 Critical band for DANSR

This uses the Bark critical bank scale using equation in order to approximate the human hearing perception. This is shown in (11.13) in chapter 11.

$$H_m(k) = \begin{cases} 0 \text{ for } k < f(m-1) \\ \frac{2(k-f(m-1))}{f(m+1)-f(m-1))f(m)-f(m-1))} \text{ for } f(m-1) \leq k \leq f(m) \\ \frac{2(f(m+1)-k)}{(f(m+1)-f(m-1))(f(m+1)-f(m))} \text{ for } f(m) \leq k \leq f(m+1) \\ 0 \text{ for } k > f(m+1) \end{cases}$$

The mapping between the linear scale in Hz and the perceptual scale is done in equation (11.13) as follows. This is also written in chapter 10 in section 11.7.

$$f(m) = \frac{N}{f_s} B^{-1}\big(B(f_l) + m\frac{B(f_h) - B(f_l)}{M+1}\big)$$

Here $B(f)$ is the Erb scale in (13.13) (see chapter 11). We have choosen Erb scale because it closely approximates the cross section of the cochliear

$$\text{Erb scale:} \quad B(f) = 21.4\log_{10}(0.00437f + 1) \tag{13.13}$$

Now similar to equation (12.10) discussed in chapter 11 in section 11.7.1, we compute equation (13.14). In the equation $|H_l(k)|^{1/3}$ is a cubic root to support power law of hearing.

$$\aleph_m(l) = \sum_{k=0}^{\frac{N}{2}} |H_l(k)|^{1/3} \vartheta_m(k) \tag{13.14}$$

Now the logarithmic perceptual power spectrum is computed by equation (13.15) and the output of the $m^{th}$ filter is

$$\iota_m[l] = 10\log_{10} |\ [\aleph_m(l)]\ | \tag{13.15}$$

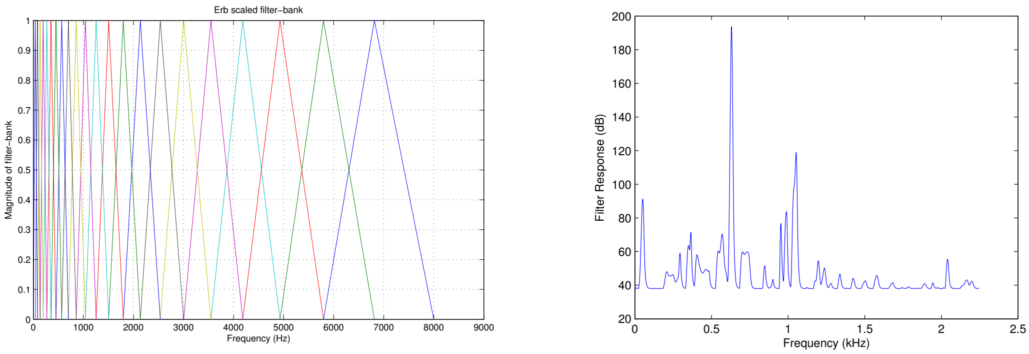


Figure 13.4: Perceptual filterbank and output of this filter bank

The output of the psycho-acoustic filter is shown in figure 13.4 where the left filter is the auditory filter-bank of the DANSR which is mapped to Erb scale and the right figure is the output of the filterbank.

#### 13.2.2.2 Intensity loudness

To formulate the intensity loudness, we have applied equation (12.20) (see chapter 12). This is explained in chapter 11 in section 11.5.1 using a basic equation given in equation (11.6). For this reason, in equation (13.14) the filter weight $|H_l(k)|$ is raised to —H$_l(k)|^{1/3}$.

## 13.3 Parametric Representation

Here we discuss the feature transformation of the perceptual spectral feature analysis. For this the perceptual entropy (PE) is selected. The PE has been used to suppress noise for the transform coding audio signal in [57]. The DCT is an example of transform coding as the DCT transforms the the signal by some linear combinations of the orthonormal local cosine bases.

### 13.3.1 Perceptual Entropy (PE)

The term perceptual entropy is used to define the smallest amount of data which is needed to encode some audio signal without any perceptual difference to the original. It indicates the average minimum number of bits in the frequency samples needed to encode a signal for this purpose [64], [58], [6]. The perceptional spectral analysis is used in [90], [105]. The perceptual spectral analysis means mainly the critical bank analysis using perceptual scales such as Bark, Mel or Erbs. The PE is defined in equation (13.16). For this definition we follow the description given in [90]. In this equation, $N$ is the number of frequency components between a certain signal frame which is used for spectral analysis using DCT and then transform this perceptual spectral analysis using Bark scale and loundness pre-emphasis given in eq in the filter-bank used $f_l$, $f_u$ where $f_l$ is the lower frequency and $f_u$ is the upper frequency limit e.g. $f_u = 20000$ Hz; $\gimel_m(k)$ is the amplitude of the frequency component and $k$ is the estimated threshold level at the frequency $T_m(k)$. This definition of the PE needs an existence of a concept of audibility and an auditory threshold. To apply PE, we have computed the perceptual power spectrum and

$$Y_m(k) = \frac{1}{N} \sum_{k=f_l}^{k=f_u} (\frac{\gimel_m(k)}{T_m(k)}) \tag{13.16}$$

It gives a lower bound estimate for the perceptual coding based on the computed mask threshold.

## 13.4 Parametric Feature Transformation

In this step the perceptual spectral features are parametrically transformed by applying IDCT- IV and the unfolding operation applying to the mathematical operations given next.

### 13.4.1 Inverse DCT-IV

The perceptual features obtained by bark critical bank spectral analysis, loudness pre-emphasis and perceptual entropy are parametrically transformed by applying the iinverse of DCT-IV shown in equation (13.17).

$$y[n] = \frac{2}{N} \sum_{n=0}^{N-1} Y_m(k) \cos[\frac{\pi}{N}[l + \frac{1}{2}][n + \frac{1}{2}]] \tag{13.17}$$

#### 13.4.1.1 Unfolding operator

Following the similar terminology as provided in equation (13.7), the unfolding operator is applied to equation (13.17) in the form of equation (13.19).

$$y[n] = \begin{cases} b_j[n]y[n] - b_j[2a_j - n]y_j[2a_j - n] & \text{if } a_j \leq n \leq (a_j + r) \\ y_j[n] \text{ if } (a_j + r) \leq n \leq a_{j+1} - r \\ b_j[n]y_j[n] + b_j[2a_{j+1} - n]y[2a_{j+1} - n] & \text{if } a_{j+1} - r \leq n \leq a_{j+1} \end{cases} \tag{13.18}$$

The ouput of this unfolded form $y[n]$ are our speech features used for classification and recognition.

The unfolding operator is now shown in equation (13.19) :

$$s[n] = \begin{cases} b_j[n]s[n] - b_j[2a_j - n]s_j[2a_j - n] & \text{if } a_j \leq n \leq (a_j + r) \\ s_j[n] \text{ if } (a_j + r) \leq n \leq a_{j+1} - r \\ b_j[n]s_j[n] + b_j[2a_{j+1} - n]s[2a_{j+1} - n] & \text{if } a_{j+1} - r \leq n \leq a_{j+1} \end{cases} \tag{13.19}$$

## 13.5 Analysis of APLTT and Standard Feature Extraction Techniques

As mentioned the APLTT is an extension of the SILTT. This has no connection with the human hearing perception. The non-stationary signal is normally decomposed in order to analyze its spectral information and to process the signal. The most common decomposition technique is the FFT transformation. This is commonly known as short time Fourier transformation (STFT). The technique is introduced in chapter 8 in section 8.1. The finite intervals are managed by applying some window function. This reduces a blocking artifact due to the decomposition by the lapped transformation discussed in [47]. This transformation has to double the blocks or the channels of the signal segment or the frame. Each segment can be processed as a finite signal by applying some window function on the frame signal.

In SILTT, the spectral analysis is obtained by the adaptive window function, folding operation and local cosine transformation (DCT-IV). The spectrally transformed speech is then used to extract the features by applying the spectral entropy using equation (13.21) preceded by IDCT-IV and the unfolding operation.

In SILTT, the spectral transformed speech is used to extract the features by computing the spectral entropy using equation (13.20) and equation (13.21). In equation (13.20), $S$ is the spectrum computed by DCT-IV and $S_k$ is the spectrum at $k$ frequency components. In equation (13.20), $S_k$ is the energy of the $k^{th}$

frequency component of the spectrum and $\breve{s}_k$ is the probability mass function (PMF) of the spectrum. Then the entropy is computed by equation (13.21).

$$\breve{s}_k = \frac{S_k}{\sum_{i=1}^{N} S_i} \quad \text{for} \quad k = 1, 2, \cdots, N \tag{13.20}$$

$$H(\breve{s}) = -\sum_{\breve{s} \in S} \breve{s}_k \log_2(\breve{s}_k) \tag{13.21}$$

The LTT is used for wheezing lung sound analysis [29], the SILTT is used for speech recognition [111], the use of LTT for the speech processing can be found in [50], audio signal analysis by using LTT is shown in [88], [75], [101], [111]. This extracts the speech features by the SILTT and uses VQ based HMM for the speech recognition.

The APLTT uses the SILTT spectral analysis and in addition this uses basic perceptual spectral analysis discussed in this chapter. These perceptual spectral speech features are used for feature classification and recognition. We consider the perceptual LTT for a GMM model based HMM recognition system.

The experimental analysis and its comparisons using other feature extraction techniques are provided in chapter 15.

# Chapter 14

# Classification and Recognition

**Outline of the Chapter** We apply a continuous Hidden Markov Model (HMM) to recognition. This uses the Gaussian mixture model (GMM) for the acoustic modeling which characterizes the probability distribution of the states. These provide latent variables. The theoretical aspects of this and its application to our recognition are provided in this chapter. First we introduce the intuitive concepts in question and then we provide the formal definitions. For the general concepts we start with HMM for an introduction into the topic. The given formal formulations of HMM model in this chapter are mainly a reflection of the literature.

The HMM given in this chapter is limited to our recognition problems. The problems are mainly finding the probability of the observations given the GMM model, finding the best path of the observations given the model, and re-estimating and training the model parameters i.e. learning. The techniques we use are forward and backward search, Viterbi search and the Baum-Welch algorithm. We also discuss the clustering. The chapter contains a combination of formal and informal elements. In the informal parts we present the main ideas for understanding the approach. For the formal elements we give mathematical formulations.

## 14.1 Formulations of HMM

The Bayes' rule is a standard method when dealing with the HMM based pattern recognition problem. In the solution approach, the HMM finds the most probable answer for the unknown states given the features and the model.

The following notations are useful:

- There the speech features are divided into some states. The states are the members of a set which is denoted by $\mathbf{Q} = \{q_1, q_2, \cdots, q_N\}$. The number of states is $N$.

- There is a set of feature vectors denoted by $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_t, \cdots, \mathbf{o}_T\}$. The features are obtained from the observations and we call them for simplicity again observations or features. The observations or features refer the same context. Each $\mathbf{o}_t$ has $n$ many feature elements and each word which may be called as an event has $T$ many feature vectors such that $t = 1, 2, \cdots, T$.
- $\lambda$ is the model parameters.

The problem is to find the most likely answer for a unknown sequence of words termed as states $\mathbf{Q}$ given the observation $\mathbf{O}$ which takes some values and a set of parameters $\lambda$ for a certain sequence of states. This is shown in equation (14.1) where $\mathbf{Q}^*$ is a given word or a sequence of states. These represent the the observation $\mathbf{O}$ and $\mathbf{Q}$ is the states for a certain word. $p(\mathbf{q}|\mathbf{o}, \lambda)$ is known as the posterior probability [130]. In equation (14.1), $\mathbf{o}$ represents the states $o_1, o_2, \cdots, o_T$ and $\mathbf{q}$ represents the states $q_1, q_2, \cdots, q_N$

$$\mathbf{Q}^* = \arg\max_{\mathbf{q}} p(\mathbf{q}|\mathbf{o}, \lambda) \tag{14.1}$$

Now we can formulate the basic equation for the whole intended process by equation (14.2) applying mainly Bayes rule. There $\mathbf{o}$ is the feature vector, $\lambda$ represents model parameters of given observations $\mathbf{O}$, $\mathbf{q}$ is the states and $p$ is the probability. In equation (14.2), $p(\mathbf{o}|\mathbf{q}, \lambda)$ is called an acoustic model. This computes the probability of the features given the states. The model looks for the most likelihood of the $\mathbf{q}$ for the features. This means the most likelihood of the feature observations given the model is obtained. Here $\mathbf{o}$ denotes any of the $\mathbf{o}_t$.

$$p(\mathbf{q}|\mathbf{o}, \lambda) = \arg\max_{\mathbf{q}} \frac{p(\mathbf{o}|\mathbf{q}, \lambda)p(\mathbf{q}, \lambda)}{p(\mathbf{o}, \lambda)} = \arg\max_{\mathbf{q}} p(\mathbf{o}|\mathbf{q}, \lambda)p(\mathbf{q}, \lambda) \tag{14.2}$$

The HMM is an extension of the Markov model. In a Markov model the probabilistic relations between the states and the features are known. In HMM, however, the relations between the states and the feature vectors are hidden. The HMM is called the double stochastic processes. Let us assume that the double stochastic processes is $\{\mathbf{O}, \mathbf{Q}\}$ where $\mathbf{O}$ is a Markov chain holding the observation transitions and $\mathbf{Q}$ is a sequence of independent random variables such that the conditional distribution of $\mathbf{Q}$ depends only on $\mathbf{O}$ [94]. Now the double stochastic processes indicate[127]:

1. $\mathbf{O}$ and $\mathbf{Q}$ are both random variable processes. These processes may have unknown probabilities. The statistics of these probabilities are continuously varying with time. By this we mean the statistics of the observation or feature vectors such as $\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_t, \cdots, \mathbf{o}_T$ are not same and they may differ from state to state that is $q_1, q_2, \cdots, \cdots, q_N$.

2. $\mathbf{Q}$ is stationary processes. These are realized by the observations $\mathbf{O}$. These are for us Gaussian mixture models (GMM). According to my knowledge the reason behind the GMM modeling is tractable mathematical manipulation for the modeling. This is a linear combination of Gaussians with some weights or mixtures. Observe that the GMM and Gaussian process do not convey the same information. Thus for modeling the feature vectors, the GMM is our general assumption.

The difference between the HMM and the Markov model is that in the HMM the state $\mathbf{q}$ that generated an observation vector $\mathbf{o}$ is hidden; we only observe $\mathbf{o}$ but we do not observe the hidden state $\mathbf{q}$ that generates the observation vectors.

In HMM, we have the transition probability $\mathbf{A}$ and the observation probability $\mathbf{B}$ for the output probability and the initial state probability $\pi$. Each state produces an output with a certain observation probability $b_j(\mathbf{o}_t)$. This is captured by an observation probability matrix $\mathbf{B}$. The states $\mathbf{Q}(q_i)$, the observations $\mathbf{O}(\mathbf{o}_t)$ are given but which state generates which observation is not given and it is hidden. Now the model $\lambda$ is formulated by equation (14.3). The elements of the model and their notations in equation (14.3) are introduced in details in section 14.2. The model parameters are mainly mean and covariance matrix of observation vectors. These are used in the Gaussian mixture model (GMM) to compute the probability density function. This is discussed in section 14.6.

$$\lambda = (\mathbf{A}, \mathbf{B}, \pi) \tag{14.3}$$

Next we introduce the HMM elements and their notations in details.

## 14.2 HMM Elements

What we directly have here is first the speech signal in discrete time via some sensor. This is here a sound recorder embedded with a microphone. These signals are compressed to feature vectors using the feature extraction technique.

For analyzing the state probabilities we now compare all the outcomes from $M$ many subspaces or subsets or events which give us the observations $\mathbf{o}$. These $\mathbf{A}$, $\mathbf{B}$ and $\pi$ are used to define the HMM. Now the HMM elements are listed:

- The matrix of the state transition probabilities is $\mathbf{A} = \{a_{ij}\}$: The communication between the states is taking place through the state transition probability matrix $\mathbf{A} = \{a_{i,j}\}$ where $i, j = 1, 2, \cdots, N$. For the matrix elements we have the equations $a_{ij} = p(\mathbf{q}_t = j|q_{t-1} = i)$ for $1 \leq i, j \leq N$ such that $a_{ij} \geq 0 \; \forall i, j$ and $\sum_{j=1}^{N} a_{ij} = 1$. For $a_{ij}$, the element of row $i$ and column $j$ of $a_{ij}$ specifies the probability of the feature vectors $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_t, \cdots, \mathbf{o}_T\}$ to go from state $i$ to state $j$. Each row of the matrix must sums up to 1.

- Clock: $t = \{1, 2, 3, \cdots, T\}$. This denotes the time index of the feature observations $\mathbf{O}$ where $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_t, \cdots, \mathbf{o}_T\}$. This means it has $T$ many feature vectors in the observation.

- $M$ events: $E = \{e_1, e_2, e_3, \cdots, e_M\}$ which give us observations. Here the number of events refer to the number of references for each command $e_m$ for $m = 1, 2, \cdots, M$. For example, if we have total $M = 100$ training samples for the command "Oeffne die Tuer", then $E = e_1, e_2, \cdots, e_M$ where $m = 1, 2, \cdots, k, \cdots, M$ and in this example $M = 100$. Then we can say $\mathbf{o}_t$ belongs to the $k^{th}$ event $e_k$.

- Observation probability matrix $\mathbf{B}$ computed from $\mathbf{b}_j(\mathbf{o}_t)$: This is the probability of the $t^{th}$ feature vector $\mathbf{o}_t$ at state $j$ for any event producing the observation. Thus for $T$ and $N$, we have a set of probable values denoted by $\mathbf{b}_j(\mathbf{o}_t)$ where $\mathbf{b}_j(\mathbf{o}_t) = p(\mathbf{o}_t|\mathbf{q}_t = j)$.

- Initial and final state: The observation $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_t\}$ is a sequence of $T$ many vectors which are observed in the states. The states $\mathbf{Q}$ has initial and final states. These are not included in the observation probabilities $b_j(\mathbf{o}_t)$. The final state has only one non-null transition that loops onto itself with a probability of 1. For the initial state we have probabilities $\pi_{\mathbf{i}}$:

$\pi_i = p[q_1 = i]$ for $i = 1$. This gives the probability for the first state $j = 1$. The value of $\pi_i$ is such that $\pi_i \geq 0$ and $\pi_i \leq 1$.

The transition probability $a_{ij}$ for $i, j = 1, 2, \cdots, N$ indicates the dynamic behavior of the feature vectors in the states. $b_j(\mathbf{o}_t)$ is the probability of the observation $\mathbf{o}_t$ given time $t$ and at state $j$. There the given stochastic process for the $\mathbf{o}_t$ will reveal the hidden stochastic process for the $b_j(o_t)$. That's why the HMM is called the double stochastic process.

The HMM has to be defined in order to use this for the pattern recognition problem.

The probability of both $\mathbf{o}$ and $\mathbf{q}$ occurring simultaneously is computed by the given equation (14.4). This is a prior probability. Here again $\mathbf{o}$ denotes any of the $\mathbf{o}_t$.

$$p(\mathbf{o}, \mathbf{q}|\lambda) = p(\mathbf{o}|\mathbf{q}, \lambda)p(\mathbf{q}|\lambda) \tag{14.4}$$

Equation (14.4) can be expanded to equation (14.5).

$$p(\mathbf{o}, \mathbf{q}|\lambda) = \pi_{q_1}.\mathbf{b}_{q_1}(\mathbf{o}_1).\mathbf{a}_{q_1 q_2}.\mathbf{b}_{q_2}(\mathbf{o}_2).\mathbf{a}_{q_2 q_3}.\cdots.\mathbf{a}_{q_{T-1} q_T}\mathbf{b}_{q_T}(\mathbf{o}_T) \tag{14.5}$$

Now the probability of the states $\mathbf{q}$ given the feature observation vector $\mathbf{o}$ is the posterior probability. This is computed by equation (14.6).

$$p(\mathbf{q}|\mathbf{o}, \lambda) = p(\mathbf{o}|\mathbf{q}, \lambda)p(\mathbf{q}|\lambda) \tag{14.6}$$

**Computations of initial $\pi_i$, initial $\mathbf{a}_{ij}$ and $b_j(\mathbf{o}_t)$:** The $\pi_i$ for $i = 1$, can be chosen randomly but this value has to be greater than 0 and less than or equal to 1. The initial $\mathbf{a}_{ij}$ can also be chosen randomly such that the $i$ th row sums to 1. The observation probability $b_j(\mathbf{o}_t)$ is computed here by the GMM.

## 14.3 Speech Aspects

The main types of variations in speech stochastic processes are the variations in the spectral composition and the variations in the time-scale [127]. In the state transition probabilities say something about the probability of the transition among the states and the variations of the duration on time-scales of the signal in each state. For example, the short or slow articulation can be expressed by self-loop transitions in the states of the model where the fast speaking or articulations can be skipped in next state connection. The state observation probabilities models the probability distributions of the spectral composition of the signal segments are associated with each state [127].

The HMM with respect to the speech recognition problem is described by figure 14.1. This figure shows how the HMM fits to the Bayes' rule in order to solve the speech recognition problem. Some explanations:

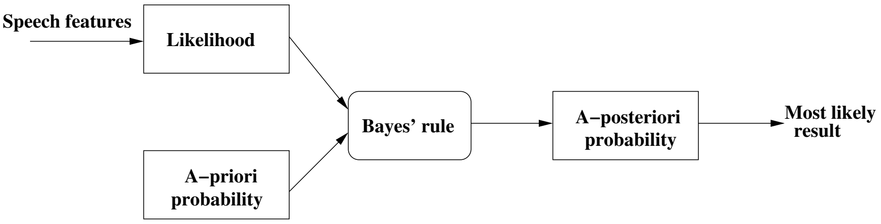


Figure 14.1: Bayes' rule in the classification and recognition problem

- Example: Suppose we have some commands in the vocabulary list : "Stop", "OeffnedasFenster", "Gehweiter". Suppose further that we have a 3 states model for the command "OeffnedasFenster". They are: i) oeffne, ii) das, and iii) Fenster and the three states are denoted by $q_1, q_2, q_3$. We have one state model denoted by $q_1$ for the command "Stop" and two states model $q_1, q_2$ for the command "Gehweiter" and "Geh" "Weiter".

- Acoustic features are obtained by the APLTT feature extraction. Each feature vector has several feature elements. Now the APLTT feature vectors for each command are denoted by $\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_T$.

- These feature vectors are used in the Gaussian mixture model to obtain the probability density function (pdf).

- Now given the features and the model i.e. the pdf of the acoustic speech features, we compute the likelihoods of the states given the features and the model.

- Likelihood: This gives us the probability of possible features given all possible states. This means for example $p(\mathbf{o}|\text{Oeffne das Fenster}_{q_1,q_2,q_3})$; $p$ denotes the probability measurement.

- Now the Bayes rule helps us to find the most likely answer i.e. what is the probability that the states belong to the given features and the model that

is
$P(\text{oeffne das Fenster}|\mathbf{o}_1, \mathbf{o}_2, \mathbf{o}_3, \cdots, \mathbf{o}_T, \lambda)$.

- A-priori probability is the computation of the pdf by the GMM.
- A-posteriori probability indicates the likelihood of the state given the features and model.
- The highest probability indicates the answer to the problem.

An example of transition between the states in the recognition stage in the Viterbi space in shown in figure 14.2. For this, we use the samples and implementation code given in [130]. This figure shows how the recognition is considered using the states following the transition path created by the feature observations. In the next section we show the HMM architecture and the computational methods to obtain the HMM parameters. We discuss this first informally. The mathematical derivations of these are discussed section 14.7.

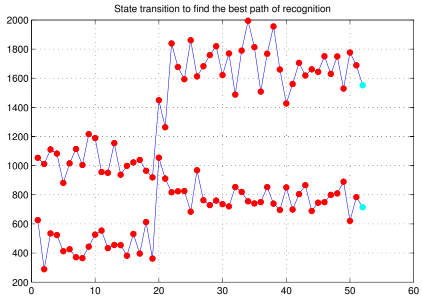


Figure 14.2: State transitions in the Viterbi search space [130]

## 14.4 Informal Discussions: HMM Architecture

In this section the definitions of the HMM for the speech recognition problem are first listed and then the approaches and their informal computation approaches for these are introduced. We have also introduced the HMM constraints for the speech recognition problem and the HMM topology.

### 14.4.1 HMM Problems and Techniques

The HMM solves a problem by its model of evaluation, searching, and re-estimating through learning the states of the observations for the optimum likelihood result. For these, given acoustic features $\mathbf{o}$ and the model $\lambda$, the HMM uses some methods.

**Evaluation** This computes the probability of the observations $p(\mathbf{o}|\lambda)$ given the model $\lambda$ and the probability of the observations of being in certain state at certain time by the forward algorithm.

**Search** A preliminary remark is that search is a method that is underlying many machine learning and optimization procedures. Here search is used to compute the optimal likelihood of the observations and a state given the mode. It tries to find the best state holding the word on the sequence of features observations at a specific time the given model. Here we use the Viterbi search algorithm.

**Learning and Re-estimation** The re-estimation adjusts the model $\lambda$ in order to maximize the probability $p(\mathbf{o}|\lambda)$ of the feature vectors $\mathbf{o} \in \mathbf{O}$. This improves the initial HMM parameters estimation. These are done by expectation maximization algorithm, in particular the Baum-Welch technique using the forward-backward algorithm.

### 14.4.2 HMM Constraints

Here we list the major constraints of HMM model. They are in general applicable for any HMM based pattern recognition problem.

- All possible states must be known prior to the system design.
- All possible connections among the states must be known.
- All the vocabularies must be known in advance in the recognition system.
- The initial probabilities and the estimate of the state observation or emission probabilities as well as state transition probabilities must be given.

### 14.4.3 HMM Topology

The HMM topology says what type of HMM model one selects for a particular pattern recognition problem. They must be defined in advance, for example:

- if the HMM is discrete or continuous;
- if the model is left-to-right or fully-connected;
- if the transition probabilities are fully defined such as in the transition probability matrix;
- if the type of HMM model is for phoneme, or words or sub-words or characters or language.

Our topology is left to right. It is shown in the next section.

## 14.5 HMM Formulations for DANSR

An overview over the structure of our approach about classification and recognition is shown in figure 14.3. There, we see how the different parts of the DANSR system interact with one another. In the figure, we refer to the elements introduced in the previous section as for instance APLTT. In this figure, the acoustic model is the GMM which computes the pdf of the features. The language model means the word in vocabulary (in the computations it is state), the search model is mainly the Viterbi algorithm.

The scenario of the situation is:

- We have a predefined list $LIST = \{w_1, w_2, \cdots, w_L\}$ of (in our application) $L = 20$ many predefined words.
- Each $w_l$ has $M = 100 + 25$ many training and testing examples. Here we have 100 for the training events or reference or examples where 100 for testing and 25 for the testing examples for each $w_l$ and we take $l = 1, 2, \cdots, M$.
- Now each $w_l$ has $N$ many states, for instance we can take $N = 5$. As a running index we have $i = 1, 2, \cdots, N$.
- Each signal (for instance a word) is segmented into signal blocks. Each state $q_l$ has feature vectors $\mathbf{o}_l$ obtained from each segmented signal block by the feature extraction technique.

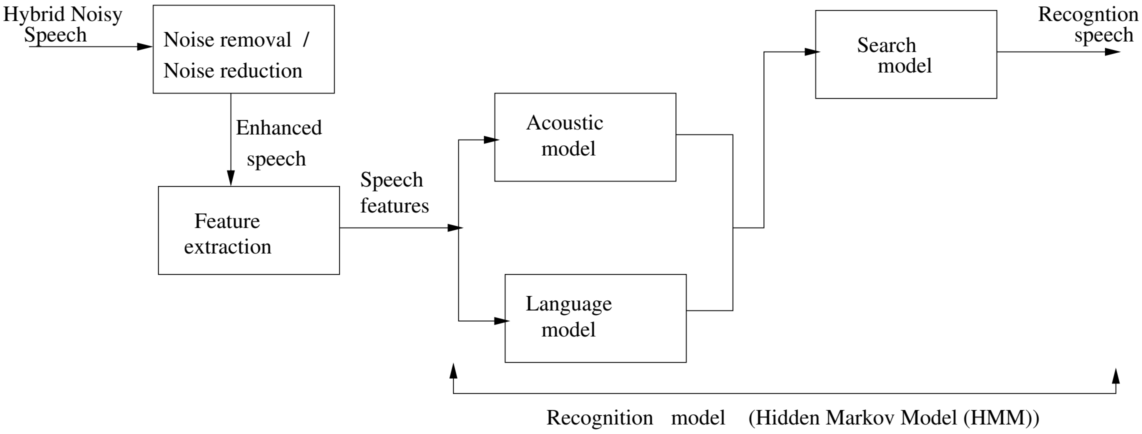


Figure 14.3: Recognition Module and its Integration with APLTT Features

Here the DANSR topology is left-to-right and continuous; mainly we have a word or sub-word model. We have characterized the feature observations by applying the GMM for the words or the sub-words for the HMM recognition. This can be seen in figure 14.4 where null indicates the beginning and end state which is 0. How we compute the observation probability of the features at certain

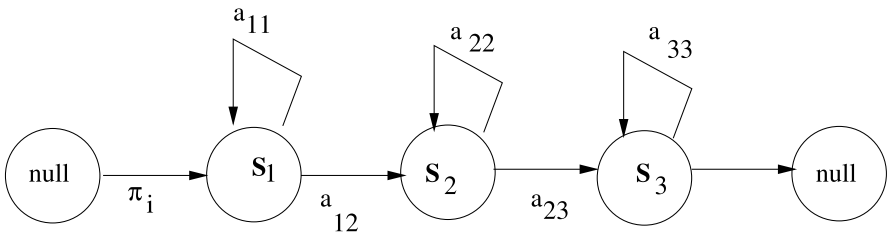


Figure 14.4: Left-right HMM topology

state and a certain event by the GMM is discussed in the next section.

## 14.6 Gaussian Mixture Model (GMM)

We have used the GMM to model the speech features. As mentioned, here the model describes the probability density functions of the feature vectors using their means and covariances. The difference between the Gaussian process and Gaussian mixture process (GMM) is that the Gaussian process is unimodal while the GMM is multimodal. The unimodal has only one peak or mode and a multimodal has more than one peak. The term ”peak” means the local maximum of the distribution. The GMM smoothly approximates an arbitrary shaped density using the mean vectors, covariance matrices of the observations and gives an insight of the process [2], [133].

The GMM is a linear combination of $G \in \mathcal{Z}$ many Gaussians. Given the feature vectors ($\mathbf{o}_t$ is a part of feature observations), each Gaussian model in the GMM has its own mean and covariance matrix as its parameters and these have to be estimated separately for each Gaussian model for the GMM. The computation of the mean and covariance matrix for Gaussian model in the GMM is not computed in a similar manner that is done for the Gaussian model because here we do not know which observation belongs to which Gaussian and which mean and covariance matrix belongs to which feature vectors which again belongs to a certain observation in the list. The GMM combines probability distributions by some weighting. These weights are unknown for a particular application and these weights have to be determined. These determinations are done by an estimation. Hence we come to an iteration process. There we can say what our initial weighs are. The initial weights are taken in such a way that they are all the same. This means for the G many Gaussian models, we have $G$ many weights and the weights are initialized as $\frac{1}{G}$. The GMM model parameters are initialized by K-means clustering discussed in section 14.8.2 but optimized by the expectation maximization (EM). The EM is an iterative process and this iteratively optimized the model parameters by using equation (14.7) where $i$ denotes the number of iteration. A detailed equation (14.7) and the EM iteration process can be found in [55]. Here we only noted how the GMM parameters are estimated by the EM iteration.

$$Q(\lambda^{(i)}, \lambda^{(i+1)}) = \sum_{t=1}^{T} \sum_{q_1, q_2, \cdots, q_N} p(\mathbf{q}|\mathbf{o}_t, \lambda^{(i)}) \log p(\mathbf{q}, \mathbf{o}_t|\lambda^{(i+1)}) \tag{14.7}$$

For the EM iteration, first the $Q$ is initialized, then the iteration is continued to estimate $Q(\lambda^{(i)}, \lambda^{(i+1)})$ and stopped when $\lambda^{(i+1)} = \text{argmax}_{\lambda^{(i+1)}} Q(\lambda^{(i)}, \lambda^{(i+1)})$. We can see the use of this in equations (14.18) and (14.22).

### 14.6.1 Computational Aspects of GMM

?? For the observation sequence $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_t, \cdots, \mathbf{o}_T\}$ for $t = 1, 2, \cdots, T$ the probability is formulated by equation (14.8) where the $c_g$ describes the mixture for the component $g = 1, 2, \cdots, G$ such that $\sum_{g=1}^{G} c_g = 1$. $\mathbf{O}$ is a $T$ dimentional feature vector such that $\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_T$ and each $\mathbf{o}_t$ vector consists of $l = 1, 2 \cdots, n$ feature elements. GMM denoted by $\lambda$ has three main parameters, the mixture components $\mathbf{c}$, the mean vector denoted by $\mu$, and the covariance matrix denoted by $\mathbf{\Sigma}$. All these three parameters have indices $g = 1, 2, \cdots, G$.

In equation (14.8), $\mathcal{N}$ is a notation used for the Gaussian mixture model and $\mathcal{N}$ contains the parameters of the Gaussian mixture model. This is shown in equation (14.9).

$$p(\mathbf{o}_t|q) = \sum_{g=1}^{G} \sum_{t=1}^{T} c_g p(\mathbf{o}_t \mid \mu_g, \mathbf{\Sigma}_g) = \sum_{g=1}^{G} \sum_{t=1}^{T} c_g \mathcal{N}(\mathbf{o}_t, \mu_g, \Sigma_g) \tag{14.8}$$

In equation (14.9), we see the model $\lambda$ for the HMM recognition task and its parameters for given observation vectors $\mathbf{o}_t$ where $g = 1, 2, \cdots, G$ and $t = 1, 2 \cdots, T$.

$$\lambda = \{c_g, \mu_g, \mathbf{\Sigma}_g | \mathbf{o}_t\} \quad \text{for} \quad \forall\, g \in G \tag{14.9}$$

We see the pdf $p(\mathbf{o}_t)$ of the feature vectors $\mathbf{o}_t$ in equation (14.10). Here $'$ indicates transpose, we avoided the conventional transpose notation $T$ for transpose to avoid confusion with $T$ that we used for $T$ dimensional observations. In the equation the dimension of each feature vector is $l$ where $t = 1, 2, \cdots, T$. The distribution is characterized by equation (14.10).

$$p(\mathbf{o}_t) = (2\pi)^{-l/2} |\Sigma|^{-\frac{1}{2}} \exp[-\frac{1}{2}(\mathbf{o}_t - \mu)' \Sigma^{-1} (\mathbf{o}_t - \mu)] \tag{14.10}$$

The mean of the feature vector $\mathbf{o}_t$ can be expressed by using equation (14.11)

$$\mu_t = E[\mathbf{o}_t] = \frac{1}{l} \sum_{n=1}^{l} o_t[n] \tag{14.11}$$

The covariance of the feature vectors $\mathbf{o}_m$ and $\mathbf{o}_n$ where $m, n = 1, 2, \cdots, t, \cdots, T$ is shown in equation (14.12) where $\mu_m$ is the mean of the feature vector $\mathbf{o}_m$. Similarly $\mu_n$ is the mean of the feature vector $\mathbf{o}_n$. $C_{m,n}$ denotes the covariance of the feature vectors $\mathbf{o}_m$ and $\mathbf{o}_n$. Similarly $C_{m,m}$ denotes the covariance of the feature vectors $\mathbf{o}_m$ and $\mathbf{o}_m$.

$$C_{m,n} = E[(\mathbf{o}_m - \mu_m)(\mathbf{o}_n - \mu_n)] \tag{14.12}$$

$$C_{m,m} = E[(\mathbf{o}_m - \mu_m)(\mathbf{o}_m - \mu_m)] = E[(\mathbf{o}_m - \mu_m)^2] = \sigma_{m,m} \tag{14.13}$$

For the GMM, it can be determined whether the covariance matrix is full or diagonal. This generally depends on the development criteria and the available data. The common approach is a diagonal covariance matrix. The effect of modeling GMM using a full covariance matrix and a diagonal covariance matrix is the same [27]. For the DANSR, the diagonal covariance matrix is considered and it is shown in eq (14.14).

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_{11}^2 & 0 & \cdots & 0 & 0 \\ 0 & \sigma_{22}^2 & \cdots & 0 & 0 \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ 0 & 0 & \cdots & 0 & \sigma_{nn}^2 \end{bmatrix} \quad \text{and} \quad \mathbf{\Sigma}^{-1} = \begin{bmatrix} \frac{1}{\sigma_{11}^2} & 0 & \cdots & 0 & 0 \\ 0 & \frac{1}{\sigma_{22}^2} & \cdots & 0 & 0 \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ 0 & 0 & \cdots & 0 & \frac{1}{\sigma_{nn}^2} \end{bmatrix} \tag{14.14}$$

Now we can see the main parameters of the GMM: $\mathbf{c}, \mu, \mathbf{\Sigma}$ for the whole feature observations that is $\{\mathbf{c} = c_1, c_2, \cdots, c_G, \mu = \mu_1, \mu_2, \cdots, \mu_G, \mathbf{\Sigma} = \Sigma_1, \Sigma_2, \cdots, \Sigma_G\}$.

Now each state has a likelihood function which is parameterized by the $G$ mixture weights, $G$ mean vectors and $G$ diagonal covariance matrices.

Now equation (14.15) is an expansion of equation (14.8).

Thus the probability density of the occurrence of the observations $\mathbf{o}_t$ the model $\lambda$ is equation (14.15). $\mid \Sigma_g \mid$ is the determinant of the covariance matrix.

$$\mathcal{N}(\mathbf{o}_t, \mu_g, \Sigma_g) = \sum_{g=1}^{G} \sum_{t=1}^{T} \{\mathbf{c_g} \frac{1}{(2\pi)^{l/2} \mid \Sigma_g \mid^{1/2}} \exp\big( - \frac{1}{2}(\mathbf{o}_t - \mu_g)' \Sigma_g^{-1} (\mathbf{o}_t - \mu_g)\big)\} \tag{14.15}$$

The connection between the HMM and the GMM is shown by eq (14.16) where the pdf of the observations is modeled by the GMM and this is the probability of the observations of an event given the state $j$.

$$\mathbf{b}_j(\mathbf{o}_t) = p[\mathbf{o}_t = \mathbf{e}_k | \mathbf{q}_t = j], \quad \text{for } 1 \leq k \leq M. \tag{14.16}$$

In equation (14.16), $\mathbf{b}_j(\mathbf{o}_t)$ of $\mathbf{o}_t$ at $t^{th}$ time instant which is the $t^{th}$ frame for the state $j$ for $j = 1, 2, \cdots, N$ is rewritten in eq (14.17).

$$\mathbf{b}_j(\mathbf{o}_t) = \sum_{g=1}^{G} c_{jg} \frac{1}{(2\pi)^{l/2} \mid \Sigma_{jg} \mid^{1/2}} \exp\big( - \frac{1}{2}(\mathbf{o}_t - \mu_{jg})' \Sigma_{jg}^{-1} (\mathbf{o}_t - \mu_{jg})\big) \tag{14.17}$$

Now the EM algorithm estimates the parameters of the GMM components iteratively in two steps: i) E-step, and ii) M-step. The E-step estimates which GMM component belongs to which observation. In the M-step, the re-estimation of the parameters of the estimation of the E-step is done to find the optimum estimation. In each iteration, the re-estimated parameters give at least as high a log-likelihood as the previous parameter values gave.

The EM iteratively computes the log-likelihood (LL) of the GMM. This measures how the model fits to the experimental data or the particular observations [70].

Now for an instance $i$ of the iteration of the EM estimation is written in equation (14.18). There $\mathcal{N}(\mathbf{o}_t, \mu_g^{(i)}, \Sigma_g^{(i)})$ denotes the value of the pdf of the $g^{th}$ GMM component at $i$ iteration, $\sum_{m=1}^{l}$ denotes the $m^{th}$ diagonal of the $l$ dimensional observation vector. $\mu_g^{(i)}$ denotes the mean of the $\mathbf{o}_t$ feature vector and $g^{th}$ component of the GMM.

$$p(\mathbf{o}_t|\lambda)^{(i)} = \frac{c_g^{(i)}\mathcal{N}(\mathbf{o}_t, \mu_g^{(i)}, \Sigma_g^{(i)})}{\sum_{m=1}^{l} c_m^{(i)}\mathcal{N}(\mathbf{o}_t, \mu_m^{(i)}, \Sigma_m^{(i)})} \tag{14.18}$$

The estimated GMM parameters for the $(i+1)^{th}$ iteration using the $(i)^{th}$ iteration is shown in equation (14.19), equation (14.20), and in equation (14.21) where $\hat{c}_g^{(i+1)}$, $\hat{\mu}_g^{(i+1)}$ and $\hat{\mathbf{\Sigma}}_g^{(i+1)}$ are the new estimated values of $(i+1)^{th}$ iteration for $c$, $\mu$ and $\Sigma$ of the $g^{th}$ component.

$$\hat{c}_g^{(i+1)} = \frac{1}{T}\sum_{t=1}^{T} p_g(\mathbf{o}_t, \lambda)^{(i)} \tag{14.19}$$

Similarly, the mean vector $\hat{\mu}_g$ is shown in eq (14.20).

$$\hat{\mu}_g^{(i+1)} = \frac{\sum_{t=1}^{T} p(\mathbf{o}_t|\lambda)^{(i)}\mathbf{o}_t}{\sum_{t=1}^{T} p_g(\mathbf{o}_t|\lambda)}^{(i)} \tag{14.20}$$

The covariance matrix $\mathbf{\Sigma}_g$ is estimated $\hat{\mathbf{\Sigma}}_g$ by eq (14.21).

$$\hat{\mathbf{\Sigma}}_g^{(i+1)} = \frac{\sum_{t=1}^{T} p_g(\mathbf{o}_t|\lambda)^{(i)}(\mathbf{o}_{m,t} - \hat{\mu}_{g,m}^{(i+1)})^2}{\sum_{t=1}^{T} p_g(\mathbf{o}_t|\lambda)^{(i)}} \tag{14.21}$$

Using $\hat{c}_g$, $\hat{\mu}_g$ and $\hat{\mathbf{\Sigma}}_{jg}$ shown in equation (14.19), equation (14.20), and in equation (14.21) we have the posterior probability $p_g(\mathbf{q}_j|\mathbf{o}, \lambda)$ for the $(i+1)$ iteration of

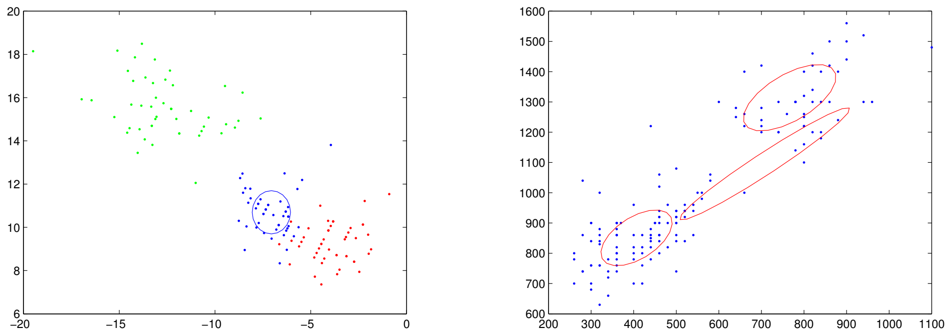


Figure 14.5: Three states are used in 3 dimensional GMM model

$g$th component in equation (14.22).

$$p_g(\mathbf{o}_t|\lambda)_g^{(i+1)} = \frac{\hat{c}_g^{(i+1)}\mathcal{N}(\mathbf{o}_t, \hat{\mu}_g^{(i+1)}, \hat{\mathbf{\Sigma}}_g^{(i+1)})}{\sum_{m=1}^{l} \hat{c}_m^{(i+1)}\mathcal{N}(\mathbf{o}_t, \hat{\mu}_m^{(i+1)}, \hat{\mathbf{\Sigma}}_m^{(i+1)})} \tag{14.22}$$

The parameters of the GMM for the EM algorithm can also be learned by some algorithms such as the K-means and the vector quantization (VQ) (see in section 14.8.3).

In figure 14.5, from left to right, we see the feature vectors are clustered and classified into one GMM, next we see the same features are clustered by 3 GMM. For the first case, less GMM components are selected and in the later case selection of GMM components more than necessary.

## 14.7 HMM Computational Approaches

Now we have the model for applying GMM. Next we use the parameters in the recognition for searching. The search takes place in a large amount of sample data that are required to train the model parameters. The amount of data we used is discussed in chapter 4. We assume $L$ many data sets for a pre-selected vocabulary $\mathbf{S}_l$ and $l = 1, 2, \cdots, L$. Each $S_l$ is an independent signal collected from the same or different speakers independently one at a time. That means each spoken command is an independent trial for the same or different vocabularies or the spoken commands. It can be said that each $S_l$ is used for feature extraction and each $S_l$ has several e.g. $K$ many feature vectors $o_k$, $k = 1, 2, \cdots, K$. Now a question is why do we need the training process? A possible answer to this question is that to maximize the model and the model parameters given the observations we need to train the model parameters. In the forward algorithm, the total probability is obtained by summing the probabilities over all possible paths to any given state. The Viterbi gives the best sequence until a particular time, but not the overall probability of being in a given state [54].

In the following section we describe the HMM problem solving techniques using the forward, Viterbi, backward and Baum-Welch algorithm.

### 14.7.1 Evaluation: Forward Algorithm

Evaluation is mainly done by forward search using the forward algorithm. In the evaluation, given the model the probability of the set of observations in a specific state sequence is estimated. Given the model, the probability of the state $\mathbf{o}_t$ being at state $i$ at time $t$ is estimated by the forward probability denoted by $\alpha$ [54]. The forward probability can in principle be computed by the recursion by using eq (14.23).

$$\alpha_t(i) = p(\mathbf{o}, \mathbf{q}|\lambda) = p(\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_t, \mathbf{q}_t = \mathbf{i}, \lambda) \tag{14.23}$$

The forward algorithm estimates the likelihood $p(\mathbf{o} \mid \lambda)$ (which means for the given model the probability of the observation at certain time at certain state) by the following three steps:

- Initialization: This uses the initial parameters to start the evaluation and this is shown in equation (14.24).

  $$\alpha_1(i) = \pi_i.\mathbf{b}_i(\mathbf{o}_1) \text{ for } 1 \leq i \leq N \tag{14.24}$$

  Substituting the initial state $i = 1$ in equation (14.24), we find equation (14.25).

  $$\alpha_1(i = 1) = 1 \tag{14.25}$$

- Induction: The probabilities of the observations from the past to the present state are computed using the previous probabilities, transition probabilities and observation probabilities in order to estimate the likelihood of the observations. The computation is shown in equation (14.26) for $t = 2, 3, \cdots, T$ and $j = 2, 3, \cdots, N$.

  $$\alpha_t(j) = \big(\sum_{i=1}^{N} \alpha_{t-1}(i).\mathbf{a}_{ij}\big).\mathbf{b}_j(\mathbf{o}_t) \quad \text{for } 1 \leq t \leq T \quad \text{for } 2 \leq j \leq N. \tag{14.26}$$

- Termination: This gives the estimate of the likelihood of the observation for a state given the model by equation (14.27) where $T$ is the length of the each features observation . This means $\mathbf{o} = \mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_t, \cdots, \mathbf{o}_T$.

$$\alpha_T(N) = p(\mathbf{o} \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{14.27}$$

### 14.7.2 Backward Algorithm

The backward search is done by the backward algorithm. This is necessary for the learning algorithm. The backward probability $\beta$ denotes the probability of the observations $\mathbf{o}_T$ through $\mathbf{o}_{t+1}$ being in state $i$ at time $t$ given a HMM model $\lambda$ by eq (14.28). Here the probability of the future sequence conditioned on the present state $j$ at time $t$ is computed.

$$\beta_t(i) = p(\mathbf{o} \mid \mathbf{q}, \lambda) = p(\mathbf{o}_{t+1}, \mathbf{o}_{t+2}, \cdots, \mathbf{o}_T \mid \mathbf{q}_t = \mathbf{i} | \lambda) \tag{14.28}$$

- Initialization : The model in state $i$ at time $T$ is 1. The transition of the observations is finished at time $T + 1$ and this is also 1 [54].

$$\beta_T(i) = 1 \quad \text{for } 1 \le i \le N. \tag{14.29}$$

- Induction: The likelihood of the observation given the model is shown in equation (14.30).

$$\beta_t(i) = \sum_{j=1}^{N} \mathbf{a}_{ij}.\mathbf{b}_j(\mathbf{o}_{t+1}).\beta_{t+1}(j) \quad \text{for } t = T-1, T-2, \cdots, 1 \quad \text{and } 1 \le i \le N. \tag{14.30}$$

- Termination: To illustrate the backward procedure, suppose $\beta_0(.)$ is the beginning state at the beginning word that emits the $\pi$ values in the transition to the first real states at time 1 [54].

$$\beta_1(1) = \sum_{j=1}^{N} \pi_j \mathbf{b}_j(\mathbf{o}_1) \beta_1(j) \tag{14.31}$$

$$\beta_1(1) = \pi_1 \mathbf{b}_1(\mathbf{o}_1)\beta_1(1) + \pi_2 \mathbf{b}_2(\mathbf{o}_1)\beta_1(2) + \pi_3 \mathbf{b}_3(\mathbf{o}_1)\beta_1(3) \tag{14.32}$$

#### 14.7.2.1 Learning: Baum-Welch Algorithm

The probability of the model given at a certain state $i$ at time $t$ is shown by equation (14.33). This is used for training the model by the Baum-Welch technique which uses the forward backward algorithm using the model parameters that are obtained from the observation sequences. The combination of the forward and backward probability is computed by (14.33). This is used to learn the model parameters from the forward and backward direction in the frame of EM algorithm. It stops learning when the likelihood is the same for both the forward and backward algorithms. For the GMM model parameters estimations, the solutions for the re-estimation formula for $\hat{\mathbf{c}}$, $\hat{\mathbf{u}}$, and $\hat{\boldsymbol{\Sigma}}$ are estimated for the observations $b_j(\mathbf{o}_t)$.

$$p(\mathbf{o}, \mathbf{q}_t = i \mid \lambda) = \alpha_t(i)\beta_t(i) \tag{14.33}$$

The Baum-Welch technique uses the following steps for training the model and model parameters of the observation sequences:

1. Compute the forward probabilities $\alpha$ by using the forward algorithm.
2. Compute the backward probabilities $\beta$ by using the backward algorithm.
3. Compute the transition probabilities $\mathbf{A}$ and the emission probabilities $\mathbf{B}$ at the current state using the observation sequences.
4. Compute the new model model parameters $\mu$, $\boldsymbol{\Sigma}$ and $\mathbf{c}$.
5. Compute the new log likelihood of the model.
6. Stop computations when there is no changes in the log-likelihood observed.

Thus the learning process of the HMM is done by re-estimating the parameters in the Baum-Welch algorithm by using the forward-backward algorithm applying the EM concept.

### 14.7.3 Searching: Viterbi Algorithm

In the Viterbi search, the best state sequence along a single path at certain time $t$, the best state and the best score are computed. The highest likelihood $\delta_t(i)$ in state $i$ at time $t$ is given in equation (14.34). The Viterbi algorithm computes the optimal state sequence $q_1, \cdots, q_{T-1}$ related to the observations with respect to their joint probability $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \mathbf{o}_3, \cdots, \mathbf{o}_T\}$ given the model.

We get for the best sequence:

$$\delta_t(i) = \max_{q_1,q_2,\cdots,q_{t-1}} p(q_1, q_2, \cdots, q_{t-1}, q_t = i, \mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_t|\lambda) \tag{14.34}$$

In general, for any value of $t$, the best score is obtained by eq (14.35).

$$\delta_t(j) = \big(\max_i \delta_{t-1}(j).\mathbf{a}_{ij}\big).b_j(\mathbf{o}_t) \tag{14.35}$$

Here the best state sequence along a single path up to time $t$ is computed by eq (14.36). This needs to keep tract of the best path up to time $t$ for each time and the state $\psi_t(j)$ which is the best state prior to state $j$ at time $t$. This can use $\psi_t(j)$ to trace back, from time $= T$ to 1, the best path.

$$\psi_t(j) = \operatorname{argmax}_{q_1,q_2,\cdots,q_{t-1}} p[q_1, q_2, \cdots, q_{t-1}, q_t = i, \mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_t|\lambda] \tag{14.36}$$

Steps in the Viterbi algorithm:

**Initialization** Equation (14.37) denotes the transition that starts from the state $\pi$ which is the initial state at the $i$ th state and it ends at state $b_i$ for $1 \leq i \leq N$ and the observation $\mathbf{o}$ at time $t = 1$. We conclude:

$$\begin{aligned} \delta_1(i) = \pi_i.b_i(\mathbf{o}_1) \quad \text{for } 1 \leq i \leq N \\ \psi_1(i) = 0 \end{aligned} \tag{14.37}$$

**Induction** In the recursion one finds the path that leads to a maximum likelihood considering the best likelihood at the previous step and the transition from it. This is then multiplied by the current likelihood given the current state and thus the best path is obtained through the induction:

$$\delta_t(i) = \max_{1 \leq i \leq N}(\delta_{t-1}(i)a_{ij}).b_j(\mathbf{o}_t) \quad \text{for } 1 \leq i \leq N \text{ and } 1 \leq j \leq N \tag{14.38}$$

$$\psi_t(j) = \operatorname{argmax}_{1 \leq i \leq N}(\delta_{t-1}(i)a_{ij}]) \quad \text{for } 1 \leq i \leq N \text{ and } 1 \leq j \leq N \tag{14.39}$$

**Termination** This finds the best likelihood when the end of the observation sequence is reached at a given final state.

$$p^* = \max_{1 \leq i \leq N}(\delta_T(i)]) \tag{14.40}$$

This finds the maximal value of $\delta_T(i)$.

$$q_T^* = \text{argmax}_{1 \leq i \leq N}(\delta_T(i)) \tag{14.41}$$

This finds the $i$ where $\delta_T(i)$ is maximal. Then the backtracking is done using backward algorithm.

**Backtracking** This finds the best sequence of states from $\psi_t$.

$$q_t^* = \psi_{t+1} q_{t+1}^* \quad \forall t = T-1, T-2, \cdots, 1 \tag{14.42}$$

The algorithm is computed in the log domain to avoid underflow errors. In the algorithm, any state can be denoted as a valid end-of-utterance state. The maximization occurs only over those states which are used in the problem as states for the valid end-of-utterance states.

## 14.8 Analysis of Standard Classification and Clustering Techniques

Here we give other standard classification and clustering techniques such as DTW, VQ, KNN. These can be used in a hybrid manner or they can be used independently for the clustering and classification.

### 14.8.1 Clustering

The clustering methods in machine learning are unsupervised. They depend on a similarity measure. The clustering is applied to multivariate data. The clustering methods can be hierarchical, partitioning or flat. Each data point is (hopefully) assigned to only one cluster. Now we consider a set of spoken words where each word is spoken several times. The unkown set of words corresponding to the same listed word is regarded as a cluster that should be generated by the clustering method. Each spoken word is now represented by feature vectors. Clustering splits a data set into different sets. The conditions for the splitting are obtained by the similarity measure. They say that the objects in one set are close together and objects in different sets are distant. If the objects are represented as points in a real space one often takes the Euclidean measure or a weighted Euclidean measure. In this case one can define the center of each cluster and the condition is now that a point belongs to a cluster (or is moved to one) where the center is closest to the point (nearest neighbor search). However, moving of objects changes the clusters and therefore one has to recompute the centers. This leads to an iteration of the clustering process.

There are two tasks that have to be done:

- Partitioning the vector space into a number of regions or clusters : This can be achieved by using vector quantization (VQ). This approach is sometimes called discrete HMM.

- Estimating the parameters of the statistical model for each cluster: This can be done by using GMM.

There are a number of clustering algorithms developed over the years. K-means is a simple method used for clustering. The GMM, VQ, and EM use the K-means algorithm to initialize the process and then recursively optimize this to find the maximum likelihood solution for a problem. The GMM is already discussed in details in section 14.6. Here first we introduce K-means clustering using the context of speech feature clustering, then VQ clustering.

### 14.8.2 K-means

The main characteristic of K-means is that the number K of clusters is fixed in advance. Each cluster set has a center and it should contain just the objects that are closest to the center. After moving the objects to the correct cluster the center has to be computed again what gives rise to an iteration. K-means minimizes the distortion for a set of vectors $o_t$ for $t = 1, 2, \cdots, T$. The objective is to find the set of centers $\mu_t$ for $k = 1, 2, \cdots, K$ that minimize the distortion. In the K-means, the squared euclidean distance is mostly used in the clustering process, while in the expectation-maximization (EM) based GMM uses a mixture resolving approach in the clustering process. The K-means has the following steps [125]:

1. Set the number of clusters K

2. Initialize the cluster centroids $\mu_1, \mu_2, \cdots, \mu_K$

3. Assign the features according to the nearest $\mu_k$

4. Recompute $(\mu_1, \mu_2, \cdots, \mu_K)$ until there is no significant changes noticed.

An additional but serious problem is the presence of different kinds of noise. The consequence is that the spoken words are corrupted dynamically. That means (among others) the observed classes can overlap to some degree. The spoken words are represented in the 2-dimensional plane. Now we refer to figure 14.6. The reference words to which we want to map the (corrupted) words are $w_1, \cdots, w_8$. The undotted lines show the true classes of the (corrupted) spoken words. The dotted lines show the classification obtained by a similarity measures. Because this measure classifies incorrectly the dotted classes are not the same as the true classes. We also see that the dotted classes can overlap as in the cases of w7 and w8.

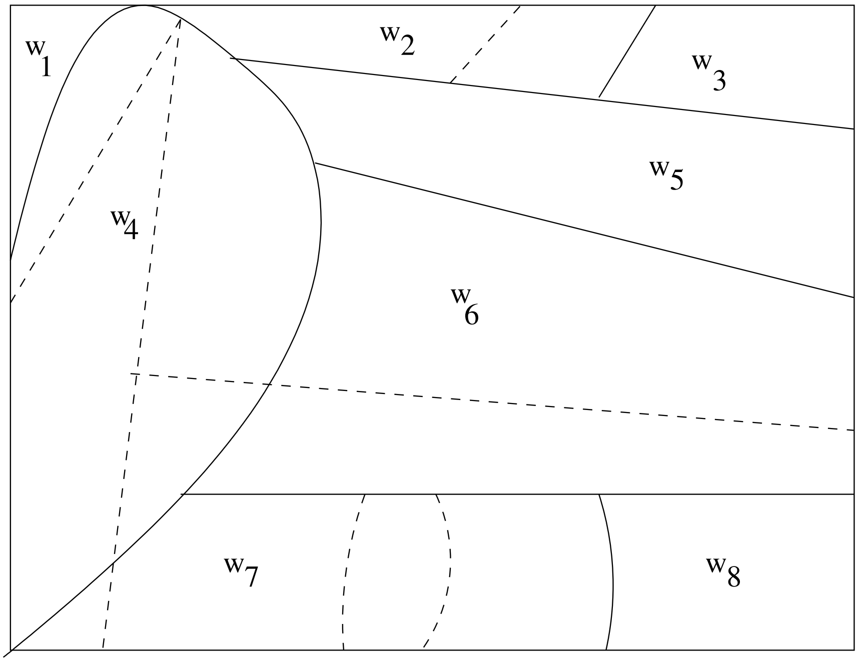


Figure 14.6: Esimated Nearest Neighbors in K-means Clustering Approaches

### 14.8.3 Clustering using VQ

One approach for estimating the parameters of the statistical model for each cluster is to divide the signal into sets of prototype vectors which are represented by their centroids. This process is again iteratively repeated for getting some effectively smaller representation of these vectors. This process is known as vector quantization (VQ). This VQ can be extended to a pdf model following a number of Gaussian mixture densities [127]. How does the assignment and the updating work using VQ? The VQ creates the clusters where the cluster weights are the ratio of points to the cluster of total points in the state. This estimates $b_j(\mathbf{o}_t)$ by computing means and covariances. $\hat{\mathbf{b}}_j(m)$ estimates $b_j(\mathbf{o}_t)$ in cluster m. We also use $\hat{\mathbf{b}}_j(m)$ for estimating numbers:

$$\hat{\mathbf{b}}_j(m) = \frac{\text{Number of vectors in cluster m and state j}}{\text{Number of vectors in state j}} \tag{14.43}$$

Then the GMM (introduced in section 14.6) is applied on the VQ based clusters to model the observations in order to obtain the most likelihood of observations that belongs to a cluster.

**Now how updating is done?** The updating of the mixture component $\mathbf{c}_j(m)$, the mean vector $\mu_j(m)$, the covariance matrix $\mathbf{\Sigma}_j(m)$, the transition matrix $\mathbf{a}_{ij}(m)$ as re-estimated update parameters is denoted by $\hat{}$ as the re-estimated mixture component $\hat{\mathbf{c}}_j(m)$, the the mean vector $\hat{\mu}_j(m)$, the covariance matrix $\hat{\mathbf{\Sigma}}_j(m)$, the transition matrix $\hat{\mathbf{A}}_j(m)(= a_{ij}(m))$ for the state $j$ and the mixture component $m$:

$$\hat{\mu}_j(m) = \text{Mean of the vectors in component m and state j} \tag{14.44}$$

$$\hat{\mathbf{\Sigma}}_j(m) = \text{Covariance matrix of the vectors in component m and state j} \tag{14.45}$$

To estimate $\mathbf{a}_{ij}$, $\mathbf{b}_j(\mathbf{o}_t)$ we use the Viterbi algorithm to segment utterances, then we re-cluster points according to the highest probability.

Then we re-estimate $\mathbf{a}_{ij}$, $\mathbf{b}_j(\mathbf{o}_t)$, repeat re-estimating $\mathbf{a}_{ij}$, $\mathbf{b}_j(\mathbf{o}_t)$ and finally the result will be obtained.

In the next section, we introduce a pattern matching method called dynamic time warping (DTW). This is frequently used in small vocabulary word recognition.

### 14.8.4 Dynamic Time Warping (DTW)

This technique originated from dealing with general temporal processes when one is searching for the nearest neighbor of a given process. For this, DTW uses mainly a similarity measurement. The DTW measures the similarity between the test and reference speech sequences and find the best match. The DTW frame is shown in figure 14.7, where the reference is in the vertical position and the test or unknown input is in the horizontal position. The optimal DTW path $D(i, j)$ is calculated from the beginning which is at left-hand side of the frame to the top right-hand side along the point $i, j$. The $i^{th}$ and $j^{th}$ entry in the DTW frame is the value of the minimum cost mapping through a cost matrix. Each point in the matrices is marked as node (it is shown in figure 14.7 by using an arrow) which can be defined as a correspondence between the respective features in the frames of the speech sequences and each node such as $i, j$ is associated with costs which can be defined as distances such as $d(i, j)$ between the respective features of the speech sequences. The cost matrix can be regarded as a weighted Euclidean distance matrix or a weighted city block distance. Euclidean distances are presented between the cepstrum coefficients of the $i^{th}$ given sequence and the $j^{th}$ reference sequence. Each row of the cost matrix specifies a vector of the cepstrum coefficients calculated during one window of the reference speech sequence, each column corresponds to a vector of cepstrum coefficients calculated during one window of the test or input speech sequence, and the entry in the cost matrix is a measure of distance between the two vectors.

$$D(i, j) = min[D(i-1, j-1), D(i-1, j), D(i, j-1)] + d(i, j) \tag{14.46}$$

The local distance is measured using the Euclidean distance presented in the equation, 14.47, where $i, j$ in $x_i$ and $y_j$ denote the speech feature coefficients in each speech segments (frame).

$$d(x, y) = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \tag{14.47}$$

The similarity (or equivalenty distance) measurement allows to compare two processes. The lengths of the two sequences were mapped to each other using zeros to make the length equal when the two sequences are of different length. The cost path in this study: horizontal $(i-1, j)$, vertical $(i, j-1)$, diagonal $(i-1, j-1)$ and the cost transitions are estimated here diagonally, or horizontally or vertically. Thus, in the DTW transitions, the three options to the next mapping are: (1) move to the next element in the first time series only, (2) move to the next element of the second time series only, or (3) move to the next element in both time series.

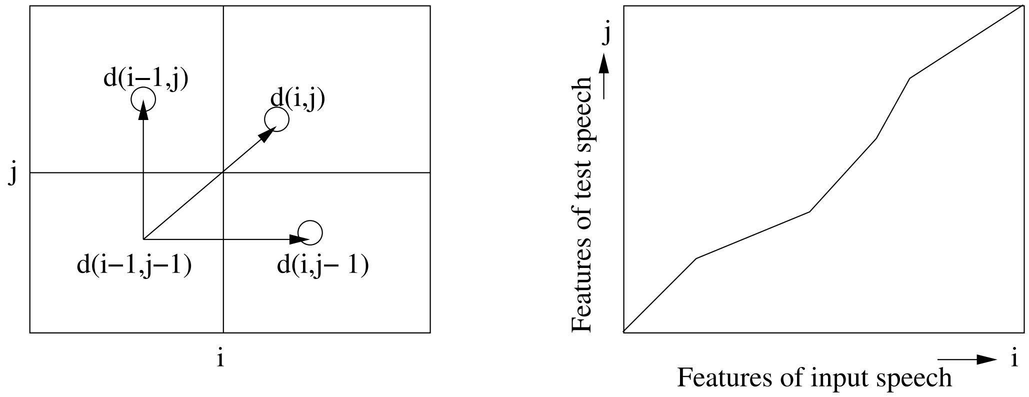


Figure 14.7: a: Computational Approaches of DTW and b: DTW alignment path in the speech features

## 14.9 Analysis and Comparison: HMM and DTW

Now we discuss how DTW differs from HMM. Both the HMM and the DTW try to find the most suitable word in the given vocabulary for a test word. However, they measure this in different ways. DTW takes the smallest distance where HMM takes the highest probability. The difference between DTW and HMM is formally expressed in equation (14.48) and in equation (14.49), [114]. In equation (14.48) and in equation (14.49), $A_v$ is word for which we want to find the most suitable word in the vocabulary. This word is denoted by $W^*$.

$$W^* = \text{argmin}_{w \in \text{vocabulary}} \text{distance}(A_v, w) \tag{14.48}$$

$$\mathbf{W}^* = \text{argmax}_{w \in \text{vocabulary}} P(w \mid A_v) \tag{14.49}$$

This shows how to apply DTW to the recognition problem in two versions.

# Chapter 15

# Remarks on Experiments

In the text we have presented many experiments. Here we add some more information. In the speech recognition the variations between the speech are main points that have to be considered. This is managed by collecting huge amount of data samples (see chapter 2). In the following sections we have shown the features extracted by using the feature extraction techniques MFCC, PLP and RASTA discussed in chapter 12 and the DANSR feature extraction technique APLTT discussed in chapter 13. The number of extracted features are 12 from each speech frame. These features are used for recognition using the GMM model based HMM recognition technique. As said, we have a list of selected spoken commands for our research. Each command in the list is repeated independently by each speaker 100 different times in the hybrid noisy industrial environment. These are our training data sets. For testing, we collected data from 25 different new people. These are our testing data sets. These data are first enhanced using our hybrid noise reduction technique, then the features are extracted using our APLTT feature extraction technique and also MFCC, PLP, RASTA features for their GMM and HMM recognition. The results of the features, perceptual analysis of the APLTT and the standard feature extraction techniques as well as the recognition performance using the APLTT and HMM, MFCC, PLP and RASTA are presented below.

## 15.1 Noisy Speech and DANSR System

In figure 15.1, in a, noisy spoken command öffne die Tür has 256 × 960 speech frames where each frames has 256 samples. In b in the same figure, the enhanced speech has 256 × 270 frames. Addition of the redundancy removal component

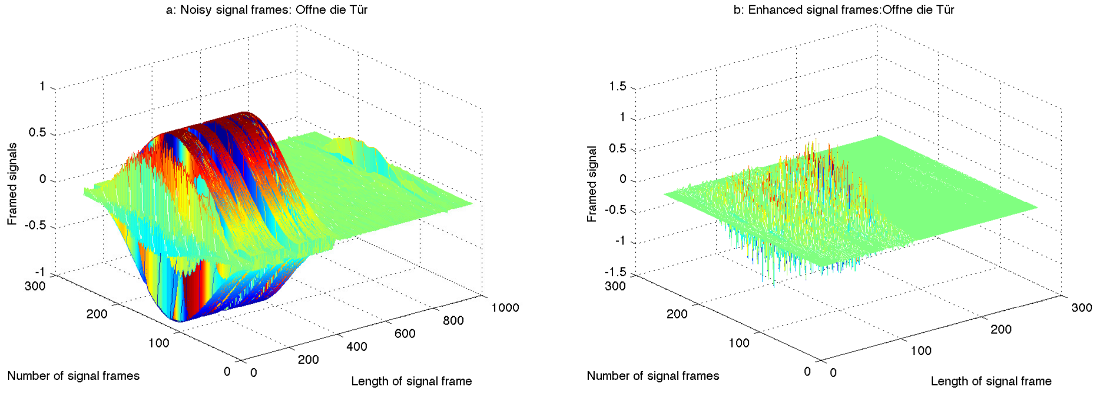


Figure 15.1: Noisy framed signal and enhanced framed features

has reduces the redundancy more than 70%. This increases the computional efficiency in the further processing.

In figure 15.2, in a, we the enhanced speech has been processed using 168 × 256 . This means the spoken command öffne die Tür has 168 frames and each frame has 256 samples. On the contrary, in b in the same figure, we see the signal has 168 frames and each frame has 12 features. Thus we can see, in the feature extraction, the signal is more compact and computational expense is reduced in a greater extent. At the same time, the features represent the originally spoken command. This can be examined followed reverse process applying some filtering [25].

## 15.2 Analysis: Feature Extraction and Features

In figure 15.3 we see the result of embedding psychoacoustic quantities in APLTT.

In figure 15.4, we see the APLTT feature variation in the classification using GMM. In figure 15.5, we see the MFCC feature using noisy and without noisy speech that is enhanced by redundancy removal, pre-emphasizing, M-band Kalman filter.

The RASTA features extraction using noisy and without noisy speech is shown in figure 15.6 in a and in b. In figure 15.7, in a, we see noisy PLP features and in b, we see the enhanced PLP speech features.

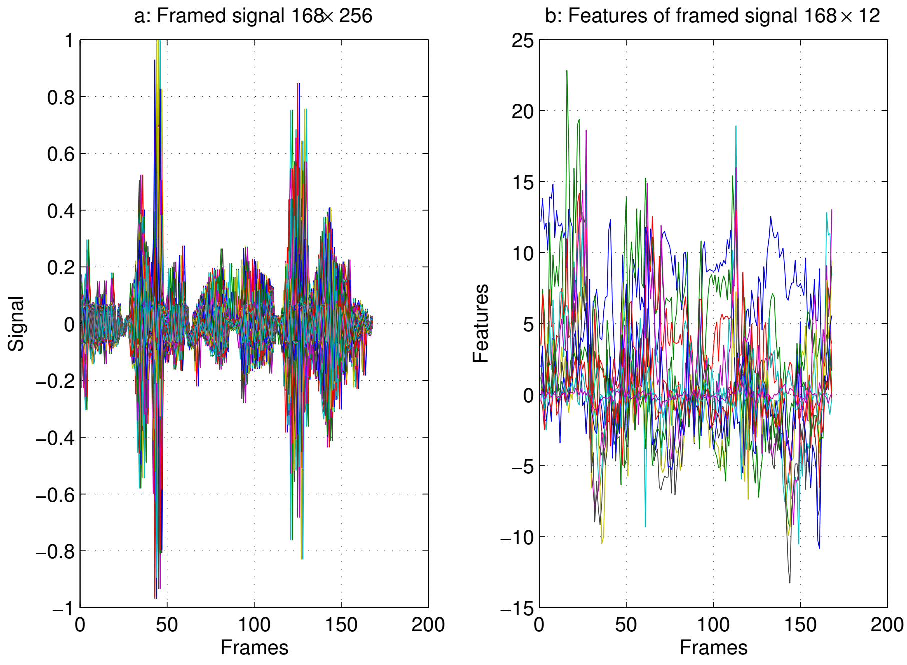


Figure 15.2: Dimensionality reduction: Framed signal and enhanced framed features

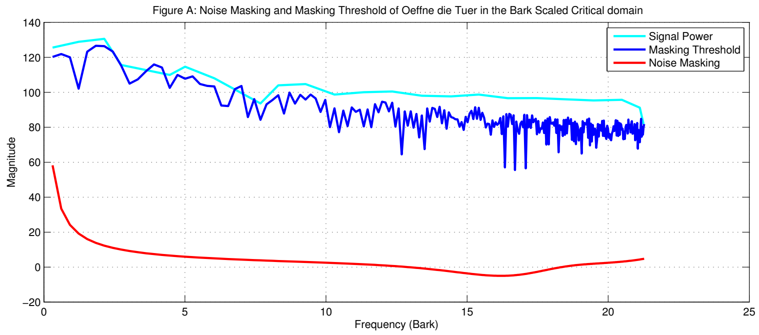


Figure 15.3: Psychoacoustic quantities embedded in APLTT and its output

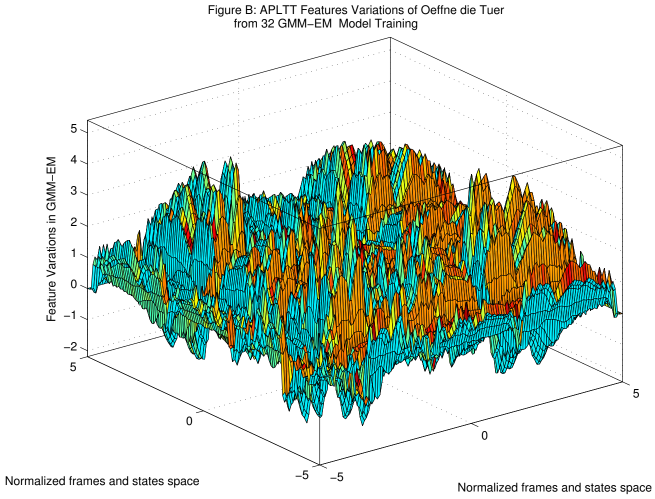


Figure 15.4: APLTT features variation using GMM

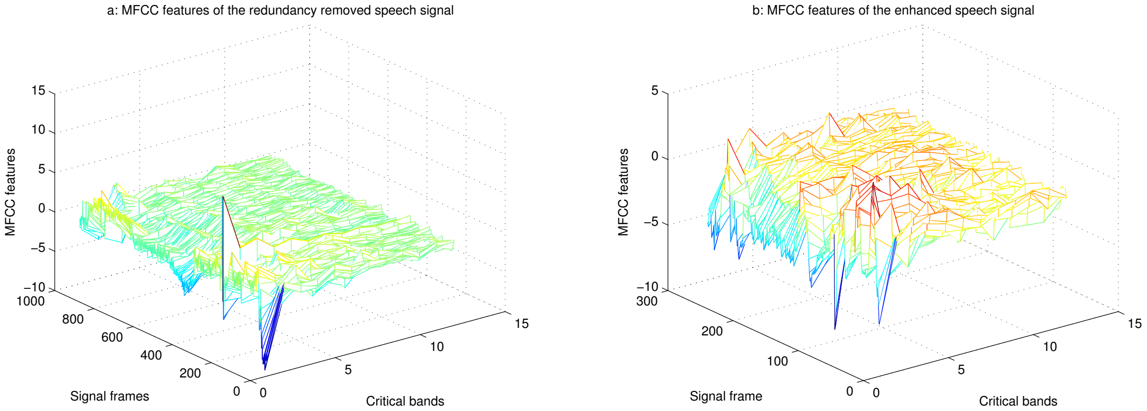


Figure 15.5: MFCC features using without noise reduction technique and with noise reduction technique

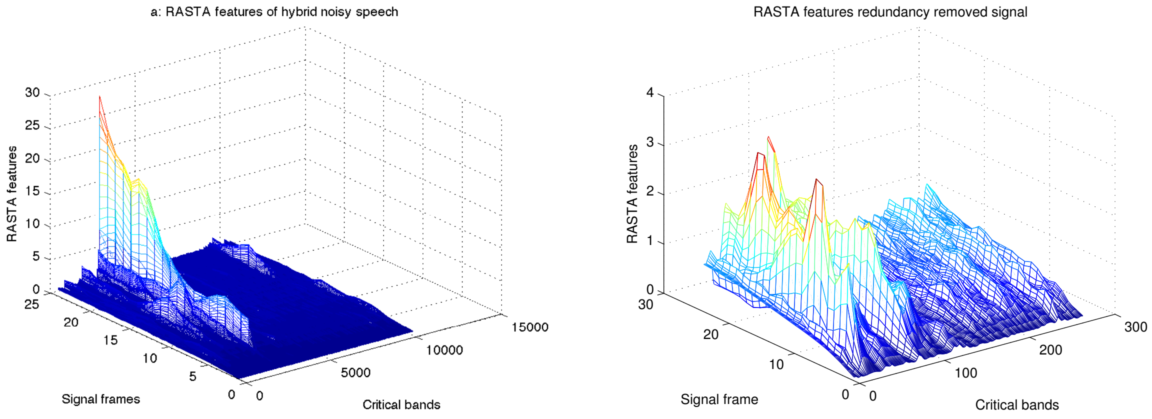


Figure 15.6: RASTA using without noise reduction technique and with noise reduction technique in a 3D plot

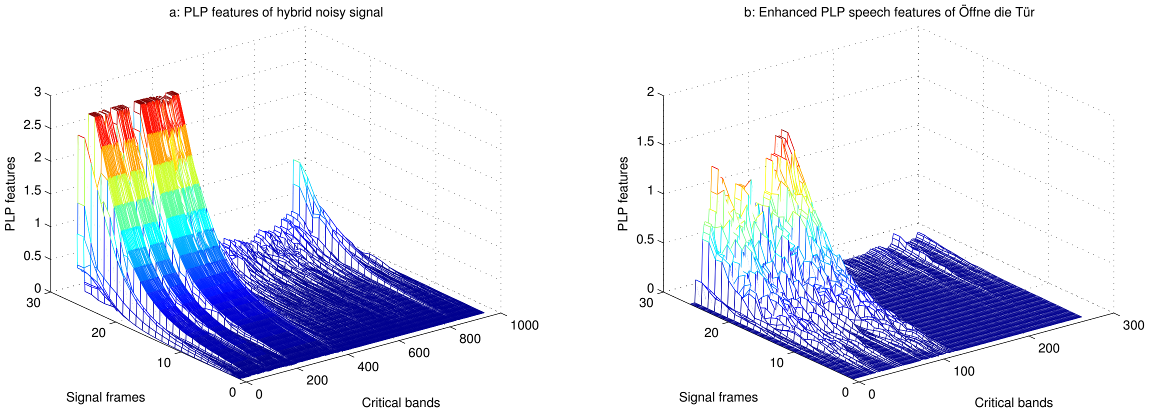


Figure 15.7: PLP using without noise reduction technique and with noise reduction technique

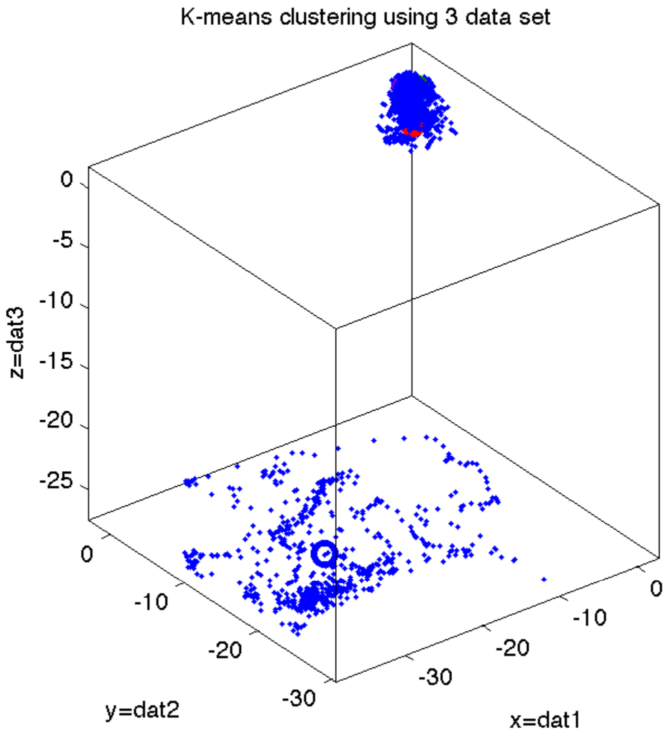


Figure 15.8: K-means clustering: 3 commands: "Öffne die Tür", "Geh weiter", "Öffine das Fenster"

## 15.3 Clustering, Classification and Recognition

Here we show clustering, classification and K-means and GMM.

**K-means** Figure 15.8 shown a clustering of using 3 data set: Öffne die Tür", "Geh weiter", "Öffine das Fenster". In this experimental demo, we have used 3 command sets in order to visualize the clustering. The feature sets in each command is 8340. Öffne die Tür" and "Öffine das Fenster". These two commands have some letters in words are common. The centers of the commands are not well seen but the centers of the clusters are computed and spotted. K-means by itself is not robust clustering approach because the center at the first place is taken in a random manner.

In figure 15.9, we see an example of features and 3 dimensional GMM in a scattered form.

Examples of some commercial speech recognition software are dragon, IBM

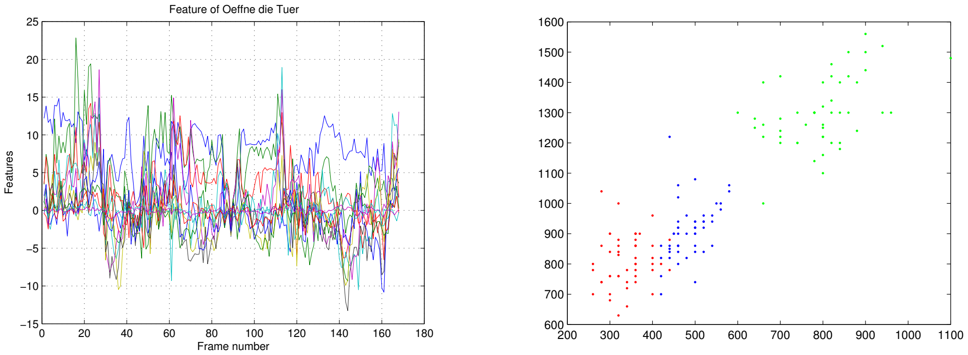


Figure 15.9: Three states are used in 3 dimensional GMM model

via Voice, Phillip speech recognition system. They deal mainly with clean speech. The probabilistic approach for the ASR are described in texts such as [80], [134], [68], [147], [11]. A number of practical applications using the probabilistic approach including the probabilistic ASR theoretical approaches are discussed in [144], [97], [48], [149].

The comparison and evaluation of the DANSR system with the commercial software such as dragon, IBM via Voice remain yet our future task. We focus in the thesis on the core development of the speech recognition algorithm that could recognize the spoken commands on the hybrid noisy environment. Therefore we first focus on enhancing the speech using hybrid noise reduction and removal techniques, then we extracted robust features using our newly developed feature extraction technique APLTT, these features modeled by the GMM for their recognition using the HMM pattern recognition tool. This whole approach is a new methodological development in the speech recognition research. But much more training, testing and evaluation have to be done in order to make this newly developed methodological approach reliable. At this stage the performance analysis of the DANSR system is about the same and in some cases it shows a little better performance than the standard developed speech recognition techniques.

A rough estimate of DANSR recognition using a small number of data set listed below in table 15.1 tells us DANSR's current status. At this development stage using a small amount of data it is performing about the same or better than other existing methods which are developed over the decades. The system requires more resources, more training and more testing in order to provide a reliable conclusion.

Table 15.1: Experimental estimate: Analysis and recognition result of small commands

| Spoken Commands | MFCC | PLP | APLTT | RASTA |
|---|---|---|---|---|
| Mach das | 20 | 18 | 22 | 22 |
| Geh vorwaerts | 21 | 22 | 23 | 21 |
| Geh weiter | 21 | 20 | 22 | 23 |
| Komm hier | 20 | 21 | 23 | 20 |
| Bleib hier | 21 | 20 | 23 | 22 |
| Stop | 20 | 23 | 22 | 20 |
| Halte es | 22 | 20 | 23 | 21 |
| Mach weiter | 23 | 22 | 24 | 22 |
| Mach es | 20 | 21 | 22 | 22 |
| Bewege dich nicht | 22 | 22 | 23 | 22 |

# Chapter 16

# Conclusions

This thesis investigated a very complex but often occurring noise recognition problem. The problem considers situations where different kinds of noise occur simultaneously. We termed these noise kinds as mild, steady-unsteady and strong. The first two kinds have been individually considered, the last kind alsmost never. However, in all these treatments there was no systematic approach that such noises happen together in a situation. To approach this problem we faced a basic difficulty: It is not sufficient to deal with just the specific noises. Each treatment will affect the whole speech system and may violate the assumptions for removing other noises. For this reason we developed an integrated system dealing with these diifficulties. As a final result we obtained the system DANSR (Dynamic Automatic Noise Speech Recognition) to deal with such situations that we called hybrid noise problems.

In order to achieve this, we reorganized the whole recognition approach as a new methodological approach by using existing methods, adding new methods and modifying existing methods. In a summary, we mention first the main achievenments:

- A practical system DANSR to deal with the problems in a real application
- A general structural approach to deal with such hybrid noisy situations.

## 16.1 Practical Results

Our applicatiion scenario is a factory room or a lab where one has noisy machines and working persons. In addition, heavy objects may fall down and generate strong noise. In this scenario persons want to give spoken commands to machines using a single microphone that are executed automatically.The commands are not arbitrary, we had a fixed list shown in the Appendix. Thus, we had a hybrid recognition problem. Our system DANSR was able to realize the recognition task. We had three example scenarios in which we tested DANSR. Our system presented a hybrid solution integrated in a single system.

## 16.2 Structural Results

Our basic view is probabilistic what is today the state of the art. For realizing the recognition task we needed to analyze the recognition from the very beginning on. We started with a new noisy speech pre-emphasizing approach. Then we focused on enhancing the speech, putting features and feature extractions in the the center of our interest. Here, a new perceptual feature extraction approach was given. Essentially, an existent adaptive local trigonometric transformation (LTT) mathematical tool is extended for a model based speech recognition system. We introduced a new approach for dealing with strong noise where we modeled it by a Poisson distribution and its treatment by matched filter. These techniques have also been used to model the psychoacoustic quantities. The last step is concerned with classification and the recognition itself. In principle, we have an HMM based speech recognition system. There we have applied the Gaussian mixture model (GMM) to model APLTT features for classification.

## 16.3 Future Extensions

The main restrictions of our approach are due to the limited possibilities for training for the recognition task. For broader applications, we would need many thousands of training examples. There, we probably need to have a new look at features and their extractions. Finally, I think that more experiments should be done for system's reliability.

# Appendix

This appendix is concerned with signal processing. We present concepts and terms we used in the thesis mainly without detailed explanations.Some of the terms are also slightly different used in the literature. This appendix is intended to support readers.

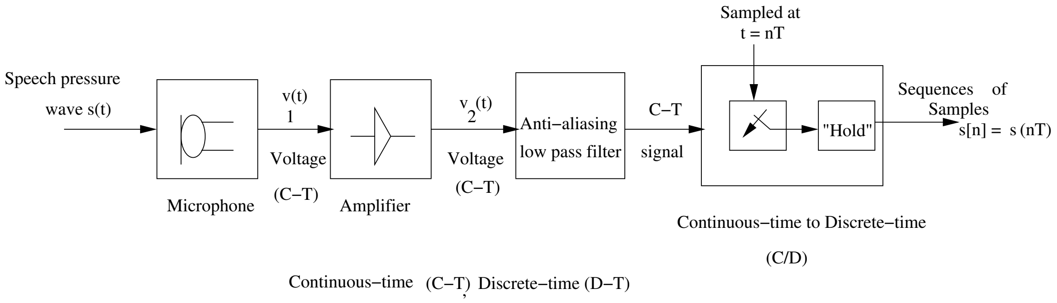


Figure 16.1: From continuous-time speech signal to discrete-time speech signal representation

**Transformation of continuous time signal to discrete time signal** Here we give an overview over the whole process about the transformation of continuous time signal to discrete time signal in figure 16.1. The steps are shown in figure 16.1 from left to right. In the figure 16.1, speech pressure waveform $s(t)$ is captured by a microphone. There the waveform is undergone some operations such as amplification, anti-aliasing low pass filtered, sample and hold technique in order to be transformed to a discrete-time signal $s[n]$. The $C/D$ converter that generates the discrete-time signal is characterized by an infinite amplitude precision. Therefore, even if the signal $s[n]$ is discrete in time, it is continuous in amplitude. A physical device does not have this infinite precision property in practice. Therefore, for the approximation to a $C/D$ converter, an analog-to-digital($A/D$) converter quantizes each amplitude to a finite set of values closest to the actual analog signal amplitude. The resulting digital signal is then discrete in time and amplitude. Associated with this are discrete-time signal systems whose input and output are sequences [131]. This is shown in figure 16.1. The purpose of sample is to hold the analog value steady for a short time while the converter or other following systems performs some operation that takes a little time.

**Spectrum** The representation of the digital signal in terms of its frequency component in a frequency domain is called the signal spectrum [84].

**Discrete Fourier Transform (DFT)** The DFT is a common tool that is used to establish a relationship between the time domain representation and the frequency domain representation. For example if $s[n[$ is a time domain signal at time instant $n$ where $n \in \mathcal{Z}$, its frequency representation is $S(k)$ of $s[n]$ using DFT is given in equation (16.1). There we considered the signal is $N$ finite length signal. In the equation, the frequencies are $\frac{2\pi}{N}k$ for $k = 0, 1, 2, \cdots, N-1$. $N$ represents the number of points that are equally spaced in the interval of 0 to $2\pi$ on the unit circle in the z-plane.

$$S(k) = \sum_{n=0}^{N-1} s[n]e^{\frac{-j2\pi kn}{N}} \tag{16.1}$$

The inverse DFT (IDFT) of $S(k)$ is $s[n]$ for $n = 0, 1, 2, \cdots, N-1$ is given in equation (16.2).

$$s[n] = 1/N \sum_{k=0}^{N-1} S(k)e^{\frac{j2\pi kn}{N}} \tag{16.2}$$

**Z-transform** The z-transform of al sequence $s[n]$ is $S(z)$ given in equation (16.3). There $z$ is a complex variable and in equation (16.3), it is considered the $s[n] = 0$ for $n < 0$.

$$S(z) = \sum_{n=0}^{\infty} s[n]z^{-n} \tag{16.3}$$

**Cut-off frequency:** It is a frequency that is characterizing a boundary between a passband and a stopband

**Bandwidth(BW)** The BW is the range of the signal's frequency that is a measure of the width of a range of the frequencies, measured normally in Hertz

(Hz).

**Sampling theorem:** A bandlimited signal with bandwidth(BW) = B Hz can be obtained by its samples as long as the sampling rate is $F_s \geq 2B$. This is also known as Nyquist theorem.

**Aliasing effect:** If samples are not taken fast enough, but a signal bandlimited to B Hz is not sampled faster than 2B times samples/second, then there is an overlapping of the samples and an error-free reconstruction is not possible. The aliasing fact displays a wrong frequency information about the signal.

**Nyquist rate and possibilities:** Sampling at the Nyquist rate is called the critical sampling. If the sampling rate is faster than the Nyquist rate, the the sampling rate is called the oversampling.

**Antialiasing Phenomena:** Antialising is a summary of methods that reduce the alias effect. It is standard to use a low pass filter before sampling. Antialiasing is to remove frequency components that would otherwise alias. As a prevention, a low pass filter is applied to obtain band limited signal during analog to digital conversion (ADC).

**Impulse Signal** The impulse signal denoted by $\delta[n]$ is defined in equation (16.4). This impulse signal is physically realizable signal and this is frequently used for designing a filter as well as for understanding a filter response.

$$\delta[n] = \begin{cases} 0 & \text{for } n \neq 0 \\ 1 & \text{for } n = 0 \end{cases} \tag{16.4}$$

In equation (16.5) we can see an abstract definition of dirac delta function. It can be thought as very thin and tall with a unit area located at the origin.

$$\begin{cases} \int_{-\infty}^{\infty} \delta(t) = 1 & \text{for } t = 0 \\ 0 & \text{for } \neq 0 \end{cases} \tag{16.5}$$

**Impulse Response** The impulse response is the response of a filter to $\delta[n]$.

**Convolution Sum** If an arbitrary input signal $x[n]$ is expressed as a sum of weighted impulses according to equation (16.6) then the response $y[n]$ can be described as in equation (16.7).

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \tag{16.6}$$

In equation (16.7), $h[n-k]$ is the system response to the delayed unit impulse sample $\delta[n-k]$ where $n$ is the time index and $k$ is the parameter that indicates the location of the input unit impulse and $h[n-k] = \mathfrak{T}\{\delta[n-k]\}$. Here $\mathfrak{T}$ is a linear time invariant (LTI) operator. $h[n]$ is an LTI system.

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]\mathfrak{T}\{\delta[n-k]\} = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = \sum_{k=-\infty}^{\infty} x(k)h(n,k) \tag{16.7}$$

The response $y[n]$ of the LTI system is a function of the input signal $x[n]$ and its delayed impulse response $h[n]$ by equation (16.7) which is known as the convolution sum [59].

**Polyphase Representation** In a naive sense, the polyphase representation means if a FIR structure $H(z)$ can be expressed as a sum of $M$ terms. Polyphase means that there are several bands for instance in this explanation we have $M$ many bands. For example, $H(z)$ is a polyphase structured FIR filter if it can be expressed as a sum of two terms, with one term containing the even indexed co-efficients and the other containing the odd-indexed coefficients.

**Strict Sense Stationary Random Process (SSS)** A random process $\mathbf{x}(m)$ is stationary in a strict sense if all its distributions and statistical parameters are time-invariant.

**Wide Sense Stationary Random Process (WSS)** A process is said to be wide sense stationary process if the mean and the autocorrelation functions of the process are time invariant.

**Spectral Temporal Resolution** This is determined by the window size, overlapping size, and the dimension of the feature vector dimension [127].

**Statistical Model Evaluation** The statistical model resolution is determined by the number of models, the number of states per model, and the numer of sub-state models per state [127].

**Tone** A pure sinusoidal signal with known frequency and timing can be termed as pure tone, a complex tone can be defined when fundamental frequency happened to be in a periodic manner or it is mixed with different frequency.

**ATH, DL and JND** The critical band is the point at which thresholds no longer increase. Absolute threshold is a minimum audible signal. Differential threshold (DL) or just noticeable difference (jnd) is just minimum perceptible change.

**DANSR Wordlist**

1. Geh weiter
2. Geh vorwaerts
3. Biege links ab
4. Biege rechts ab
5. Bring mir das
6. Geh an dem Fenster vorbei
7. Geh an den Tuer vorbei
8. Stop
9. Schliesse das
10. Schliesse die Tuer
11. Schliesse das Fenster
12. Oeffne es.
13. Oeffne die Tuer
14. Oeffne das Fenster

15. Komm her

16. Mach weiter

17. Halte es

18. Bleib hier

19. Heb es auf

20. Mach es jezt

# Abstrakt in Deutsch

In diesem Abstrakt geben wir die Thematik und die Ziele der Dissertation an und schildern in einem Ueberblick die wesentlichen Inhalte. Wir starten mit allgemeinen Inhalten, die sich in der Arbeit an verschiedenen geeigneten Stellen finden.

## 16.4 Der Rahmen

Die folgenden Veroeffentlichungen die in engem Zusammenhang zu dieser Dissertation stehen sind in der Zwischenzeit publiziert worden:

- Sheuli Paul, Michael Richter, Steven Liu, Hybrid solution to single- channel hybrid noisy speech for an industrial environment. ISSPIT,IEEE, Vietnam, Ho Chi Minh City (2012).

- Sheuli Paul, Michael Richter, A dynamic automatic noisy speech recognition (dansr) system for a single-channel hybrid noisy industrial environment.Vol. 165. ICA, Canada, Montreal (2013)

- Sheuli Paul, Michael Richter, Human Speech Recognition (SR) and A New Feature Extraction Technique for SR Systems. MDLM and MDA, New York 2013.

- Sheuli Paul, Michael Richter, Volker Michel. Reaction to Hybrid Noise in Communication. Noise Pollution: Sources, Effects on Workplace Productivity and Health Implications. Invited Book Chapter, Nova publication, March, 2014, NY, USA.

Alle sonst verwendeten Quellen wurden nach bestem Wissen angegeben.

## 16.5 Die allgemeine Thematik

In dieser Arbeit behandeln wir Themen der automatischen Erkennung gesprochener Sprache. Generell ist dies ein seit Jahrzehnten vielfach behandeltes Thema. Dabei hat es sehr interessante Erfolge gegeben. Ein Beispiel ist die Taetigkeit in einem Anwaltsbuero, wenn eine Person einen Text in ein Mikrophon spricht und dieser Text dann automatisch in Schriftform uebersetzt wird. Die Erkennungsrate lag hier oft bei 100 Prozent. Diese mannigfachen Erfolge bearbeiten jedoch stets Aufgaben die ganz bestimmten Beschraenkungen unterworfen sind. Fuer diese Arbeit sind zwei Restriktionen von zentralem Interesse:

- Die gesprochene Sprache geschah in einer "sauberen" Umgebung. Das heisst, stoerende Gerausche waren nicht zugelassen. Beim Sprechen in das Mikrofon hatte dieses direkt vor dem Mund des Sprechers zu sein.

- Es wurde nur der Wortlaut als solcher erkannt. Elemente der Sprache, die den Sinn der Worte durchaus veraendern koennen wie Pausen oder Betonungen wurden nicht beruecksichtigt. Durch die Art des Sprechens kann der Sinn der Aussage am Ende sogar ins Gegenteil verkehrt werden.

Es gab durchaus verschiedene Ansaetze, Rauschen in der Umgebung zu integrieren. Es war jedoch in der Regel so, dass nur bestimmte Geraeusche zugelassen wurden. Diese Geraeusche kamen mehr oder weniger aus einer bestimmten Quelle und nicht aus einem heterogen Umfeld mit sehr unterschiedlichen Quellen. Die Art des Sprechens faellt in den Bereich der psychoakustischen Signale. Dies steckt meist immer noch in den Anfaengen.

## 16.6 Der allgemeine Ansatz

Um die genannten Probleme in den Griff zu bekommen, haben wir den Grundansatz der Spracherkennung aufgerollt. Dies fuehrt zu einem neuen Ansatz, der insbesondere ein einheitliches Vorgehen fuer unterschiedliche Fragestellungen erlaubt. Dieser Ansatz benutzt im einzelnen meist bekannte Dinge. Diese musssen jedoch nach ihrer Selektion entsprechend modifiziert und zu einem Gesamtsystem integriert werden.

Der Ausgangspunkt ist, das gesprochene Sprache Toene produziert. Toene kommen aber auch aus verschiedenen Geraeuschquellen. Gesprochene Sprache wird im Koerper des Sprechers hergestellt und diese Produktion wurde als erstes analysiert, wobei im wesentlichen auf vorhandene Arbeiten zurckgegriffen wurde. Dazu modellieren wir die beteiligten Koerperteile als eine Maschine. Diese erzeugt Schallwellen die an das Ohr des Empfaengers kommen. Dort werden die Schallwellen digitalisiert und mathematisch verarbeitet und dann dem Gehirn zum Verstaendnis zugeleitet. Doe folgenden Abbildungen zeigen dies. Die erste Abbildung zeigt die rein menschlicheVerarbeitung. Im zweiten Bild sehen wir, was eine Maschine ersetzen soll. Dabei wird deutlich, um welche komplexe Aufgabe es sich handelt. Die dritte Abbildung zeigt unser gesamtes Szenario inklusive der verschiedenen Arten des Rauschens.

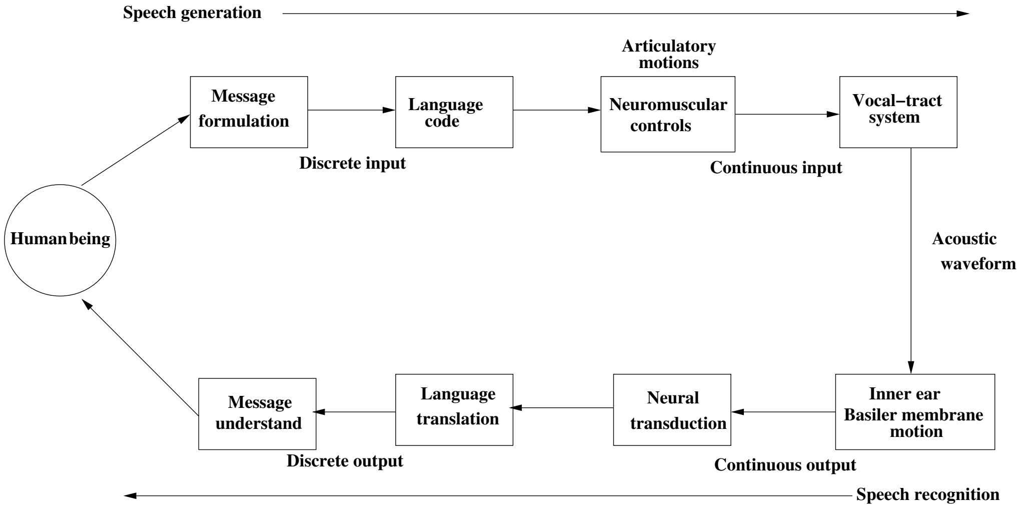


Figure 16.2: Speech Generation and Speech Recognition

### 16.6.1 Rauschen

Ein gesprochenes Wort kann mehr oder weniger verrauscht sein. Ein erstes Rauschen kommt vom Sprechen selbst und wird im wesentlichen vom Kehlkopf erzeugt. In dieser Arbeit interessieren wir uns aber mehr fuer von der Umgebung erzeugtes Rauschen. Diese Umgebung wird fuer uns durch eine Arbeitshalle in einer Fabrik geliefert. Das Ziel ist, hier Kommandos zu sprechen, die dann automatisch erkannt und anschiessend automatisch ausgfuehrt werden. Das Rauschen ist von einer hybriden Natur mit unerschiedlichen Quellen. Im wesentlichen unterscheiden wir drei Arten von Rauschen:

- Leichtes Rauschen.

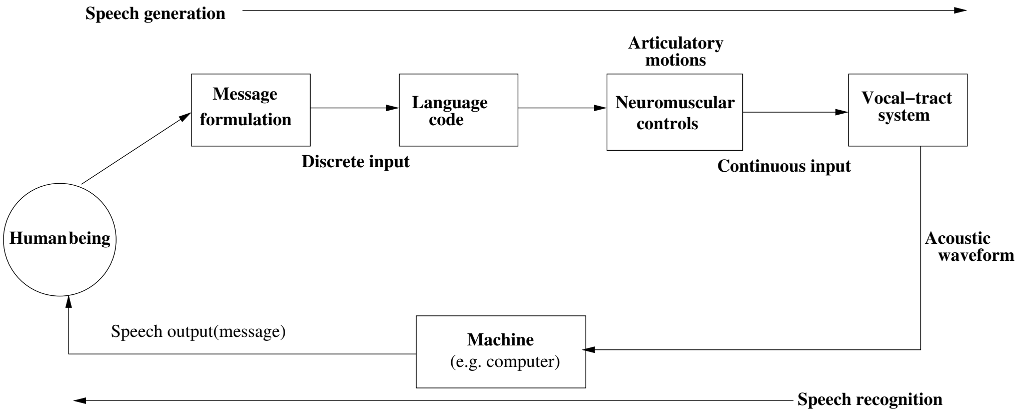


Figure 16.3: Speech Generation and Machine for the Speech Recognition

- Zeitveraenderliches dynamisches Rauschen.
- Starkes Rauschen. Dieses ist sehr laut aber kurzzeitig. Es ist unregelmaessig und wird durch eine Poissonverteilung beschrieben.

Das starke Rauschen hat insofern eine besondere Bedeutung als es alles andere ubertoent, man kann gar nichts mehr verstehen. Da es nur kurz und unregelmaessig erfolgt sind es fuer uns Ausreisser. Diese muessen identifiziert werden und es hat gegebenenfalls eine Nachfrage auf Wiederhlung zu erfolgen.

### 16.6.2 Features

Ein gesprochenes Wort erzeugt zu viele Signale um sie alle kombinatorisch verarbeiten zu koennen; dies wird auch durch Segmentierungen nicht behoben. Die Methode der Kompression besteht in der Erzeugung von Feature Vektoren. Dies sind reellwertige Vektoren. Es sind relativ wenige und sie haben eine kurze Laenge. Die Featurevektoren sind nun das einzige was zur weiteren Behandlung uebrig bleibt und deshalb muessen sie alle wichtigen Informationen beinhalten. Hier fragt es sich, warum es ueberhaupt moeglich sein kann, so viele Signale auf so wenige Features zu komprimieren. Eine Antwort dazu liefern die unterschiedlichen Datenstrukturen: Signale sind binaer, Features aber reellwertig. Es gibt nun eine ganze Reihe von solchen Features die fuer verschiedene Zwecke gedacht sind. Wir orientieren uns an den MFCC, SILTT, LPC und LPPC Features. Die Extraktion von Features ist ein zentraler Punkt der Arbeit und nimmt einen breiten Raum ein.

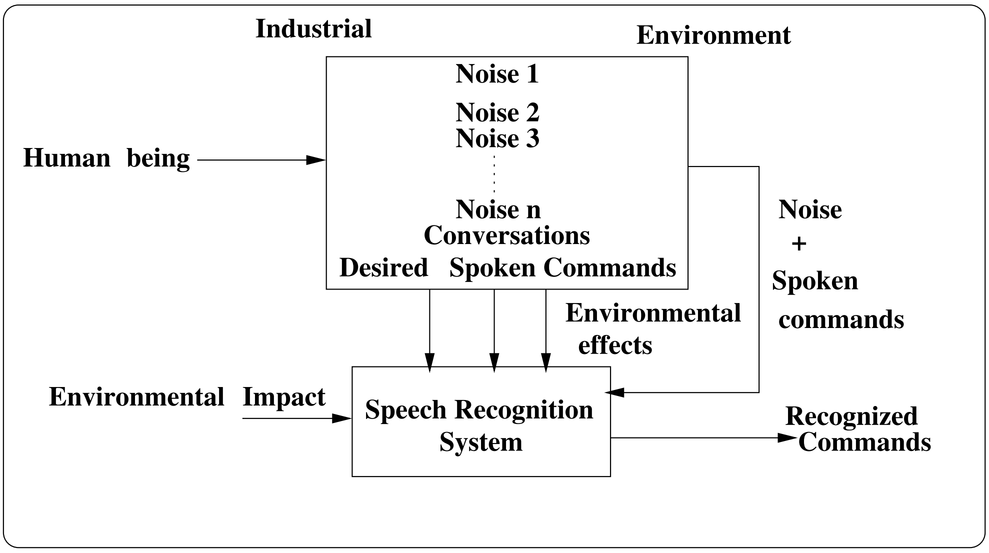


Figure 16.4: Hybrid Noise and Industrial Environment

## 16.7 Mein System DANSR

Mein System ist hauptsaechlich auf die Behandlung von Rauschen und darueber hinaus auch auf Psychoakustik ausrichtet. Aus diesem Grunde ist die Behandlung von Features zentral. Die Schwierigkeit liegt darin, dass kurze Featurevektoren immer noch die gesamte in den Signalen vorhandene Information enthalten soll. Die wichtigsten Punkte sind:

- Eigenschaften von Features.
- Technische Vorbereitungen.
- Parametermodellierung und das Modell fuer die Auswirkungen der Quellen.
- Verschiedene Prozesse und Vorhersagemoeglichkeiten
- Featureextraktionen.

### 16.7.1 Kapiteluebersicht

Die Dissertation hat vier Teile:

- Teil A: Hier wird in den Gegenstand eingefuehrt.Es wird ein Ueberblick ueber Spracherkennung und Spracherzeugung gegeben sowie ueber den Einsatz einer Maschine. Dies beinhaltet die Kapitel 2 und 3. Im letzteren wird auch auf Rauschen eingegangen, es bereitet Kapitel 5 vor.

- Teil B: Hier beschreiben wir unsere Loesung zum Problem der Spracherkennung in einer hybrid verrauschten Umgebung. Dies geschieht in den Kapiteln 4-10. Kapitel 4 enthaelt die Datenvorbereitung: In Kapitel 5 behandeln wir die Loesung fuer das Problem des starken Rauschens wobei dies durch die Poissonverteilung modelliert wird. In den Kapiteln 5 bis 8 konzentrieren wir uns auf die Parameter basierte Modellierung der Signale fuer das Sprachkonstruktionsmodell. Kapitel 6 enthaelt dabei eine Diskussion der verwendeten Verstaerkungsmethoden. Kapitel 7 behandelt autoregressive Methoden im Zusammenhang mit dem Kehlkopfmodell. In den Kapiteln 8 und 9 werden die linearen Vorhersagen und ihre Parameter behandelt. Diese beiden Kapitel haengen eng zusammen. Kapitel 9 behandelt quadratische Fehler fuer die Vorhersage. Diese basieren auf der Dekomposition von "Subbands". Sie beinhalten die Minimierung des Rauschens sowie die Verwendung von Kalmanfiltern fuer nicht-stationaere farbige Rauschsignale. Kapitel 10 behandelt klassische Loesungsansaetze soweit wir sie verwendet und modifiziert haben. Dazu gehoeren u.a. die Ansaetze der Autokorrelation und Kovarianz, der Ansatz von Burg sowie der Kleinste Quadrate Ansatz ohne Constraints.

- Teil C: Hier werden die psychoakoustischen Groessen behandelt. Wir diskutieren ihre Verwendung fuer unseren Ansatz. In Kapitel 11 stellen wir unsere Vorstellung und Behandlung von quantitativen Groessen fuer diese Zwecke. Das hat wesentliche Auswirkungen auf die Perzeptionsmodelle. Zu diesem Zwecke wird ein Gehoermodell eingefuehrt. Zur Beschreibung verwenden wir unterschiedliche Skalierungen, fuer die dann anschliessend entsprechende Filter benutzt werden. Das Ganze wird dann Teil von DANSR. Im Mittelpunkt der Kapitel 13 und 14 stehen die Features und ihre Extraktion. In Kapitel 13 werden unterschiedliche Beschreibungen fuer Features untersucht. Das ist insofern zentral als die Features die einzigen Elemente sind, die Informationen zur weiteren Verwendung enthalten. Man findet hier hier die Cepstrum Features sowie die MFCC, SILTT, LPC, LPCC und die PLP Features. Kapitel 14 behandelt dann die Extraktionsmethoden in DANSR. Dazu gehoeren zentral die Signaldekomposition und die Verwendung von Windowtechniken um Ueberlappungen zu beschreiben. Zur letzteren verwenden wir DCT-IV und ihr Inverses.

- Teil D: Hier behandeln wir Klassifikation und die eigentliche Spracherkennung. Dies geschieht in Kapitel 15. Grundlegend verwenden wir das Hidden Markov Modell (HMM) zur Bschreibung des Sprachprozesses in einem Modell. Dabei wird das akustische Modell als Gaussian Mixture Modell (GMM) beschrieben. Die korrekte Spracherkennung ist dann ein Zuordnungsproblem, fuer welches Lernen und Suchen verwendet wird. Technisch gesehen verwenden wir u.a. Vorwaerts- und Rueckwaertssuche, Viterbisuche sowie den Baum-Welch Algorithmus. Zum Vergleich von Wortfolgen fuehren wir den DTW ein. In Kapitel 16 stellen wir Experimente und ihre Resultate fuer die gesamte Arbeit vor.

- Am Ende findet man die Resultate, die Auswertungen und Abschlussbetrachtungen.

Die einzelnen Kapitel:

- Kapitel 2 gibt einen Ueberblick ueber die Spracherzeugung und Erkennung. Beides geschieht sowohl bezueglich eines Menschen wie auch einer Maschine.

- Kapitel 3 fuehrt in unsere Methodologie ein, insbesondere wenn die Sprache durch Rauschen gestoert ist. Fuer letzteres wird auch der allgemeine Hintergrund mit seinen vielen Fazetten aufgearbeitet.

- In Kapitel 4 stellen wir unsere Behandlung des starken Rauschens vor. Hierzu gehoert auch die Modellierung mittels Poissonverteilungen.

- In Kapitel 5 diskutieren wir unsere Vorgehensweise fuer ein generelles parametrisches Sprachproduktionsmodell.

- Kapitel 6 praesentiert unsere Vorverarbeitungsmethoden.

- Autogressive Prozesse im Kehlkopf sind Gegenstand von Kapitel 7.

- Lineare Vorhersage und ihre Parameter werden in den Kapiteln 8 und 9 untersucht, zwei sehr stark verbundene Kapitel.

- Kapitel 10 behandelt die Dekomposition von Signalen und ihre Rauschminimisierung. Kalman Filter werden werden fuer die nicht-lineare Behandlung von farbigen Rauschsignalen verewndet.

- Kapitel 11 wendet sich psychoakustischen Phaenomenen in Bezug auf ihre Quantifizierung und ihre Behandlung in DANSR zu.
- In Kapitel 12 werden allgemeine Features mit ihren Eigenschaften und Extraktionsmethoden vorgestellt.
- In Kapitel 13 findet man unsere Form der Extraktion innerhalb von DANSR.
- Kapitel 14 behandelt schliesslich die Form der Klassifikation als eigentliche Spracherkennung in einem Lernsystem. Die Modellierung geschieht in HMM und GMM.
- Kapitel 15 beinhaltet Loesungen und experimentelle Resultate.
- Abschliessende Bemerkungen enthaelt Kapitel 16.

# Bibliography

[1] H. Hermansky, N. Morgan, A. Bayya and P. Kohn . Compensation for the effect of the communication channel in auditory-like analysis of speech rasta-plp. isca, (1991). Eurospeech, ISCA, 1991. 188, 190

[2] A. I. Hanna. Gaussian mixture models - algorithm and matlab code. Wavepage of Sagoforest, November 2006. 214

[3] A. J. Oxenham. Frequency selectivity and masking. Harvard-MIT Division of Health Sciences and Technology, 2005. 148

[4] A. Muthukumarasamy. *Impact of Microphone Positional Errors on Speech Intelligibility*. Master's thesis, Kentucky University, USA, 2009. 47

[5] A. Papoulis. *Probability Random Variables, and Stochastic Processes*. McGraw-Hill, NY,USA, 3rd edition. 59

[6] A. Spanias. Speech coding: A tutorial review. **82**, October 1994. 127, 148, 200

[7] A. Spanias, T. Painter and V. Atti. *Audio Signal Processing and Coding*. John Wiley and Sons, 2007. 127

[8] A. V. D. Veen and G. Leus. 4235 digital signal processing. Available: http://ens.ewi.tudelft.nl/Education/courses/et4235/, 2008. 79, 80

[9] A. V. Oppenheim and R. W. Schafer. *Digital Signal Processing*. Prentice Hall, NJ, USA, 1975. 80, 174, 175, 188

[10] A. Zaknich. *Principles of Adaptive Filters and Self-learning Systems*. Springer-Verlag, London, UK, 2005. 77, 79, 80, 81, 133

[11] B. -H. JUANG AND S. FURUI. Automatic recognition and understanding of spoken language—a first step toward natural human–machine communication. **88**, pages 1142–1165. Proceedings of IEEE, August 2000. 235

[12] B. C. J. MOORE. *Handbook of Perception and Cognition : Hearing*. Academic Press, Burlington, MA, USA, 2nd edition, 1995. xv, 148, 149, 155, 156, 159

[13] B. MILLER. Auditory system. Available: http://www.youtube.com/watch?v=-Gs3Tx2pGC8, March 2012. 152

[14] B. P. BOGERT, M. J. R. HEALY AND J. W. TUKEY. The quefrency alanysis of time series for echoes: Cepstrum, pseudo autocovariance, cross-cepstrum and saphe cracking. pages 209–243. Proceedings of the Symposium on Time Series Analysis, 1963. 176

[15] B. PELLOM. Automatic speech recognition: From theory to practice. Lectruenotes, September 2004. 161

[16] B. RAJ, M. L. SELTZER AND R. M. STERN. Reconstruction of missing features for robust speech recognition. **43**, pages 275–296. Elsevier, Speech Communication. 189

[17] B. SCHULLER, M. WOELLMER, T. MOOSMAYR AND G. RIGOLL. Recognition of noisy speech: A comparative survey of robust model architecture and feature enhancement. EURASIP Journal on Audio, Speech, and Music Processing, Hindawi Publishing Corporation, 2009. 3

[18] B. W. SILVERMAN. Density estimation for statistics and data analysis. London: Chapman and Hall, 1986. Monographs on Statistics and Applied Probability. 33

[19] C. BECCHETTI AND L. P. RICOTTI. *Speech Recognition Theory and C++ Implementation*. John Wiley and Sons, England, 2002. 45

[20] C. E. SHANNON. A mathematical theory of communication. pages 379–423. The Bell System Technical Journal, 1948. 3

[21] C. K. CHUI AND G. CHEN. *Kalman Filtering with Real-Time Applications*. Springer-Verlag, 3rd edition, 1998. 131, 133, 134

[22] C. Sungwook, Y. Kwon, Y. and Y. Sang-Il. Speech recognition system using adapted local trigonometric transforms. pages 157 – 160, Pittsburgh, PA, October 1998. IEEE-SP International Symposium. 5

[23] D. A. Bies and C. H. Hansen. *Engineering Noise Control: Theory and Practice*. Taylor and Francis, KY, USA, 4th edition, 2009. 27, 28, 43

[24] D. G. Manolakis, V. K. Ingle and S. M. Kogon. *Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modeling, Adaptive Filtering and Array Processing*. Artech House, 2000. xiv, 19, 95, 105, 108, 112

[25] D. P. W. Ellis. Plp and rasta, and mfcc, and inversion in matlab. Wavepage, 2005. 161, 230

[26] D. R. Lanman. Design of a sound level meter. Laboratory Report EN 253: Matlab Exercise, November 2005. 27, 28

[27] D. Reynolds. Gaussian mixture models, February, 2008. 216

[28] D. Rocchesso. *Introduction to Sound Processing*. Universit'a di Verona Dipartimento di Informatica, Italy, March 2003. 148

[29] E. Ademovic, J.-C. Pesquet and G. Charbonneau. Wheezing lung sounds analysis with adaptive local trigonometric transform. **6**, pages 41–51. Technology and Health Care, IOS Press, November 1998. 203

[30] E. Ambikairajah. Elec9344 speech and audio processing. Lecturenotes Available:http://tv.unsw.edu.au/pdf/chapter-2-pdf. 10, 18, 68, 69

[31] E. Haensler and G. Schmidt. *Acoustic echo and noise control: A practical Approach*. Wiley-IEEE press, June 2005. 133

[32] E. Wesfried and M. V. Wickerhauser. Adapted local trigonometric transforms and speech processing. **41**. IEEE Transactions on Signal Processing, December 1993. 5, 189, 194

[33] F. -H. Liu, R. M. Stern, X. Huang and A. Acero. Efficient cepstral normalization for robust speech recognition. pages 69–74, Pennsylvania(PA), USA, 1993. Proceedings of the workshop on Human Language Technology (HLT). 188

[34] F. Mustiere, M. Bolic and M. Bouchard. Improved colored noise handling in kalman-based speech enhancement algorithms. pages 000497 – 000500, Ontario, Canada, May 2008. IEEE, CCECE. 123, 125, 127

[35] F. Nolan. Intonational equivalence: An experimental evaluation of pitch scales. Number 15. International Congress of Phonetic Sciences, 2003. 160

[36] G. Blanchet and M. Charbit. *Digital signal and image processing using Matlab*. ISTE, London, Newport Beach (Calif.), 2006. 61

[37] G. D. Moore. Physics and psychophysics of music. 154

[38] G. Doblinger. Computationally efficient speech enhancement by spectral minima tracking in subbands. pages 1513–1516, Madrid, Spain, September 1995. Eurospeech. 123, 129

[39] G. Strang. The discrete cosine transform. *SIAM Review*. 197

[40] H. Bourlard. Non-stationary multi-channel (multi-stream) procesing towards robust and adaptive asr. pages 1–10. Proceedings of the ESCA Workshop on Robust Methods for Speech Recognition in Adverse Conditions. 188, 189

[41] H. Fastl and E. Zwicker. *Psychoacoustics: Facts and Models*. Springer, Deutschland, Heidelberg, 2007. 156, 157, 186

[42] H. Hermansky. Perceptual linear predictive (plp) analysis of speech. **87**, Santa Barbara, California, November 1989. Accoustical Society of America. 148, 159, 184, 185

[43] H. Hermansky and N. Morgan. Rasta processing of speech. **2** of *4*, pages 578 – 589. Speech and Audio Processing, IEEE Transactions, October 1994. 188

[44] H. Hermansky, N. Morgan, and H. -G. Hirsch. Recognition of speech in additive and convolutional noise based on rasta spectral processing. **2**, pages 83–86. ICASSP, IEEE, April 1993. 188

[45] H. Puder. Kalman-filters in subbands for noise reduction with enhanced pitch-adaptive speech model estimation. **13** of *2*, pages 139–148. European Transactions on Telecommunications. 2

[46] H. Puder. Noise reduction with kalman-filters for hands-free car phones based on parametric spectral speech and noise estimates. In *Acoustic Echo and Noise Control*, pages 385–427. Springer, 2006. 2, 22, 133, 143

[47] H. S. Malvar and D. H. Staelin. The lot:transform coding without blocking effects. **37**, pages 553–559. ICASSP, April 1989. 202

[48] E. Haensler and G. Schmidt. *Acoustic echo and noise control: a practical approach*. Wiley-IEEE press, New Jersy, USA, June 2005. 235

[49] I. C. Couvreur. Acoustics software survey. Available: http://pims.grc.nasa.gov/plots/user/acoustics/survey.html. 27

[50] I. Cohen. *Shift-Invariant Adaptive Wavelet Decompositions and Applications*. PhD thesis, Senate of the Technion, Israel Institute of Technology, 1998. 189, 203

[51] I. Cohen, Y. Huang and J. Chen. *Noise Reduction in Speech Processing*. Springer, Deutschland, 2009. 27, 28

[52] A.C. Instituto Linguistico de Verano. Organs articulation, 2005. xiv, 66

[53] J. -L. Wu. Information theory and coding technique. Lecture notes of csie 7614, National Taiwan University, April 2005. 172

[54] J. -P. Hosom. Cs 552/652: Automatic speech recognition with hidden markov models, 2011. 180, 219, 220

[55] J. A. Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Techreport, April 1998. 214

[56] J. Berger. https://ccrma.stanford.edu/courses/120-fall-2003/lecture-5.html. Lectruenotes, Fall 2003. 161

[57] J. D. Johnston. Estimation of perceptual entropy using noise masking criteria. **5**, pages 2524–2527. ICASSP, April 1988. 4, 200

[58] J. D. Johnston. Transform coding of audio signals using perceptual noise criteria. **6**. IEEE, Februrary 1988. 148, 200

[59] J. G. Proakis and D. G. Manolakis. *Digital Signal Processing Principles, Algorithms, and Applications.* Prentice Hall, NJ, USA, 3rd edition, 1996. 242

[60] J. G. Proakis and D. G. Manolakis. *Digital Signal Processing Principles, Algorithms, and Applications.* Prentice-Hall, New Jersy, USA, 3 edition, 1998. 67, 112, 119, 188

[61] J. Harrington and S. Cassidy. *Techniques in Speech Acoustics.* Kluwer Academic Publishers, 1999. 157

[62] J. Hou, L. Rabiner and S. Dusan. Auditory models for speech analysis. Technical report, Rutzer University, Available: http://www.ece.gatech.edu/research/labs/asat/slides/meet-111204/Auditory-lrr.pdf. 190

[63] J. J. Shynk. *Probability, Random Variables, and Random Processes : Theory and Signal Processing Applications.* Wiley, NJ, USA, September 2012. 79, 80

[64] J. J. Thiagarajan and A. Spanias. *Analysis of the Mpeg-1 Layer III (Mp3) Algorithm Using Matlab.* Morgan and Claypool Publishers, December 2012. 200

[65] J. M. Zanker. Auditory perception: Hearing noise and sound. http://www.pc.rhul.ac.uk/staff/J.Zanker/PS1061, June 2013. xv, 152, 153

[66] J. Makhoul. Linear prediction: A tutorial review. **63**, pages 561–580. Proceedings of IEEE, April 1975. 69

[67] J. O. Smith. *Spectral Audio Signal Processing.* W3K Publishing, http://books.w3k.org/, 2011. 161

[68] J. P. Campbell. Speaker recognition: A tutorial. **85**. IEEE, September 1997. 190, 235

[69] J. R. Deller, J. H.L. Hansen and J. G. Proakis. *Discrete-Time Processing of Speech Signal.* IEEE Press, New York, USA, 2000. 15, 19, 65, 68, 174

[70] J. Verbeek. Mixture density estimation. Wavepage, 2010. 217

[71] K. K. Paliwal. Neural net classifiers for robust speech recognition under noisy environments. pages 429 – 432, Albuquerque, NM, April 1990. ICASSP, IEEE. 4

[72] K. K. Paliwal and A. Basu. A speech enhancement method based on kalman filtering. **12**, pages 177 – 180. IEEE, ICASSP, April 1987. 139

[73] K. Ohkura and M. Sugiyama. Speech recognition in a noisy environment using a noise reduction neural network and a codebook mapping technique. **2**, pages 929 – 932, Toronto, Ontario, May 1991. ICASSP. 2

[74] L. Deng and D. O'Shaughnessy. *Speech Processing A dynamic and Optimization-oriented Approach*. Marcel Dekker Inc, New York, NY, USA, 2003. 148

[75] L. F. Villemoes. Adapted bases of time-frequency local cosines. **10**, pages 139–162. ScienceDirect, March 2001. 203

[76] L. G. Johansen. Psychoacoustics and audibility -fundamental aspects of the human hearing. Lecture notes TI-EAKU, University Colleege of Aarhus, 2006. 155

[77] L. Hong, J. Rosca and R. Balan. Independent component analysis based single channel speech enhancement using wiener filter. **3**, pages 522 – 525. ISSPIT, December 2003. 2

[78] L. McManus, M. Schoenwiesner, T. Lysaght and J. Timoney. Implementing loudness models in matlab. Number 7, Naples, Italy, October 2004. DAFX. 160

[79] L. Mimic'. *Multirate Filtering for Digital Signal Processing*. IGI Global, 2009. 126

[80] L. R. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, New Jersy, USA, 1993. xiii, 10, 11, 15, 48, 68, 99, 101, 111, 235

[81] L. R. Rabiner and R. W. Schafer. *Digital Processing of Speech Signals*. Prentice-Hall, 1978. 18

[82] L. R. Rabiner and R. W. Schafer. *Introduction to Digital Speech Processing*, **1**. Foundations and Trends, Santa Barbara UC, USA, 2007. 148

[83] L. Tan. *Digital Signal Processing : Fundamentals and Application*. Academic press, 2nd edition, September 2007. 126

[84] L. Tan and J. Jiang. *Digital Signal Processing : Fundamentals and Application*. Academic Press, USA, 2nd edition, 2013. 126, 240

[85] M. Brookes. Voicebox: Speech processing toolbox for matlab. http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html. 151, 162

[86] M. Fujimoto and Y. Ariki. Noisy speech recognition using noise reduction method based on kalman filter. **3**, pages 1727 – 1730, Istanbul, 2000. ICASSP. 2, 22

[87] M. Hasegawa-Johnson. Lecture notes in speech production, speech coding, and speech recognition. Wavepage of University of Illinois at Urbana-Champaign, 2000. 17, 18, 91

[88] M. Hazas. *Processing of Non-Stationary Audio Signals*. Master's thesis, Department of Engineering, University of Cambridge, August 1999. 189, 203

[89] "M. J. L. De Hoon, T. H. J. J. Van Der Hagen, H. Schoonewelle, and H. Van Dam". Why yule-walker should not be used for autoregressive modeling. Techreport, Interfaculty Reactor Institute, Delft University of Technology, Delft, Merkelweg, the Netherlands. 122

[90] M. Kahrs and K. Brandenburg. *Applications of Digital Signal Processing to Audio and Acoustics*. Kluwer Academic Publisher, Boson, USA, 2002. 148, 166, 200, 201

[91] M. Moeser. *Engineering Acoustics: An Introduction to Noise Control*. Springer, 2009. 27, 28, 30

[92] M. Najim. *Modeling, Estimation and Optimal Filtration in Signal Processing*. Wiley-ISTE, January 2010. 79, 80, 81

[93] M. P. Ghaemmaghami, F. Razzazi, S. Sameti and S. Dabbaghchian . Noise reduction algorithm for robust speech recognition using mlp neural network. **1**, pages 377 – 380, Wuhan, November 2009. Computational Intelligence and Industrial Applications. 4

[94] M. Sewell. Hidden markov models. Technical report, Department of Computer Science University College London, 2008. 206

[95] M. V. Wickerhauser. *Adapted Wavelet Analysis from Theory to Software*. IEEE Press, New York, USA, 1994. 5, 189, 194

[96] M. Wickert. Lecture notes in statistical signal processing. Course Wavepage of University of Colorado, August 2013. 86

[97] M. Wolfgang, N. Satoshi and M. Konstantin. *Incorporating Knowledge Sources into Statistical Speech Recognition*. Springer, New York, USA, 2009. 235

[98] M. Zoltowski. Ece 538 digital signal processing. EE538 Lecturenotes of Univesity of Purdue, 2010. 121

[99] N. Morgan and H. Hermansky. Rasta extensions: robustness to additive and convolutional noise. pages 115–118, Cannes-Mandelieu, France, November 1992. SPAC. 188

[100] O. Cheng, W. Abdulla and Z. Salcic. Performance evaluation of front-end processing for speech recognition systems. School of Engineering Report 621, Faculty of Engineering, University of Auckland, 2005. 166, 185, 186

[101] P. Aparna and S. David. Adaptive local cosine transform for seismic image compression. pages 254–257, Surathkal, December 2006. ADCOM, IEEE. 189, 203

[102] P. Boersma and D. Weenink. Praat: doing phonetics by computer [computer program]. 51

[103] P. C. -Y. Yip. *The Transformation and Applications Handbook*. CRC Press, 2nd edition, Februrary 2000. 180, 197

[104] P. Leechor, C. Pornpanomchai and P. Sukklay. Operation of a radio-controlled car by voice commands. pages V1–14 – V1–17, Kyoto, August 2010. ICMEE. 4

[105] Proceedings of the IEEE. *Perceptual Coding of Digital Audio*, **88**. T. Painter and A. Spanias, April 2000. 148, 150, 200

[106] R. A. Losada. *Digital Filters with Matlab*. The MathWorks, Inc, May 2008. 46, 47

[107] R. C. V. Dalen and M.J.F. Gales. Extended vts for noise-robust speech recognition. **19** of *4*, pages 733 – 743. IEEE Transactions on Audio, Speech, and Language Processing, 2011. 4

[108] R. L. Allen and D. Mills. *Signal Analysis : Time, Frequency, Scale, and Structure*. Wiley-IEEE press, NJ,USA, 2004. 46, 48

[109] R. Pintelon and J. Schoukens. *System Identification : A Frequency Domain Approach*. Wiley-IEEE, 2nd edition, April 2012. 79, 80

[110] S. B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. **28**, pages 357 – 366. Acoustics, Speech and Signal Processing, IEEE Transactions, August 1980. 148, 160, 177

[111] S. Chang, Y. Kwon and S.-l. Yang. Speech recognition system using adapted local trigonometric transforms. pages 157 – 160. IEEE-SP International Symposium, Pittsburgh, PA, October 1998. 189, 203

[112] S. E. Eddy. What is dynamic programming. **22**. Nature Biotechnology, July 2004. 5

[113] S. Errede. Uiuc physics 406 : Acoustical physics of music. 147, 148

[114] S. F. Chen, M. A. Picheny and B. Ramabhadran. Eecs e6870: Advanced speech recognition. IBM T.J. Watson Research Center Yorktown Heights, NY, USA, September 2009. 51, 228

[115] S. Furui. Speaker-independent isolated word recognition using dynamic features of speech spectrum. **34**, pages 52–59. IEEE Transactions ASSP, 1986. 180

[116] S. Ganot, D. Burshtein and E. Weinstein. Iterative and sequential kalman filter-based speech enhancement algorithms. IEEE, IEEE Transactions on Speech and Acoustics, 1998. 141

[117] S. Hamaguchi, N. Kitaoka and S. Nakagawa . Robust speech recognition under noisy environments based on selection of multiple noise suppression methods. Sapporo, May 2005. NSIP, IEEE-Eurasip. 2, 21

[118] S. J. Orfanidis. *Optimum Signal Processing*. McGraw-Hill Publishing, New York, USA, 2007. xiii, 17

[119] S. J. Orfanidis. *Introduction to Digital Signal Processing*. Prentice Hall, 2010. 127

[120] S. J. Young. *HTKBook*, **3.4**. Cambridge University Engineering Department, 2006. 43

[121] S. L. Marple, Jr. *Digital Spectral Analysis with Application*. Prentice Hall, NJ, USA, 1987. xiv, 82, 84, 91, 111, 117

[122] S. L. Marple, Jr. A fast computational algorithm for the modified covariance method of linear prediction. **1** of *3*. Digital signal processing, Academic press, 1991. 117, 119

[123] S. Moon and J. -N. Hwang. Noisy speech recognition using robust inversion of hidden markov models. **1**, pages 145 – 148, Detroit, MI, May 1995. ICASSP. 4

[124] S. Paul, M. M. Richter and S. Liu. Hybrid solution to single-channel hybrid noisy speech for an industrial environment. **12**, Ho Chi Minh City, Vietnam, December 2012. ISSPIT, IEEE. 144

[125] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic press, Elsevier, UK, 4th edition, 2009. 224

[126] S. V. Vaseghi. *Advanced Signal Processing and Digital Noise Reduction*. Wiley, Teubner, Leipzig, Germany, 1995. 136

[127] S. V. Vaseghi. *Multimedia Signal Processing: Theory in Speech, Music and Communications*. Wiley, USA, 2008. 101, 131, 133, 135, 206, 208, 209, 226, 242, 243

[128] S. Young. Hidden markov model toolkit (htk). http://htk.eng.cam.ac.uk/, 2013. 4

[129] T. D. Rossing, editor. *Springer Handbook of Acoustics*. Springer, USA, New York, 2007. 159

[130] T. Dutoit and F. Marques. *Applied Signal Processing*. Springer, 2009. xvi, 148, 205, 210

[131] T. F. Quatieri. *Discrete-Time Speech Signal Processing: Principles and Practice*. Prentice Hall, New Jersy, USA, 2001. 67, 68, 240

[132] T. Q. Nguyen and P. P. Vaidanathan. Structures for m-channel perfect reconstruction fir qmf banks which yield linear-phase analysis filters. **38**, pages 433–446. Acoustics, Speech and Signal Processing, IEEE Transactions, March 1990. 123

[133] T. Virtanen, R. Singh and B. Raj. *Techniques for Noise Robustness in Automatic Speech Recognition*. Wiley, first edition, October 2012. 214

[134] Tsukuba Research and Development Center. *Signal Modelling Techniques in Speech Recognition*. IEEE, June 1993. 235

[135] U. Bies. Weighting filter set. Available: http://www.beis.de/Elektronik/AudioMeasure/WeightingFilters.html, October 2006. 27, 28

[136] U. H. Yapanel and J. H. L.Hansen. A new perceptually motivated mvdr-based acoustic front-end (pmvdr) for robust automatic speech recognition. **50**, pages 142–152. Speech Communication, Elsevier, 2008. 189

[137] unknown. Kernel density estimation, March 2005. 33

[138] unknown. Box plot, unknown. 33

[139] V. Atti. *Algorithms and Software for Predictive and Perceptual Modeling of Speech*. Morgan and Claypool Publishers, March 2011. 148, 156

[140] P.P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice Hall, NJ, USA, 1993. 143

[141] W. C. V. Etten. *Introduction to Random Signals and Noise*. John Wiley and Sons, August 2005. 58, 61

[142] W. M. Hartmann. *Signals, Sound, and Sensation*. Springer-Verlag, AIP Press, September 1997. 154, 159

[143] W. M. Hartmann. *Signals, Sound, and Sensation*. Springer, New York, USA, January 1998. 148, 158

[144] M. Woelfel and J. McDonough. *Distant Speech Recognition*. Wiley, UK, April 2009. 190, 235

[145] X. Anguera. 5-perceptual-models. Available on google: 5-Perceptual-Models.pdf, 2011. 148, 149, 155

[146] X. Cui and Y. Gong. A study of variable-parameter gaussian mixture hidden markov modeling for noisy speech recognition. **15** of *4*, pages 1366 – 1376. Audio, Speech, and Language Processing, IEEE Transactions, May 2007. 2

[147] X. Huang, A. Acero and H. -W. Hon. *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Prentice Hall, 2001. 51, 155, 235

[148] X. Zhang, L. Hua and G. Evangelista. Warped filter banks used in noisy speech recognition. pages 1385 – 1388, Kaohsiung, December 2009. Innovative Computing, Information and Control (ICICIC). 2, 21

[149] Z. -H. Tan and B. Lindberg. *Automatic Speech Recognition on Mobile Devices and Over Communication Networks*. Springer, January 2008. 235

# Index

# Curriculum Vitae

**Sheuli Paul**

**Education**

| | |
|---|---|
| 1/2004 – 09/2007 | **University of Calgary** |
| | **Department:** Electrical and Computer Engineering<br>Degree: M. Sc |
| | Thesis: Dynamic Time Warping for Small Vocabulary Word Recognition. |
| 1/1995 – 11/1999 | **Chittagong University of Engineering Technology** |
| | **Department:** Electrical and Electronics Engineering<br>Degree: B. Sc. |
| 10/1992 – 12/1994 | **Chemistry, Chittagong University** |
| 01/1990 – 09/1992 | **Chittaong Govt. College (Abitur II)** |
| | Degree: Higher School Secondary in Science (H. Sc.) |
| 01/1979 – 12/1989 | **Cox's Bazar Primary and High School (Abitur I)** |
| | Degree: School Secondary in Science (S. Sc.) |

**Experience**

| | |
|---|---|
| 11/2009 – 02/2014 | **University of Kaiserslautern** |
| | **Department:** Electrical and Information Technology |
| | Ph.D. Student. |

| | |
|---|---|
| 08/2008 – 09/2009 | **Geomathematik, University of Kaiserslautern**<br>Scientific Research Assistant. |
| 01/2008 – 07/2008 | **Deutsches Forschungsinstitut für Künstliche Intelligenz (DFKI), Kaiserslautern**<br>Software Engineer |
| 01/2002 – 09/2005 | **University of Calgary**<br>**Department:** Electrical and Computer Engineering<br>Student and Research Assistant. |
| 06/2000 – 09/2001 | **Logican Technology, Edmonton**<br>Quality Control. |