Karlsruhe Institute of Technology

Evaluation of Synchronized Damped Oscillator Cepstral Coefficients for Robust Speech Recognition

Bachelorarbeit
von

Juan Hussain

am Institut für Anthropomatik und Robotik
der Fakultät für Informatik

Erstgutachter:	Prof. Dr. Alex Waibel
Zweitgutachter:	Dr. Sebastian Stüker
Betreuender Mitarbeiter:	M. Sc. Markus Müller

Bearbeitungszeit: 15. Juli 2014 – 11. November 2014

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Karlsruhe, den 11. November 2014

Abstract

In this thesis, we have implemented and evaluated Synchronized Damped Oscillator Cepstral Coefficients for robust speech recognition. At the beginning, we present fundamental knowledges about speech production process and hearing mechanism in order to help understand the theory behind the approach. Then, we describe the automatic recognition systems including the one we used for the evaluation, and show the components for extracting Synchronized Damped Oscillator Cepstral Coefficients in detail. We conclude by showing cases in which these new input features show promising results.

Zusammenfassung

In dieser Arbeiten haben wir Synchronized Damped Oscillator Cepstral Coefficients für robuste Spracherkennung implementiert und evaluiert. Zum besseren Verständnis zeigen wir die Funktionsweise des Sprachproduktionsprozesses und des Hörens auf. Wir beschreiben Systeme zur automatischen Spracherkennung und insbesondere das zur Evaluation verwendete System. Insbesondere beschreiben wir die einzelnen Komponenten die zum Erzeugen der SyDOCCs nötig sind. Zum Schluss zeigen wir die Experimente, bei denen die SyDOCCs vielversprechende Ergebnisse geliefert haben.

Contents

1. Introduction

Speech recognition occupies more and more employment fields: From using it as a replacement of the keyboard to investing it in the automatic translation in international conferences or war areas to help other people with different languages. However, it is still a very wide research field in terms of the speech recognition in noisy environments.
This thesis has the purpose of implementing and evaluating the Synchronised Damped Oscillator Cepstral Coefficients (SyDOCCs) for robust speech recognition: A new method motivated by the hearing mechanism and new scientific findings about hearing in different species. It has the goal to make speech recognition more robust toward environmental variability, and to achieve insensitivity to all kinds of noises which the human ear is insensitive to. The current used techniques are very sensitive to noise, where the performance degrades highly in a noisy environment or by using a noisy channel.

1.1 Related works

Typically Mel-Frequency Cepstral Coefficients MFCCs are used as acoustic features in a traditional automatic speech recognition system. MFCCs perform very well under the clean conditions, but the performance degrades when dealing with noisy speech. There are many approaches to deal with this problem, for example, RelAtive SpecTrA Perceptual Linear Prediction (RASTA-PLP) shows more robustness under noisy Environment compared to MFCCs. Researchers have explored human perception based speech analysis techniques for feature acoustic generation, such as Power Normalized Cepstral Coefficients (PNCC) and Perceptually motivated Minimum Variance DistoRtions (PMVDR).

1.2 The structure of the thesis

To go into the deep details at the heart of this thesis, from the theory until the evaluation, beginners should understand the basic knowledge about speech production, the anatomy of the speech apparatus, and the basic units of speech and their physical nature. These are subjects of the first part of chapter 2 (Speech production

and auditory system) section 2.1 (speech production) . The hearing mechanism, in the second part (section 2.2 auditory system), is the core of the theory behind this work where we explain basic knowledges about ear physiology. However, we go in more details about the inner ear (basilar membrane and hair cells), and the function of coding the sound to electrical signals, followed by a brief description of this signals related to our work. Chapter 3 (automatic speech recognition) describes the typical architecture of an automatic speech recognition system and the process of finding a sequence of words being recognized, starting with some details of how a typical acoustic features are extracted, where we explain the steps of extracting Mel-Frequency Cepstral Coefficients MFCCs from a row signal. Then, we come to a component called dictionary or lexicon where we find all words to be recognized by a system, and the language model, which give information about the frequency or probability of a word or a sequence of words in a certain language. Chapter 4 (Synchronized Damped Oscillator Cepstral Coefficients) is the main part of this thesis. It goes step by step through all the components needed to build the Synchronized Damped Oscillator Cepstral Coefficients (SyDOCCs): First, windowing the signal then passing it to a form like band-pass filters to analyse the frequency components. Then, synchronizing the outputs and passing them to the damped oscillators, which act like hair cells. We also explain the difference in system performance by using root compression, used in SyDOCCs, and log compression, used in MFCCs. We conclude with the evaluation (chapter 5), where we present the experiments performed to examine SyDOCCs. This chapter is divided into sections, one for each language, describing the used data for test and training, system settings, and the summery of the results shown in tables and diagrams.

2. Speech production and auditory system

2.1 Speech production

Sound travels throw the vocal tract of a speaker taking different wave shapes or, as we say in the signal processing point of view, the sound wave is being modulated by the vocal tract filter. We will see the travel way closely in section 2.1.1. The term **phoneme** refers to the smallest unit of speech sound in a language, and the term **phone** is an acoustic realisation of a phoneme. Phonemes can be divided into two basic classes [HuAH01]:

- Consonants: articulated in the presence of constrictions in throat or mouth. The phonemes /m/, /l/ and /k/ are examples.
- Vowels: articulated without major constrictions. The phoneme /a/ in bat is an example.

More details are in section 2.1.2 and 2.1.3.

2.1.1 Atriculators

Speech is produced by air-pressure waves generated from the lung as an air stream. Figure 2.1 illustrates at the upper panel the human speech production system. The speech production process begins with an air stream passes through the **vocal folds**, where voiced and unvoiced sound are being distinguished at this stage depending on whether the vocal folds oscillate or not. **The velum** controls the air passage to go in the **oral** or **nasal cavity**. This cavities can be better seen in the vocal tract model in the lower panel of figure 2.1. They play an important role to form phonemes, for example, the travel of air through the nasal cavity is necessary to spell the phoneme /m/ or /n/ correctly. **The tongue** plays a major role to shape vocal tract configurations. For example, the configuration of the phoneme /L/ is formed when the tongue touches the **hard palate**, which is a hard surface at the roof inside the mouth. **Lips** are also very important to form vowels, for example, by rounding lips, and to form consonants such as /m/, /p/ and /b/ as well. **Teeth** are also important to form many consonants such as /s/ and /f/.

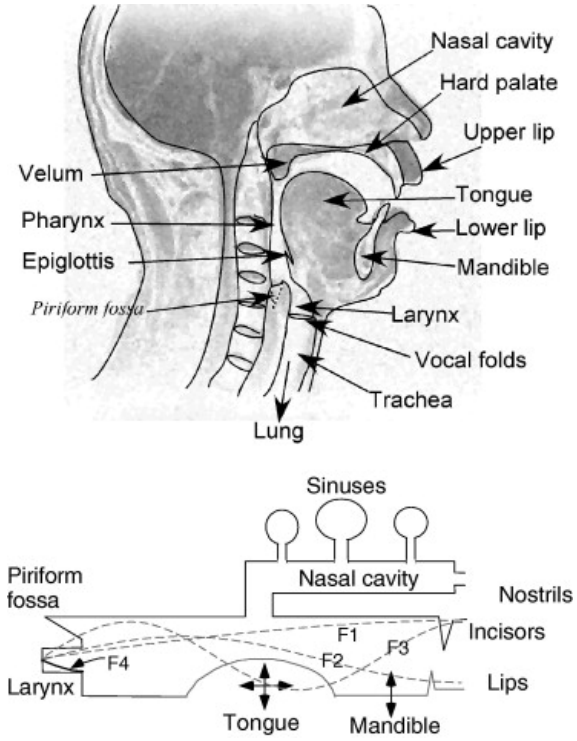

Figure 2.1: Vocal tract of the human, and a model of the vocal tract with side branches. The standing waveforms of four formants are also illustrated in the vocal tract [LuDa08]

2.1.2 Vowels

Like a wind music instrument, the vocal tract acts with each configuration, which could be a certain lip or tongue shape or tongue positioning, as a filter with certain resonance frequencies called formants (Figure 2.1 lower panel). For vowels, we have two major resonance frequencies called the first formant (F1) and the second formant (F2). Both are very important for the classification task. We see in table 2.1 typical formant frequencies for English vowels.

Table 2.1: English vowels and its typical formants [HuAH01]

Vowel Lables	Mean F1 (Hz)	Mean F2 (Hz)
iy (f**ee**l)	300	2300
ih (f**i**ll)	360	2100
ae (g**a**s)	750	1750
aa (f**a**ther)	680	1100
ah (c**u**t)	720	1240
ao (d**o**g)	600	900
ax (c**o**mply)	720	1240
eh (p**e**t)	570	1970
er (t**u**rn)	580	1380
ow (t**o**ne)	600	900
uh (g**oo**d)	380	950
uw (t**oo**l)	300	940

2.1.3 Consonants

Consonants are being generated with more constricted vocal tract configurations. For instance, the Arabic phoneme /r/ is pronounced by vibrating the tongue touching the hard palate. Table 2.2 contains other examples of English consonants.

Table 2.2: Consonant manner of articulation [HuAH01]

Manner	Phone	Example	Mechanism
Polsive	/p/	tat, tap	Closure in oral cavity
Nasal	/m/	team, meat	Closure of nasal cavity
Fricative	/s/	sick, Kiss	Turbulant airstreem noise
Retroflex liquid	/r/	rat, tar	Vowel-like tongue high and curled back
lateral liquid	/l/	lean, kneel	Vowel-like tongue central, side airstream
Glide	/y/, /w/	yes, well	Vowel-like

2.2 Auditory system

The auditory system is the sensory system for hearing, which begins with travelling the sound, which is a mechanical wave of pressure, to the outer ear and through the middle ear into the inner ear, where it is analysed and transformed to electrical waves before it reaches the auditory center of the brain, where it is interpreted.
Figure 2.2 shows the outer, middle and outer ear. The outer ear gathers the sound waves and plays a major role to help the brain determine the direction of the sound. sound waves travel through the long ear channel ending with the eardrum, which is a thin cone-shaped membrane with an area of about 500 mm^2. The middle ear consists of three very small bones known as the malleus, incus, and stapes. The middle ear collects the sound pressure over the eardrum, and focusses it on the stapes footplate, whose area is much smaller than the area of the eardrum. This helps amplify the vibrations to overcome the mechanical resistance of lymph fluid, which is behind the stapes footplate, filling the cochlea.
Now the vibrations are in the inner ear making the basilar membrane vibrates particularly greater in a certain places than other dependent on the resonance frequencies of this places. On the surface of the basilar membrane sit hair cells, which transduce the mechanic vibrations into neuroelectrical signals. We will have a close sight on the basilar membrane and hair cells in the next sections.

2.2.1 Basilar membrane

The basilar membrane is inside the cochlea and subdivides it into two spaces filled with the lymph. A special characteristic of the basilar membrane is the one that is narrow and stiff at one end and it gets wider and floppier toward the other end. This is the reason why the basilar membrane has gradually increased resonance frequencies from one end to the other as figure 2.3 illustrates. We notice also that the incrementation of this frequencies are not linear but logarithmic. This characteristic makes the basilar membrane acts like a mechanic frequency analyser. This means that if we hear a sound with only one frequency, only a small part of the basilar membrane will vibrate much stronger than others, i.e it will resonate with this frequency. The hair cells on this part will be simulated to tell our brain the place of the vibration producing a feeling of hearing that frequency.

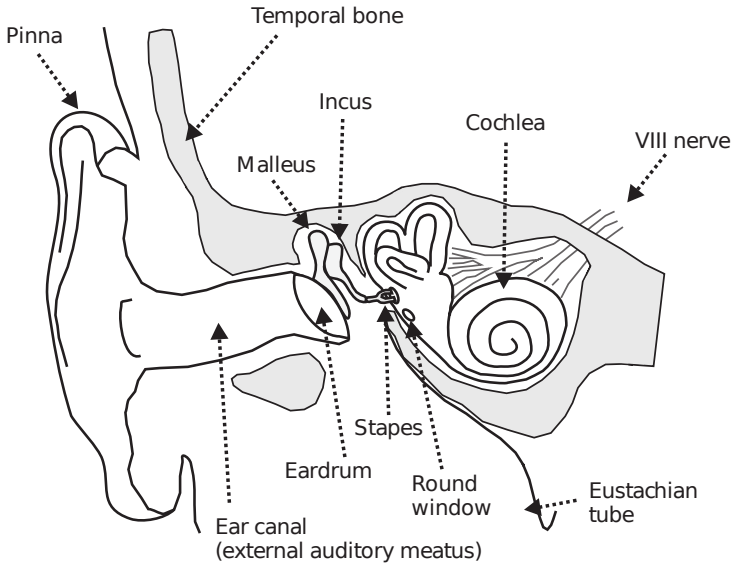

Figure 2.2: A Cross-section of the Ear with its parts: outer, middle and inner ear [ScNK11].

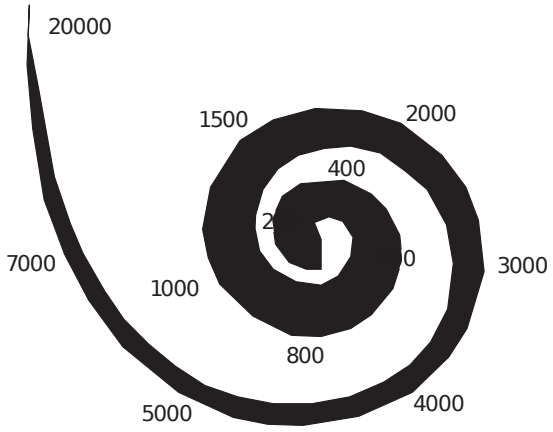

Figure 2.3: A model of basilar membrane shows the placement of resonance frequencies on it [ScNK11].

2.2.2 The Hair cell

The task of the hair cell is mainly the translation of mechanical signals into electrical signals. The organ of Corti, shown in figure 2.4, is attached with the basilar membrane, and runs along its entire length so that the organ of Corti vibrates with the basilar membrane. We see also in figure 2.4 that this organ curves and folds back over the hair cell. The fold , which known as tectorial, membrane, comes in contact to the outer hair cell stereocilia, a tiny hairs on the hair cell, from above. While the contact of the tectorial membrane transmits the mechanical vibration to the outer hair cell, the fluid flowing back and forth causes the oscillations of inner hair cell.

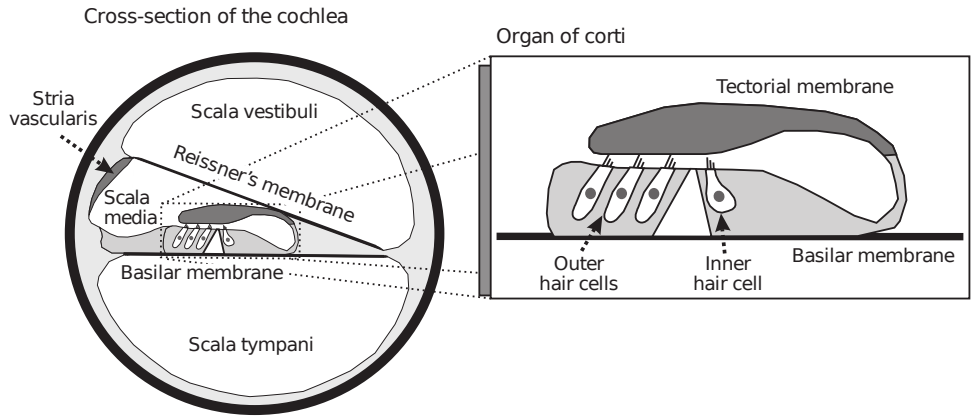

Figure 2.4: The organ of Corti; a cross-section and a schematic view [ScNK11].

The hair cells are supplied with many stereocilia on the top of them. As we see in figure 2.5, the stereocilia do not have the same length, and are connected with fine protein fiber strands, known as tip links, that cause a synchrony oscillation of stereocilia. During an oscillation, pushing stereocilia toward the longest stereocilium causes a tension on the tip links, and pushing them to the other direction causes the release of the tension. This can be seen in Figure 2.5. The tip links are connected with tiny ion channels, which open during the tension of the tip links, causing the K^+ ions to flow inside the hair cell depolarising it. Greater tension leads more channels to open causing a greater depolarising of the hair cell. This makes a correspondence between the mechanical and electrical pattern of vibration. For more details, see [ScNK11]

We will see that the damping nature of the electrical oscillation which hair cells exhibit with a certain resonance frequency is particularly important to us in this work. In the paper [FeFu99], Fettiplace and Fuchs, who made experiments on hair cells of a turtle, show that hair cells exhibit an electrical dumped oscillation with a certain resonance frequency in response to the mechanical vibration. The Experiments show also that there is little or no filtering of the acoustic stimuli, by the basilar membrane for example, before the hair cell mechano-electrical transduction. This is because the basilar membrane in turtles is not developed like the one in humans. This means that the hair cell of turtles play the major role in frequency analysis instead of the basilar membrane. Figure 2.6 shows two hair cells with different resonance frequencies labelled on them. It also shows an electrical resonance with a damped nature and a frequency equals to the resonance of the hair cell. The

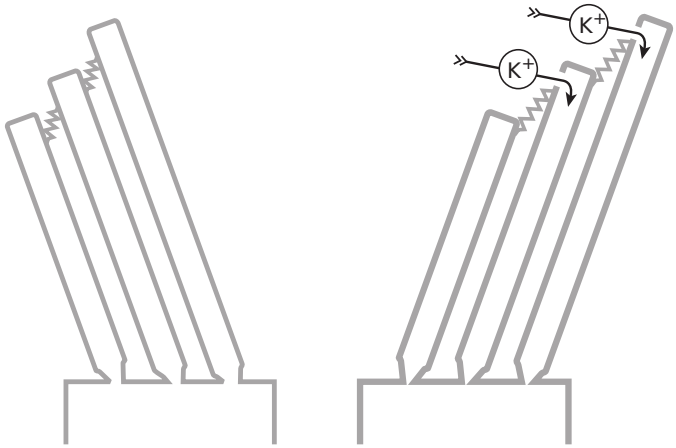

Figure 2.5: Transduction mechanism of the hair cell[ScNK11]

density of the ion channels as well as the length of the stereocilia determine the distinct resonance frequency of a hair cell. The higher the density of ion channels is and the shorter stereocilia are, the higher the resonance frequency is [FeFu99].

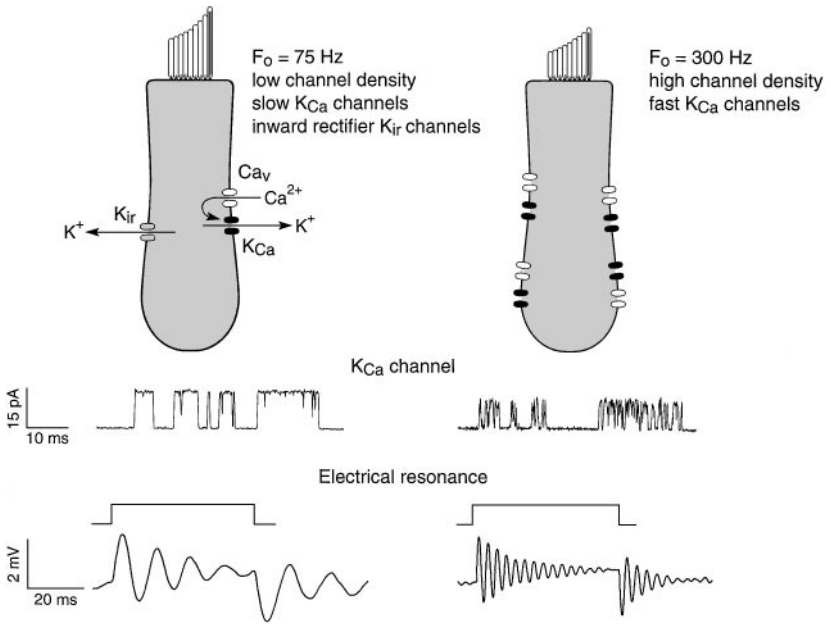

Figure 2.6: Schematic drawing of two hair cells from a turtle. Beneath each cell are shown a representative ion channel record and the electrical response [FeFu99].

3. Automatic speech recognition

Automatic speech recognition (ASR) is the translation of spoken to written words. Figure 3.1 illustrates a statistical speech recognition system, the most typical in practice. The voice of a speaker is recorded as the input of the system, and a sequence of words, a recognition hypothesis, is then computed as the output. The first component of a recognition system is the signal processing, which extracts the most relevant information for the speech recognition task, and omit unimportant ones. More details are in section 3.1.
The extracted information of one time window of the speech is formed in a so called feature vector. This is then the input of a component called decoder, which depends for its task on three components: dictionary, language model, and acoustic model. The dictionary contains all known words to be recognized, mapping them to phonemes or other word subunits (section 3.2). The language model holds knowledge about the possibility of a sequence of words in a certain language (section 3.3). Finally, the acoustic model tells how probable is a sequence of phones, subphones, or other word subunits given a sequence of words. The most probable word sequence can be found with graph search algorithms after the following formula:

$$\hat{W} = \underset{W}{argmax} P(X|W) \bullet P(W) \tag{3.1}$$

Where we search among all word sequences W for the one which maximizes the posterior probability, which is a probability of W given a sequence of acoustic observation $X = X_1 X_2 \dots X_n$; it is given by the Bayesian formula:

$$P(W|X) = \frac{P(X|W).P(W)}{P(X)} \tag{3.2}$$

We get $p(X|W)$ from the acoustic model, and $P(W)$ from the language model. $P(X)$ is a normalizing term that does not affect the search for the maximum, and hence can be ignored here.

3.1 Speech signal processing

There are many signal processing approaches to extract the information relevant to the speech recognition task from the audio signal. Linear Predictive Code (LPC)

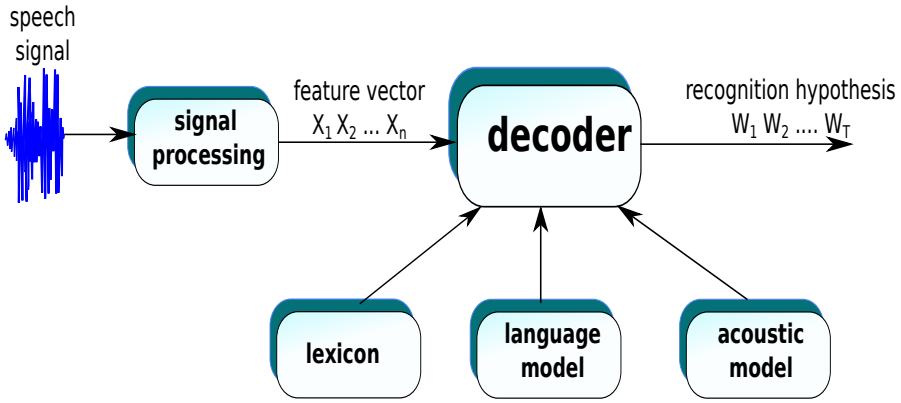

Figure 3.1: The block diagram of statistical speech recognition system

and Mel Frequency Cepstral Coefficients (MFCC) are examples of feature extraction methods.
LPC estimates the common speech parameters. It assumes that the vocal tract can be represented as a concatenation of lossless tubes and that the glottis produces a buzz (loudness + pitch). With this model, LPC can approximate the speech signal and extract only the information important to speech recognition.
MFCC on the other hand uses several steps for constructing the feature vector: First we calculate the Fast Fourier Transformation (FFT) of a windowed frame of the audio signal. Typically the windows have a duration of 16 ms, and a shift of 10 ms. A window length of 16ms offers the best compromise between temporal and frequency resolution. We pass the windowed frame to a filter-bank with triangle filters (shown in figure 3.2), Mel-scaled center frequencies (see section 3.1.2), and increasing bandwidth. Then, we calculate the log-energy at the output of filters, and calculate the Cepstrum explained in section 3.1.1. From the Cesptrum coefficients we take the first 13 as MFCC vector for one frame. We usually stack a number of vectors together to capture dynamic information (changes over time), or we calculate delta features explained in section 3.1.3 for the same purpose.

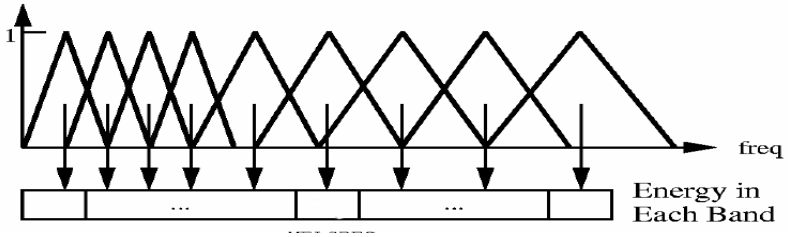

Figure 3.2: Mel scaled triangle filter bank used in MFCC [MuBE10].

3.1.1 Cepstral processing

After the source-filter model of speech production (see chapter 6 of [HuAH01]), we can consider the speech signal as in the following equation:

$$f_n = e_n * h_n$$

where f_n is the speech signal produced from a filtered excitation e_n; h_n is the impulse response of the filter. This model considers nasal and mouth configuration as a filter.

The excitation originates from the glottis in vocals and a white noise in consonants. For the Source-Filter separation, Cepstral processing convert the convolution in the equation into a sum. This is showed in the following equations [Schu95]

$$FT\{f_n\} = FT\{e_n\} \cdot FT\{h_n\}$$

$$logFT\{f_n\} = logFT\{e_n\} + logFT\{h_n\}$$

$$FT^{-1}\{logFT\{f_n\}\} = FT^{-1}\{logFT\{e_n\}\} + FT^{-1}\{logFT\{h_n\}\}$$

With the absolute value, we call $FT^{-1}\{log|FT\{f_n\}|\}$ real Cepstrum. In the Practice, we use Fast Fourier Transformation (FFT) or Discrete Cosine Transformation (DCT) (see [HuAH01]) instead of Fourier transformation.
After calculating the Cepstrum coefficients, we take those with low order that represent the macrostructures of the signal, which contain the formants information.

3.1.2 Mel-frequency-scale

The auditory system performs a frequency analysis of sounds. The cochlea acts like overlapping filters with different bandwidths, which are not placed linearly. The Mel-scale is linear under 1 KHz and logarithmic above that. This scale is more close to sensitivity of human ear. It can be approximated by [HuAH01]:

$$B(f) = 1125 \cdot ln(1 + f/700) \tag{3.3}$$

where f is a frequency to which a part of the basilar membrane resonate, and $b(f)$ is its bandwidth. Figure 3.3 illustrates the Mel scale together with the Bark scale and a uniform (linear) scale. As MFCCs are being calculated with Mel scale, BFCCs are based on Bark scale. In [ShPa03] it is shown that MFCCs and BFCCs yield similar performance, but with little advantage for both compared to the uniform scale. In this work, we use another scale called equivalent rectangular bandwidth (ERB, section 4.2)

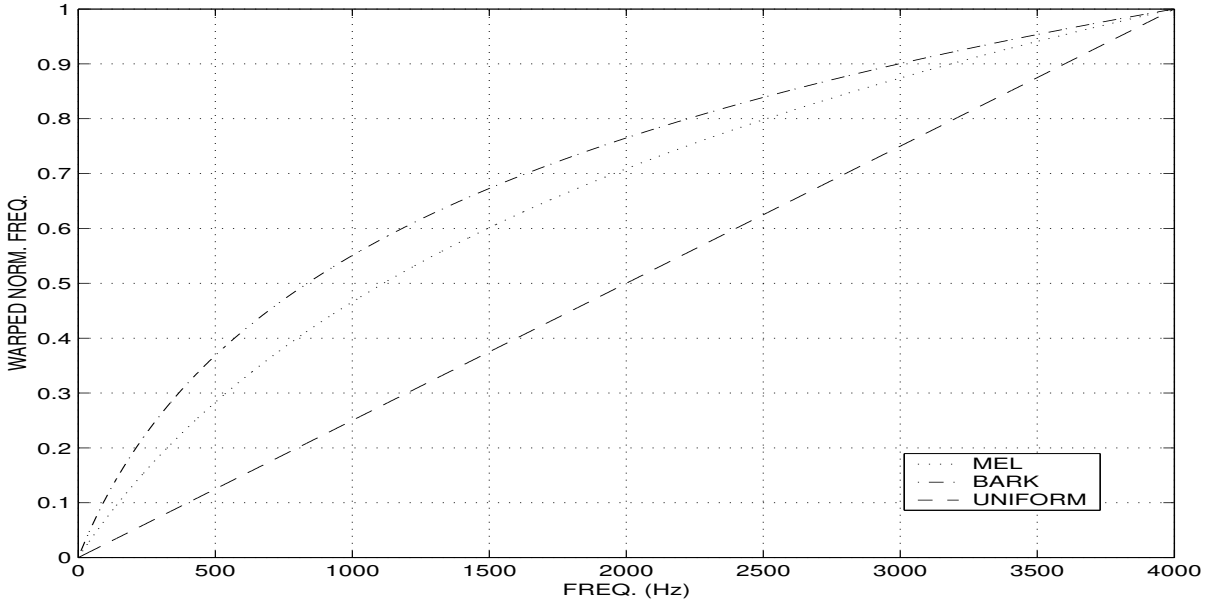

Figure 3.3: Mel scale compared with Bark scale and linear scale (uniform)[ShPa03]

3.1.3 Dynamic Features (delta coefficients)

The dynamic information contains temporal changes in spectra which play an important role in human perception (see [HuAH01]). Delta coefficients measure the change to the neighbour features. We append them usually to other static features like MFCCs. Different order of delta coefficients can be calculated as following:
1st-order delta: $\Delta c = c_{k+1} - c_{k-1}$.
2nd-order delta: $\Delta\Delta c = \Delta c_{k+1} - \Delta c_{k-1}$ where c_k are static feature vectors like MFCCs feature vectors.

3.2 Dictionary

A very important unit of any speech recognition is the dictionary. It contains a list of all words to be recognised by a system. The use of words as basic units for the classification is not efficient, because of the need of a very large training dataset to cover all the words and their contexts. The solution is then to use subunits such as phonemes or subphonemes in phoneme based system. These subunits make better use of the available training data, because subunits come more frequently in the training set than a whole word; therefore, we need smaller training data set for subunits based systems. The dictionary maps each word to its subunits, and may additionally contain alternative pronunciations of word. Dictionaries are created by experts or generated automatically.
In some languages, like Italian, the mapping of letters to sounds is close to 1:1. Here, it is possible to use the written letters as acoustical units. These units are called **graphemes** used in grapheme based systems.

3.3 Language model

In a speech recognition system, we use in addition to the acoustic model the information about the possibility of a given word sequence. The language model is represented by the term $P(W)$, which gives the priori probability of the sequence of words $W = W_1 W_2 \dots W_T$.
One way to estimate $P(W)$ is using a parser, which decides whether a certain word sentence follows a given grammar or not. Such parsers are used with programming languages as well. But by using this approach, $P(W)$ will be either 0 or 1 depending on whether the sentence is allowed by the grammar used or not. The first problem is that we need more information about how probable a sentence is. The second problem is that spoken languages do not necessarily follow grammars.
Another approach are stochastic language models (SLM). N-Gram is the most widely used SLM .

3.3.1 N-Gram Language Models

Since $P(W)$ tells how probable a word sequence W comes in a language, we can write it as[HuAH01]:

$$\begin{aligned} P(W) &= P(w_1, w_2, \dots, w_n) \\ &= P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)\dots P(w_n|w_1, w_2, \dots, w_n - 1) \\ &= \prod p(w_i|w_1, w_2, \dots, w_{i-1}) \end{aligned} \quad (3.4)$$

where $p(w_i|w_1, w_2, \ldots, w_{i-1})$ is the probability of the word w_i when it appears after the sequence $w_1, w_2, \ldots, w_{i-1}$, which is called the history of the word w_i. Since the training corpus only has a finite number of sentences, the use of a a so called n-gram language model helps estimate more robust probabilities, where it compute the probability of words by only using a history of length $n-1$ words. We call $p(w_i)$ unigram, $p(w_i|w_{i-1})$ bigram where the probability of w_i only depends on the previous word, and $p(w_i|w_{i-2}, w_{i-1})$ trigram. To estimate the trigram, we use the following equation [HuAH01]:

$$p(w_i|w_{i-2}, w_{i-1}) = \frac{C(w_{i-2}, w_{i-1}, w_i)}{C(w_{i-2}, w_{i-1})} \tag{3.5}$$

where $C(w_{i-2}, w_{i-1}, w_i)$ is count of the sequence w_{i-2}, w_{i-1}, w_i, and $C(w_{i-2}, w_{i-1})$ is count of the history: w_{i-2}, w_{i-1} in the corpus, which is a big collection of texts in a language.

3.3.2 N-Gram Smoothing

If the training corpus for n-Gram calculation is not large enough, the count of some word sequences could be zero This leads to the fact that the nominator of equation 3.5, and the Term $P(W)$ could be zero. This would mean that this sequence is impossible and will be excluded. Some other sequences would be assigned a very small probability and as a result lose weight in the searching.
There are many solutions which depend on a better distribution of the probability mass to be more robust for the unseen data. **Deleted Interpolation Smoothing** interpolates, for example, a bigram and a unigram as follows[HuAH01]:

$$P_I(w_i|w_{i-1}) = \lambda p(w_i|w_{i-1}) + (1-\lambda)p(w_i) \tag{3.6}$$

where $(0 \leq \lambda \leq 1)$. This equation says that the probability of a sequence of two words would not be zero if the the first word is probable. Other techniques like **Backoff Smoothing** could be read in [HuAH01].
It is also worth to mention the **Adaptive Language Model**, where the idea is to create a local dynamic n-Gram model online from the so far dictated speech. We interpolate a static (offline trained) h-Gram with it, for example, the online bigram to give the new dynamic (cached) n-Gram. With more dictating, we weight the cache bigram more, because it is going to be more informative.

3.4 Acoustic model

We have seen in equation 3.1 that we need the term $P(X|W)$, which is the probability of an acoustic observation $X = X_1X_2 \ldots X_n$ given a sequence of words W. As the dictionary map words to subunits, i.g phonemes in phoneme based system or graphemes in grapheme based system, observation vectors then work at the level of this subunits. The best statistical method to find the probability of an observation is the hidden Markov model, which we will see closer in the section 3.4.1. For hidden Markov model, we use Gaussian mixture model to describe the likelihood of the acoustic subunits after training precisely. To do that, we need a so called Estimation Maximisation (EM) algorithm for Gaussian mixture model explained in section 3.4.2.

3.4.1 Hidden Markov models

The Hidden Markov Model (HMM) is a statistical method of characterizing the observed data sample of a discrete-time series [HuAH01]. As known by now, we extract the feature vector from discrete windowed time frame, and become a discrete-time series of speech frames; therefore, HMM is very suitable to model speech signal, and this is proofed empirically.
The word "hidden" characterizes a Markov chain with the property of generation of the output observation in any state, not in a certain one. Now it is important to mention the **Markov assumption**: Let $X = (X_1, X_2, \ldots, X_n)$ be random variables. We have the following after Bayes' rule:

$$P(X_1, X_2, \ldots . X_n) = P(X_1) \prod_{i=2}^{n} P(X_i | X_1, X_2, \ldots X_{i-1}) \tag{3.7}$$

The Markov assumption says that the probability of a random variable at a given time depends only on the value at the preceding time. Hence equation 3.7 becomes

$$P(X_1, X_2, \ldots . X_n) = P(X_1) \prod_{i=2}^{n} P(X_i | X_{i-1}) \tag{3.8}$$

The Markov assumption allows us to model Markov model as finite state process, because we need the probability of transition between only two neighbour states independent of the previous states. Figure 3.4 illustrates a HMM with 3 states. The topology of a HMM used in speech recognition is the left-to-right topology which have only forward edges. This is logical since the speech makes only a forward progress. Figure 3.4 clears also the 5 tuples of the **formal definition of HMM** [HuAH01]:

- $O = \{o_1, o_2, \ldots, o_M\}$: the output observation alphabet In the speech recognition, it is the different acoustic realisations (the feature vectors) of the phonemes, subphonemes or graphemes depend on the system at hand.
- $\Omega = \{1, 2, \ldots N\}$: a set of states. Each state s_t at time t are a subword emites different acoustic realisation (an observation symbol o_k).
- $A = \{a_{ij}\}$: a transition probability matrix, a_{ij} is the probability of taking transition from the state i to the state j, $a_{ij} = P(s_t = j | s_{t-1} = i)$. In our application, for example, it is the probability to say a phoneme after another.
- $B = \{b_i(k)\}$: an output probability matrix, where $b_i(k)$ is the probability of emitting a symbol o_k when a state i is entered: $b_i(k) = P(X_t = o_k | s_t = i)$, where $X = X_1, X_2 \ldots X_t, ..$ the observation output of the HMM. In our application its the probability of a certain acoustic realisation of a phoneme or grapheme.
- $\pi = \{\pi_i\}$: an initial state distribution where $\pi(s_0 = i)$.

We have the following constraint since we deal with probabilities: $a_{ij} \geq 0$, $b_i(k) \geq 0$, $\pi_i \geq 0 \ \forall i, j, k$ and

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{3.9}$$

$$\sum_{k=1}^{M} b_k = 1 \tag{3.10}$$

$$\sum_{i=1}^{N} \pi_i = 1 \tag{3.11}$$

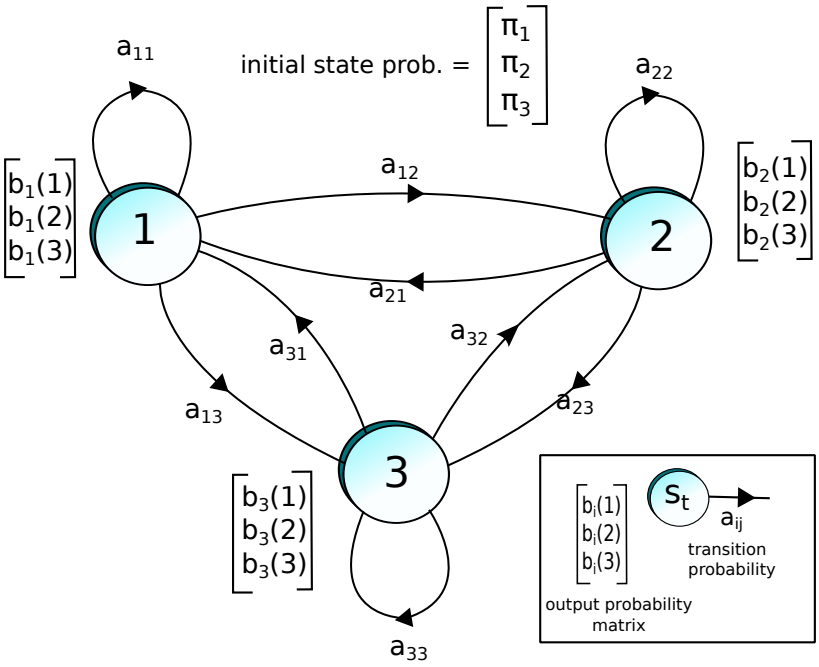

Figure 3.4: HMM diagram illustrates the five tuples

First-order HMM is a HMM with two assumptions. First the Markov assumption seen above, and second the *output independence assumption*, which says that emitting an observation symbol at time t depends only on the state s_t and is conditionally independent of the past observation. This constraint of memory reduces the number of parameter to be estimate, and does not affect the performance in practice.
There are three basic problems when dealing with HMMs:

- **The evaluation problem**:
  What is the probability that a HMM generate an observation. Solution: *The Forward Algorithm.*

- **The decoding problem**:
  What is the most likely state sequence in a HMM that generate a certain observation. Solution: *The Viterbi Algorithm*

- **The learning problem**:
  How can we adjust the HMM parameters to maximize the probability of emitting a certain observation. Solution: *Baum-Welch Algorithm* or known as *Forward-Backward Algorithm.*

The algorithm details are beyond the scope of this work; for further reading see [HuAH01].

It is important to distinguish between discrete, continuous and semi continuous HMMs.

- **Discrete HMMs** use a discrete output probability distribution b_j. However, the output vectors x itself are from continuous space. We solve the problem of assigning a continuous range of vectors discrete probabilities, by quantizing the observation vector to a vector $\hat{x}_k$,called codeword, with a minimum distortion (minimum distance), where there is a collection (called codebook) of pre-existent indexed vectors $\hat{x}$ have the pre-calculated probability $b(k)$.
- **Continuous HMMs** use a continuous density for the output probability distribution. We choose usually Gaussian mixture model (section 3.4.2).
- **Semicontinuous HMMs** is a bridge between the discrete and continuous HMM. We use the discrete output probability b_j as weights for a Gaussian mixture to calculate the output probability of $b_j(x)$ of a vector x:

$$b_j(x) = \sum_{k=1}^{M} g_j(k) f(x|o_k) = \sum_{k=1}^{M} N(x, \mu_k, \Sigma_k) \tag{3.12}$$

  where o_k is the kth codeword. The codeword contains a mean vector and a covariance matrix. The codebook is shared from many models which weights the Gaussian densities with own weights to become the probability value.

3.4.2 Estimation Maximization(EM) for Gaussian mixtures

Gaussian mixtures are the weighted sum of many Gaussian density functions (Gaussian bells). The mixture can approximate any density more precisely as we use a large number of Gaussian density. The formal equation is given as follows:

$$P(x|c, \{\mu, \Sigma\}) = \sum_{\nu=1}^{N} c_\nu N(x|\mu_\nu, \Sigma_\nu) \tag{3.13}$$

where x is a random variable and c are the vector of weights; $N(x|\mu_\nu, \Sigma_\nu)$ denote a single Gaussian density function with the means vector μ and the covariance matrix Σ. Figure 3.5 shows a Gaussian mixture with about 10 weighted Gaussian bells. We reach such forms with Estimation Maximization (EM) algorithm, runs on the training data.
The **(EM) algorithm** is a statistical method to find a likelihood where a distribution $P(x, u|\theta)$ is to estimate. Whereas values of the random variable x are observable, values of u are not; θ are the parameters to estimate. For Gaussian mixtures the parameters are vectors of means, covariance matrices and weights. The unobservable u inform us which data item belongs to which class (Gaussian bell), which is not known. To be able to apply maximum likelihood estimator that maximises the term

$$\pounds_{EM}(\theta) = logP(x|\theta) = log \int_u P(x, u|\theta) du \tag{3.14}$$

we need to have u or θ to be known. For this reason the EM algorithm optimises the parameter iteratively. It iterates over two steps:

- **1.Expectation step (E step):** Calculates the expectation over all data (x,u) under the current estimate of parameter θ

$$Q = (\theta, \widehat{\theta}) = E[logP(x, u|\widehat{\theta})|x, \theta]. \tag{3.15}$$

- **2.Maximization step (M step):** Finds the new parameter set:

$$\theta^{i+1} = \underset{\theta}{argmax} Q(\theta^i, \theta) \tag{3.16}$$

with initial parameter θ^0

For further reading see [Schu95].

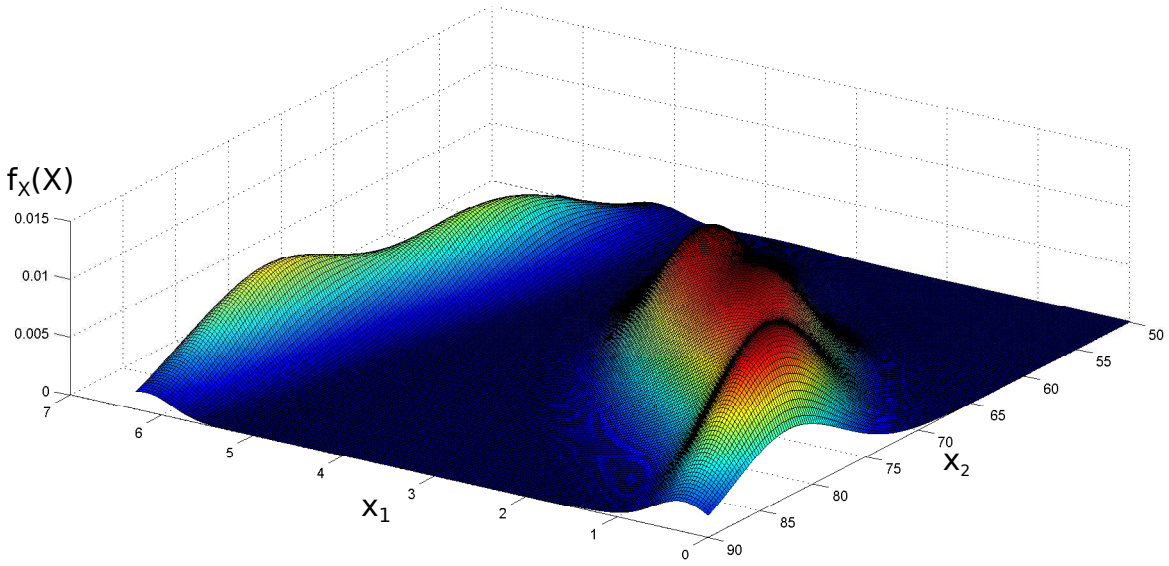

Figure 3.5: A Plot of a 2 dimensional Gaussian mixture distribution

3.5 Evaluation of ASR systems

It is important to have a measure of evaluation for ASR system performance in order to assess the performance, and to compare different systems. This helps to develop the system and to increase the performance.
The three typical types of word recognition errors are[HuAH01]

- **Substitution:** an incorrect word was substituted for the correct word.
- **Deletion:** a correct word was omitted in the recognized sentence.
- **Insertion:** an extra word was added in the recognition sentence.

Which help to form the equation of the *Word Error Rate* (WER):

$$Word\ Error\ Rate = 100\% \times \frac{Subs + Dels + Ins}{No.\ of\ words\ in\ the\ correct\ sentence} \tag{3.17}$$

where $Subs$ is the number of substitutions, $Dels$ the number of deletion, and Ins is the number of insertion.
The *Word Error Rate* (WER) is a widely used and typical measurement for the performance of a speech recognition system.

As an example of calculating the WER, Suppose we have the following two sentences:
Reference: my love is automatic speech recognition.
Hypothesis: mail is automatic peek race condition.
where the first is what is being uttered (the reference) and the second is what the recognition system recognized (the hypothesis). We notice that in the Hypothesis the words: "my", "speech", and "recognition" are substituted by the words "mail", "peek" and "race" so we have 3 substitution. The Word "love" is deleted and the word "condition" is inserted, so we have one deletion and on insertion. Then we have $WER = 100\% \times \frac{3+1+1}{6} = 66.66\%$, which we consider as a high value.
We evaluate the system by generating hypothesises from a *test-set*, which empirically need to be more than 500 sentences and are from 5 to 10 different speakers to reliably estimate the recognition error rate[HuAH01]. We have also a so called *dev-set*, which are unseen data in the training, for tuning the parameters at the development stage.

4. Synchronized Damped Oscillator Cepstral Coefficients

Synchronized Damped Oscillator Cepstral Coefficients or shortly SyDOCCs is based on a new feature extraction method, performed in [MiFG13], for a robust speech recognition. The idea behind SyDOCCs extraction is modeling some components of the human auditory system. The block diagram of figure 4.1 illustrates the steps used to perform the feature extraction. At first, we filter the speech signal using a high pass filter called pre-emphasis filter (section 4.1). Then, we use a gamma-tone filter-bank, which models the way the basilar membrane analyses the signal to its frequency components. As we will see in section 4.2, gamma-tone filter bank has an advantage of smooth filters compared to sharp pick filter used for MFCC. Next component in figure 4.1 is the synchronized forcing functions (section 4.3). They shift the output of the gamma-tone filter bank to become in phase and then perform a product to amplify the correlated components and suppress the uncorrelated parts, which we suppose to be noise. After synchronizing, damped oscillators, which model hair cell, amplify the frequencies of the input signals equal to their resonance frequencies, and suppress all others; see section 4.4 for more details. Modulation filter, at the next step, gives out the envelope of the input signals. Modulation filter and power computation are described in section 4.5. Before the last step, we perform a root compression, which has advantages of energy compaction, and robustness with noise, compared with logarithmic compression used in MFCC (see section 4.6). The last component is the calculation of the cepstral coefficients by calculation the discrete cosine transformation and the dynamic features delta, discussed in chapter 3; so we will not talk about it any further in this chapter.

4.1 Pre-emphasis filter

Pre-emphasis filter is a high-pass filter used to emphasis high frequencies, i.e. to amplify high frequency formants. Figure 4.2 illustrates frequency response of a pre-emphasis filter. Pre-emphasis filter can be given as a first order finite impulse response (FIR) filter:

$$y[n] = x(n) - \alpha \cdot x(n-1) \tag{4.1}$$

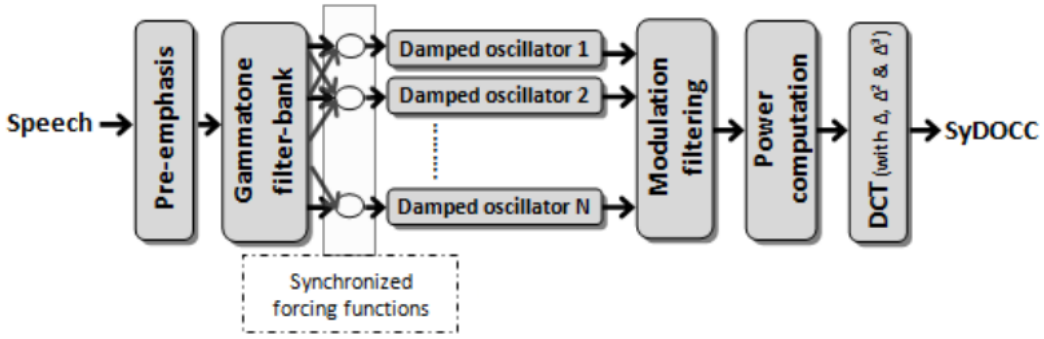

Figure 4.1: Block diagram of the synchronized damped oscillator based features[MiFG13]

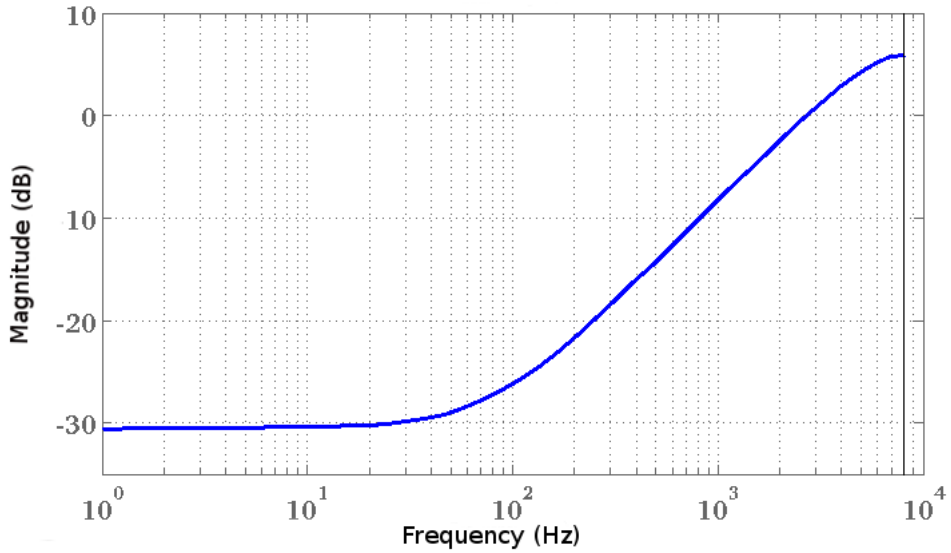

Figure 4.2: Amplitude response of the pre-emphasis filter

where x(n) is a time discrete input speech signal, and α is a real value coefficient. We get a high pass filter when $\alpha < 1$. The Z-transformation of Equation 4.1 gives the following:

$$Y(z) = X(z) - \alpha z^{-1} X(z) \Rightarrow$$
$$Y(z) = (1 - \alpha z^{-1}) X(z) \quad (4.2)$$

In the implementation, we use the Matlab function $filter(a, b, x)$, where, after the documentation of this function, a is the coefficients vector of the denominator, and b is the coefficients vector of the numerator of the following equation:

$$Y(z) = \frac{b(1) + b(2)z^{-1} + \cdots + b(nb+1)z^{-nb}}{1 + a(2)z^{-1} + \cdots + a(na+1)z^{-na}} X(z). \quad (4.3)$$

We get the coefficients vectors $a = [1]$, and $b = [1, -\alpha]$ by comparing equations 4.2 and 4.3.

4.2 Gamma-tone Filter-bank

We have seen in section 2.2.1 that the cochlea operates as a kind of mechanical frequency analyser. We can consider the cochlea as a set of filters, each based on a piece of the basilar membrane that vibrates maximal (resonate) in response to a certain frequency. We will call this the center frequency of the filter. These cochlear filters are spaced approximately logarithmic after a scale called equivalent rectangular bandwidth (ERB), derived by Glasberg and Moore [GlMo90]. At moderate sound levels, the ERB in Hz is given analytically in [GlMo90]:

$$ERB(f) = 24.7(4.37F + 1) \quad (4.4)$$

where $ERB(f)$ is the bandwidth in Hz and F is the center frequency in kHz. Figure 4.3 shows the ERB scale approximations.

Now we come to the gamma-tone filter-bank, which is a set of linear filters, gives an approximation of cochlear filters; It is published in [PNSHR87] by R. Patterson, I. Nimmo-Smith, J. Holdsworth and P. Rice in 1987. Figure 4.4 illustrates the frequency responses of a gamma-tone filter-bank, and the distribution of center frequencies with respect to these on the basilar membrane.

For a better understanding of how this filters work, we are going to see their impulse responses (see figure 4.5). The impulse response of each filter is a sinusoid with a certain center frequency. The sinusoid is in turn windowed with a gamma function, which approximates a shape of a bell. Figure 4.6 illustrates filtering of FM sweep or chirp signal (a signal with increasing frequency) with a gamma-tone filter. The output has a maximum amplitude in the same place where the frequency of the FM sweep is the same as the center frequency of the gamma-tone filter. In this way, we can analyse a signal with whole filter-bank with different center frequencies like the cochlea does.

An advantage of using gamma-tone is the smooth frequency response of its filters compared to the sharp peaks of the triangular filters used in MFCC as we saw in chapter 3. The sharp peaks makes the filter sensitive to small changes of frequency,

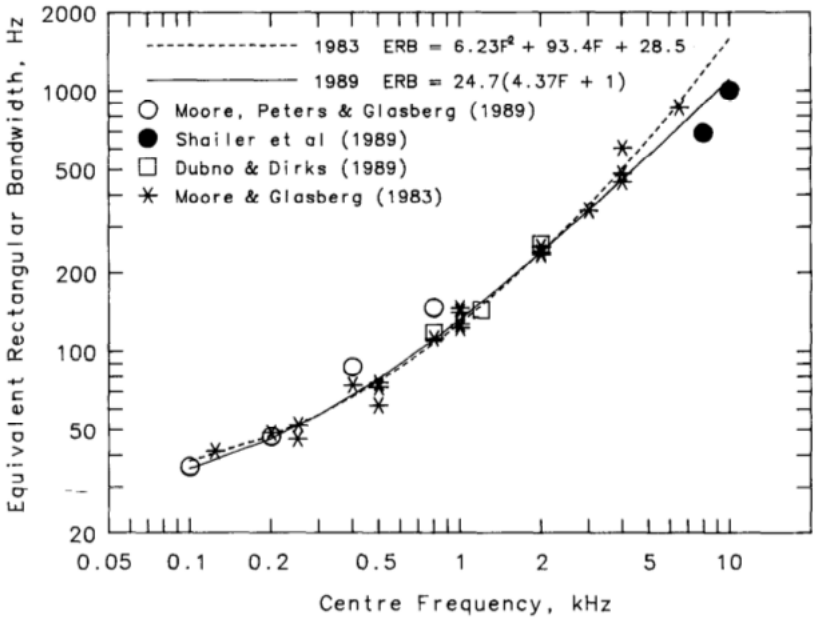

Figure 4.3: ERB scale approximations [GlMo90].

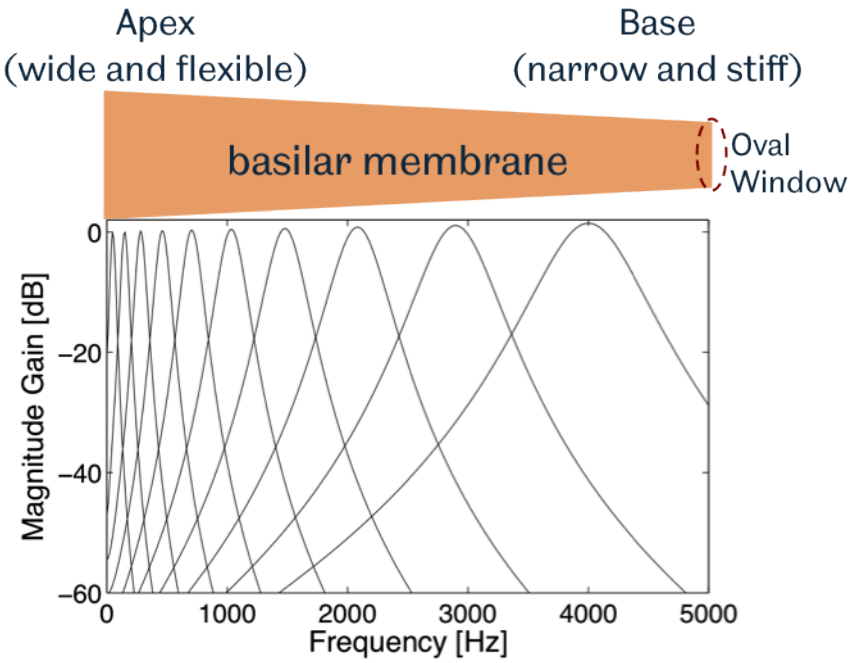

Figure 4.4: Frequency responses of a gamma-tone filter-bank with ten filters whose center frequencies are equally spaced between 50 Hz and 4 kHz on the ERB-rate scale [Ma].

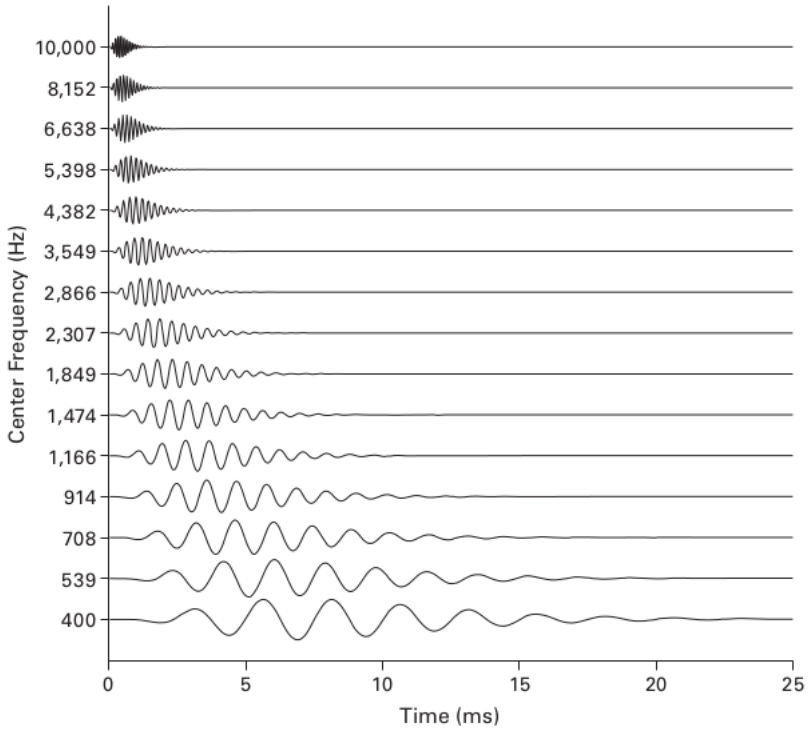

Figure 4.5: a gamma-tone filter-bank can serve as a simplified model of the basilar membrane[ScNK11].

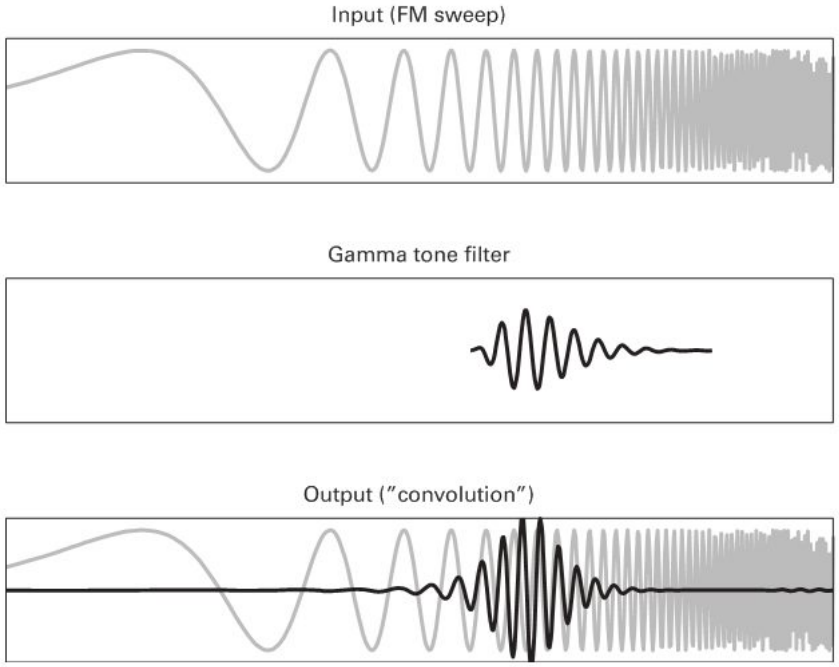

Figure 4.6: An FM sweep(chirp signal) filtered by a gamma-tone filter[ScNK11].

to which (changes) we are perceptually insensitive [RaAS].

In the implementation, we choose after [MiFG13] 50 channels for gamma-tone filter-bank, and a range of center frequencies between 200 Hz and 7 kHz for 16 kHz sampling rate. For 8 kHz sampling rate, we choose 40 channels, and center frequencies between 200 Hz and 3750 Hz.

4.3 Synchronized forcing function

The outputs of gamma-tone filters (section 4.2) give information about how much of each center frequency is contained in the input signal. Suppose that the outputs of the filters with center frequency f_i would be $F_{e,i}[n]$ for $i = 1, \cdots, m$, where m is the number of gamma-tone filters, then we call the result of the following equation:

$$F_{syn,i}[n] = F_{e,i-1}[n - \Delta_{i,i-1}] \cdot F_{e,i}[n] \cdot F_{e,i+1}[n - \Delta_{i,i+1}] \tag{4.5}$$

the synchronized output of the filter i, where $\Delta_{i,j}$ is the time lag between $F_{e,i}[n]$ and $F_{e,j}[n]$. It expresses the similarity shift; We will see more about this later in this section.
Equation 4.5 shifts three adjacent signal so that they have the most similarity, and then multiplies them all. In the first and second plots of 4.7, we can see the shift result. In the third plot, we see the resulting signal of the multiplication of synchronized signals in blue, and of unsynchronized in red. The idea of synchronizing is to approximate the cross-correlation coefficient of adjacent outputs. The result preserves the sinusoid components of the output and suppresses the noisy uncorrelated components [MiFG13].

To obtain the time lag $\Delta_{i,j}$ between the output of channels i and j we use the Average Magnitude Difference Function (AMDF). To do this we calculate first:

$$\gamma_{i,j}[k] = \sum_m |F_{e,i}[n+m]w[m] - F_{e,j}[n+m-k]w[m-k]| \tag{4.6}$$

where $w[m]$ is a rectangular window with the duration of four periods: $4f_s/f_i$, and f_s is the sampling rate of the signal. Then we take the minimum

$$\Delta_{i,j} = \min_k \gamma_{i,j}[k]. \tag{4.7}$$

When we shift the signal $F_{e,j}$ by $\Delta_{i,j}$, we would have the most similarity of it with $F_{e,i}$, because $\Delta_{i,j}$ is equal k, which makes the absolute of the difference of the first signal and the second signal shifted by k at its minimum.

4.4 Damped oscillator

As we saw in section 2.2.2, the hair cell demonstrates a damped oscillations. In SyDOCC features, we use the synchronized outputs of a gamma-tone filter-bank as a forcing function ($F_e[n]$) for the oscillators. The equation

$$x[n] = \frac{(2\zeta\Omega_0^2)F_e[n] + 2(1+\zeta\Omega_0)x[n-1] - x[n-2]}{(1+2\zeta\Omega_0+\Omega_0^2)} \tag{4.8}$$

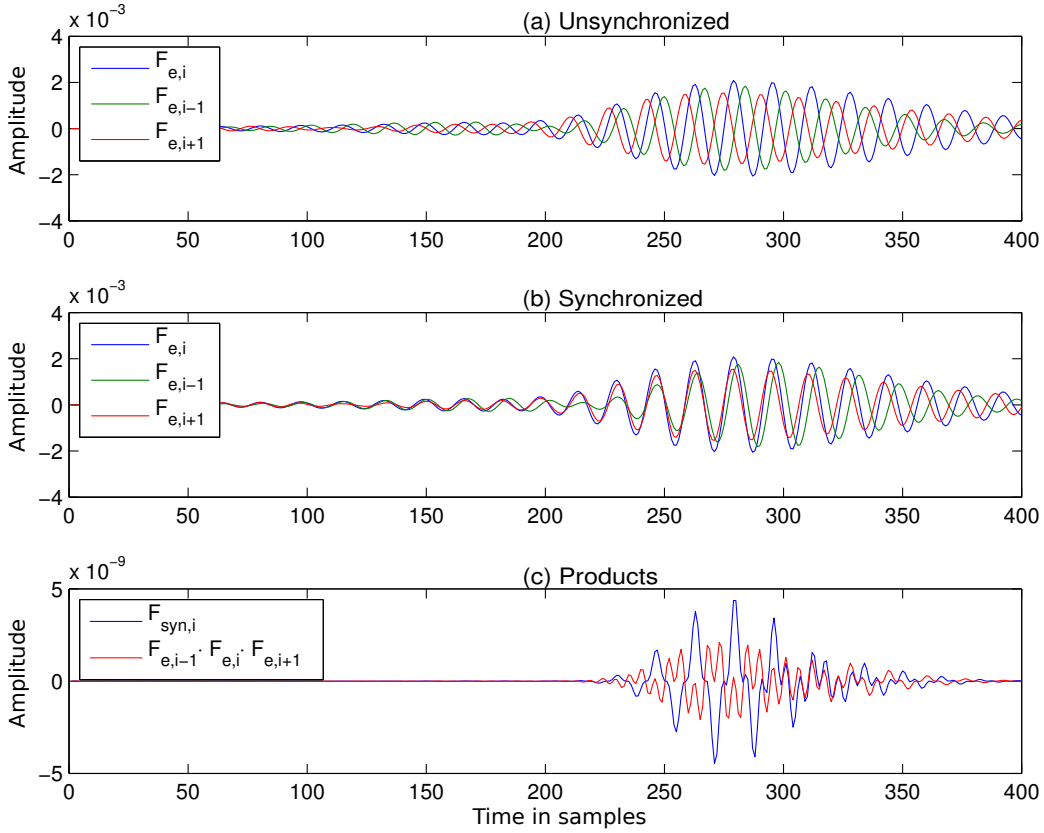

Figure 4.7: (b) Unsynchronized signals. (b) signal after apply the synchronizing. (c) The signal in blue is the product of the signals in (b), in red is the product of signals in (a)

is derived in [MiFG13] from the physical equations of a simple and damped harmonic oscillator, where ζ is called the damping ratio, and $\Omega_0 = \omega_0 T$, where ω_0 is the undamped angular frequency of the oscillator, and $T = 1/f_s$, where f_s is the sampling rate. When $\zeta < 1$ we get damped oscillations as we see in Figure 4.8.

Damped oscillators amplify the frequencies equal to their resonance frequency and suppress all others with different grades. Figure 4.9 illustrates the amplitude response diagram, where the pick is at the frequency $100Hz$, which represents in this example the resonance frequency. After being synchronized, each output of the gamma-tone filter-bank channels stimulates a damped oscillator whose resonance frequency is the same as the center frequency of that channel. The output of the damped Oscillator, as we see in figure 4.10, smooths the signal, and keeps mainly the resonance frequency undamped. The phase shift of the output has no effects in our task.

To use the equation 4.8 in the implementation, we use ,like in section 4.1, the Matlab function $filter(a, b, x)$. The Z-transformation of 4.8 gives the following:

$$X[z] = \frac{(2\zeta\Omega_0^2)F_E[z] + 2(1 + \zeta\Omega_0)z^{-1}X[z] - z^{-2}X[z]}{(1 + 2\zeta\Omega_0 + \Omega_0^2)} \tag{4.9}$$

We bring equation in a form to be compared with the transfer function of the *Matlab* filter function 4.3 to extract the coefficients vectors, where here $X[z]$ is the

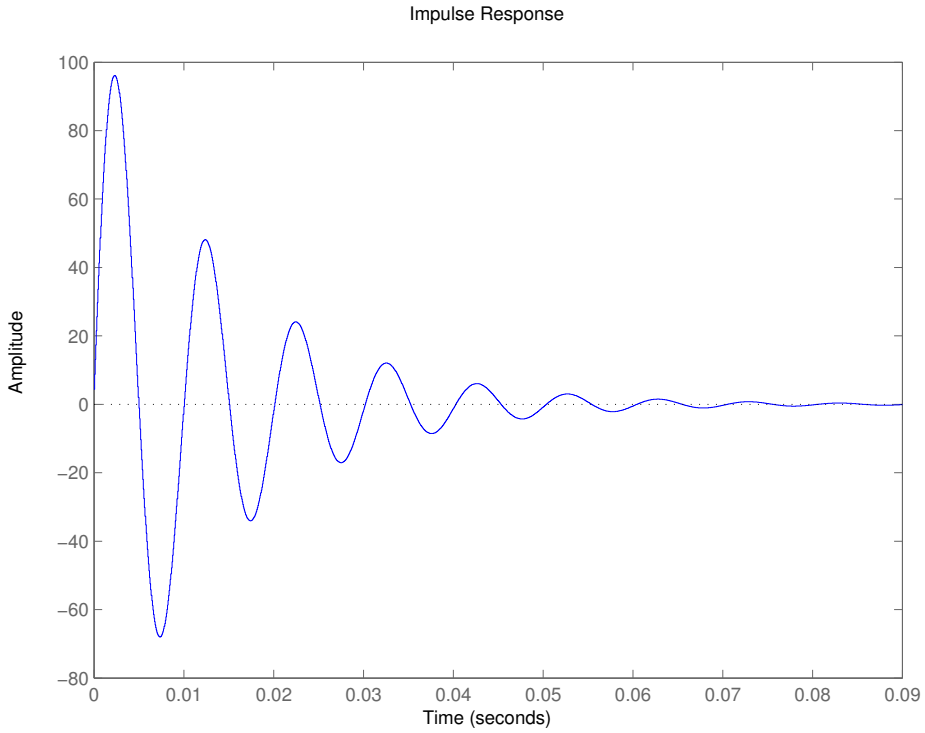

Figure 4.8: Impulse response of a damped oscillator with a damped ratio $\zeta = 0.09$ and center frequency $\Omega = 100Hz$

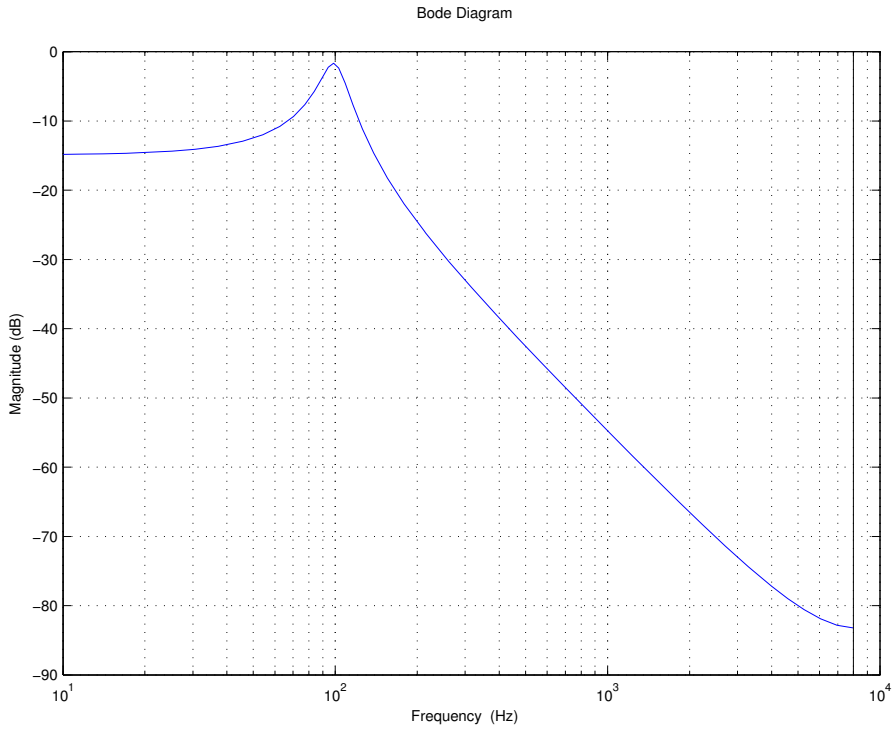

Figure 4.9: Amplitude response of a damped oscillator with a damped ratio $\zeta = 0.09$ and a center frequency $\omega = 100Hz$, showing a pick at $100Hz$

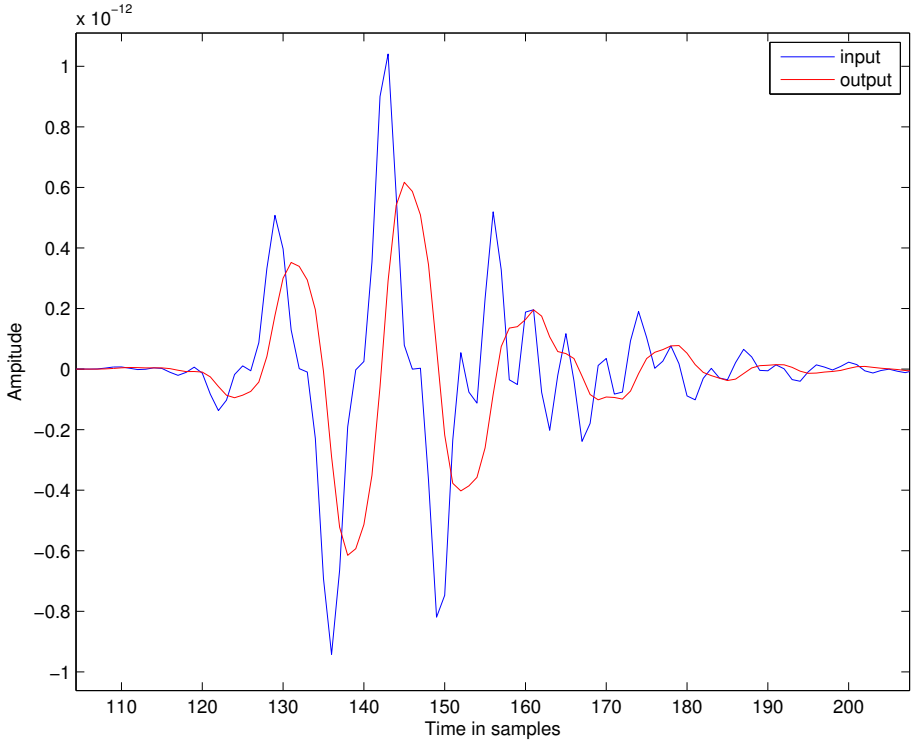

Figure 4.10: input and output of a dumped oscillator

Z-transformation of the filter output $x[n]$ and $F_E[z]$ is the Z-transformation of $F_e[n]$. From

$$X[z] = \frac{(2\zeta\Omega_0^2)}{(1 + 2\zeta\Omega_0 + \Omega_0^2) - 2(1 + \zeta\Omega_0)z^{-1} + z^{-2}} F_E[z] \tag{4.10}$$

we get the coefficients: $a(1) = (1 + 2\zeta\Omega_0 + \Omega_0^2)$, $a(2) = -2(1 + \zeta\Omega_0)$, $a(3) = 1$, and $b(1) = (2\zeta\Omega_0^2)$

4.5 Modulation filtering and power computation

In the block diagram 4.1, we see modulation filter Block comes after the damped oscillators. Its task is to give a smoothed envelop of the output of the damped oscillators. We see in figure 4.11 such an output and its modulation filtered signal with a little phase shift. The idea behind modulation filtering is to keep the macro structure, and to make the power computation, which will follow directly, not sensitive to the high frequency noise. We calculate the power of the signal using the following:

$$P = \sum_n (x[n])^2 \tag{4.11}$$

4.6 Root compression

After calculating the power of the signal, we do then the root compression $(\cdot)^\alpha$ with $0 < \alpha < 1$. As described in [MiFG13], we choose for SyDOCC $\alpha = 1/7$. Paper [RaAS] revalidate a previous result shows that root compression is better than logarithmic compression for noise robustness. Figure 4.12 shows some performance

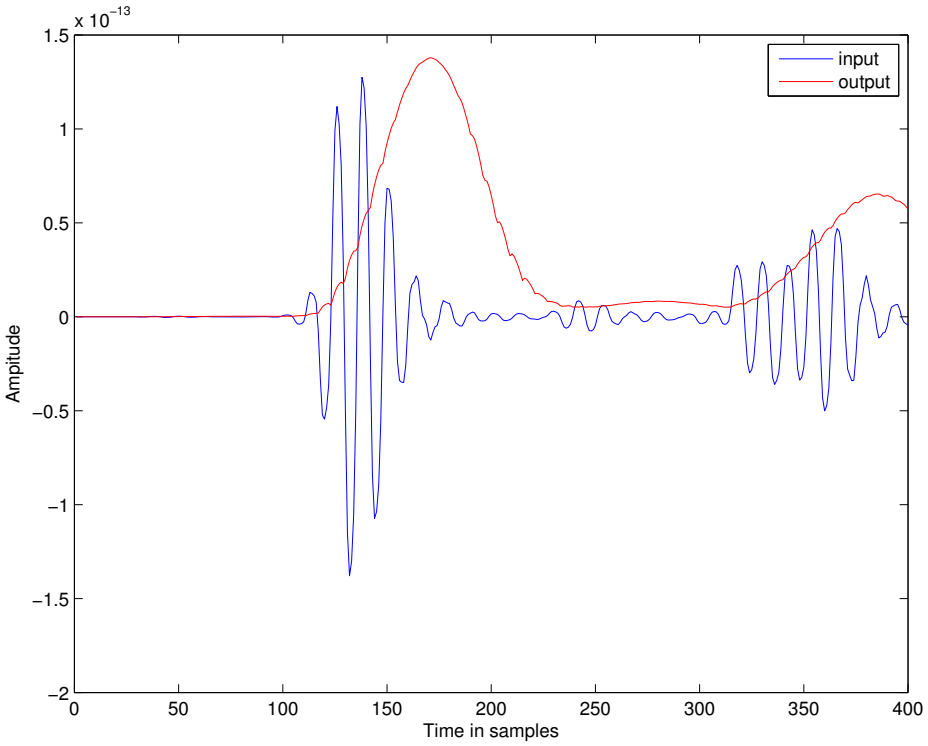

Figure 4.11: input and output of the modulation filter

results of a classifier using MFCC features once with root and once with log compression.
Paper [RaAS] performs another experiment, which shows that root compression followed by Discrete Cosine Transformation (DCT) leads to better compaction of energy. The experiment compares the results of reconstruction error of a signal amplitude, which compressed once with log and once with root. The results at both times are DCT transformed. The experiment, as seen in figure 4.13, shows that even with a few number of coefficients, the reconstruction error in the case of root compression is very small compared to log compression.

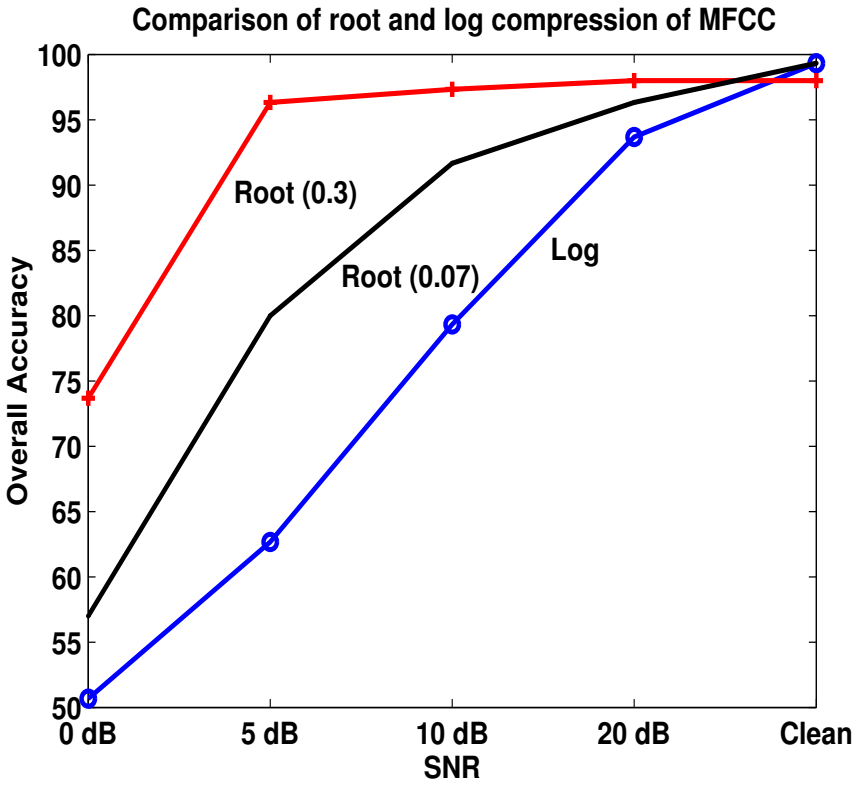

Figure 4.12: A classifier performance using MFCC features with log or root compression, for root is $\alpha = 0.3$ and $\alpha = 0.07$.[RaAS]

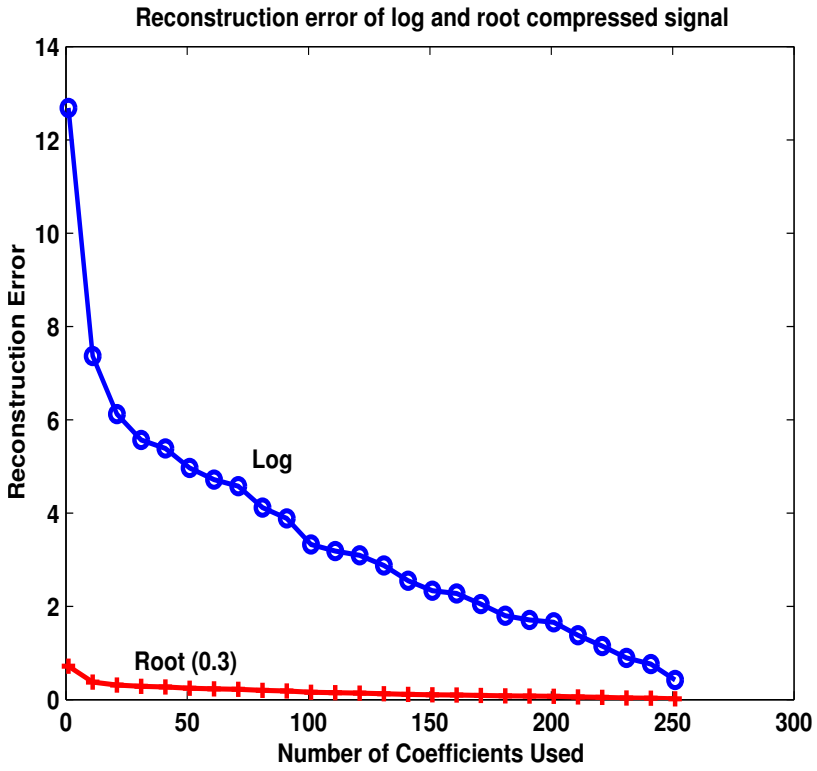

Figure 4.13: A comparison between log and root compression followed by DCT in reconstruction of signals with n coefficients. [RaAS]

5. Evaluation

Now we discuss the evaluation results of the speech recognition system with SyDOCCs. We have made many experiments to tune SyDOCCs parameters. The value of the damping ratio, as we have seen in section 4.4, determines how much the oscillators are damped, the smaller the damping ratio is the more the frequencies are damped and the more formed peak we get at the resonance frequency of the oscillator. The other parameter is the resonance frequency of the damped oscillator itself. After [MiFG13], the resonance is given as user defined, and after the earlier version, it is given as the center frequency of the corresponding gamma-tone filter. One of our experiments shows that the results for the resonance 200 Hz for all damped oscillators with damping ratio 0.9 is better than the resonance equal to the center frequency of gamma-tone filter bank with damping ratio 0.09. Another experiment without using the damped oscillators shows worse results than the one with damped oscillators. Other free parameters are the number of gamma-tone filter-bank channels and the bandwidth. For the experiments we chose 50 channels for 16 kHz data sampling rate with a bandwidth from 200 Hz to 7 kHz, and 40 channel for 8 kHz with the bandwidth from 200 to 3750 Hz. For splitting the signal into frames, we have other free parameters: window duration and its overlapping . We have tested with 16 ms and 25 ms window duration with 10 ms overlapping in the both cases. The results are a little better for the duration 25 ms.
Following sections describe the experiments and the results of comparison between SyDOCCs and MFCCs as baseline.

5.1 Experiments with Italian

We have performed experiments with Italian language data, where the training set are 70 hours recordings of broadcast news from Euro-news channel. The test-set is a relatively small data set with about 24 minutes recordings from 12 speakers.
The recordings for the training set and test set are direct and do not contain noise. We use the word *clean* to denote this set. To simulate the experiments in a noisy environment, we mix street noise and white noise with this set. We used a grapheme based system with N-gram language model trained on the ECI corpus (about 3

million sentences), the transcription of the training, and webcrawl text (a collection of text from the world wide web). The system is bootstrapped using a flatstart approach, which starts a system with randomly chosen weights for the Gaussian mixture models. We first build a context-independent system, and then use its labels to build a context-dependent system with 6000 Gaussian models, where preliminary experiments have shown that this number of models yields to the best performance.

5.1.1 Comparisons MFCC with SyDOCC

As we see in table 5.1 and its visualisation in figure 5.1, we perform three comparisons between SyDOCCs and MFCCs. In both cases, we stack the features using 15 adjacent feature vectors together to capture the dynamic information. In the first comparison, we consider the experiments with a clean training set and clean test set. Then we use a test with an additive street noise. We noticed that MFCCs show better performance at both experiments, however the performance difference is smaller with the noised test set.
In the third comparison, we considered the experiment with a training set with additive white noise and the same clean test set. The signal noise ratio (SNR) is about 9.38 dB. The performance of SyDOCC is much better than MFCC. We will see this phenomena in another experiment.

training	test	MFCC	SyDOCC
clean	clean	26.2%	30.1%
clean	with street noise	36.3 %	38%
with white noised	clean	80.6 %	73.8%

Table 5.1: Comparison of SyDOCCs with MFCCs for the language Italian. The percentage values are the WER of the best results

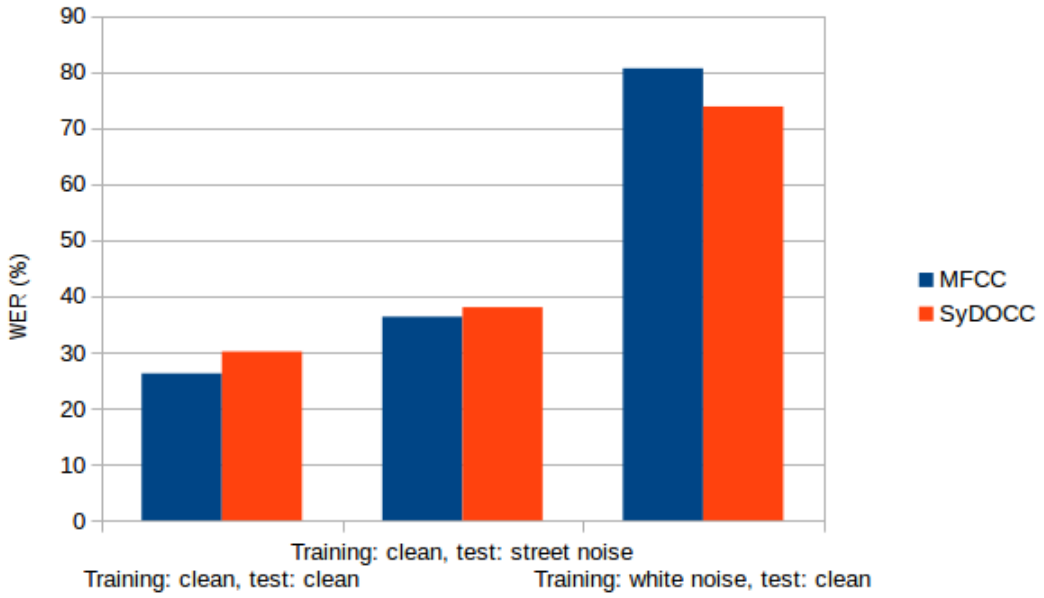

Figure 5.1: Comparison of SyDOCC and MFCC for the language Italian.

5.1.2 Comparison of MFCCs with the combination MFFC and SyDOCC

In these experiments, we compare the combination (MFCC ,SyDOCC) with MFCC. The combination, denoted as a tuple (,), is done simply by stacking a vector of MFCCs upon the one of SyDOCCs. We use the same clean and noised data like in section 5.1.1. We also use the stacking of adjacent frames. Table 5.2 shows the results of the comparison, and figure 5.2 visualises them. We noticed that in the first and second comparisons, the combination is still a tiny worse than MFCC. However the combination is better with the training with additive street noise for both clean and noisy test. It is the same remarkable situation like the previous section.

training	test	MFCC	(MFCC,SyDOCC)
clean	clean	26.2 %	28.3 %
clean	with street noise	36.3 %	37.5 %
with street noise	clean	27.7 %	27.5 %
with street noise	with street noise	36.4 %	35.1 %

Table 5.2: Comparison of MFCC with (MFCC,SyDOCC) for the language Italian. The percentage values are the WER of the best results

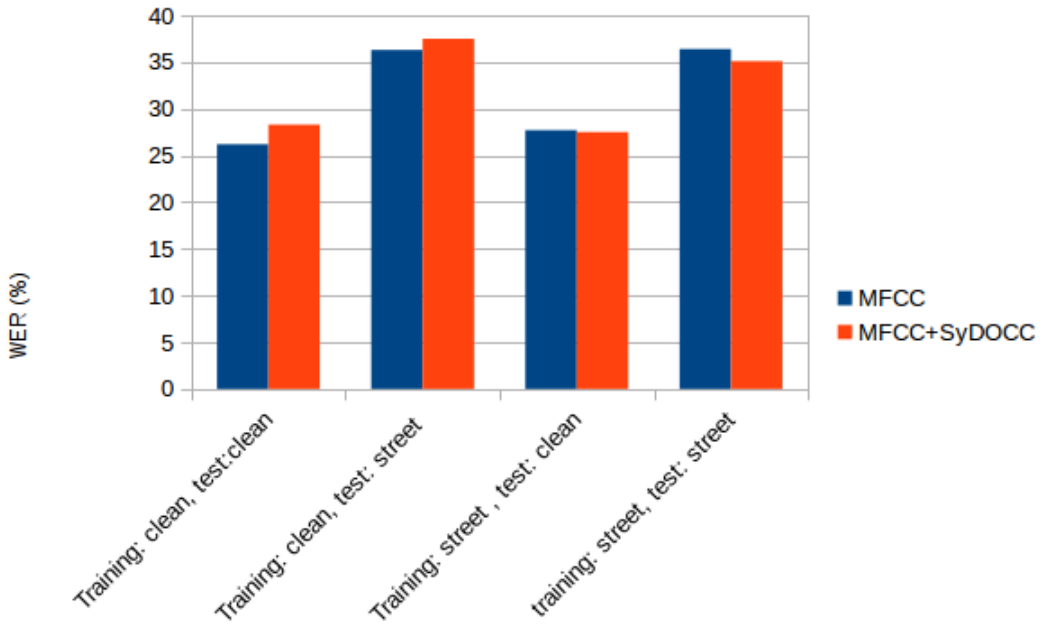

Figure 5.2: Comparison of MFCC with (MFCC,SyDOCC) for the language Italian. The description of the data set is under the columns, where we denote "street" for the data set with an additive street noise, and "white" for the one with additive white noise

5.1.3 Comparison SyDOCC and MFCC with noised test

In this comparison, we will see the ability of SyDOCCs to deal with noise. We tested and trained two systems with a clean training set, one uses SyDOCCs and the other MFCCs. Both use stacked 15 adjacent vectors to capture the dynamic information. The test set is noised with an additive white noise with increasing volume. We see in Table 5.3 the decreasing noise signal ratio (SNR) of the test data signal. Figure

5.3 visualises the results in Table 5.1. We see at the very beginning that MFCCs are better than SyDOCCs but the performance of the system with MFCCs degrades more rapidly and almost quadratic, while the line of the performance of the system with SyDOCCs is flatter. This leads us to the conclusion that SyDOCCs is robuster towards this type of noise.

SNR(test)	MFCC	SyDOCC
24.93	27%	30.10%
18.99	28.9 %	30.50%
15.42	30.9 %	31.80%
12.90	35.1 %	32.60%
11.03	41.4%	34.60%
9.38	48%	36.70%
8.08	55%	39.60%
6.90	62.1%	43.80%
5.87	70.6%	48.60%
4.97	77.1%	52.9%

Table 5.3: Comparison of MFCC with (MFCC,SyDOCC) for the language Italian with decreasing Noise Signal Ratio (SNR) of the test data. The percentage values are the WER of the best results.

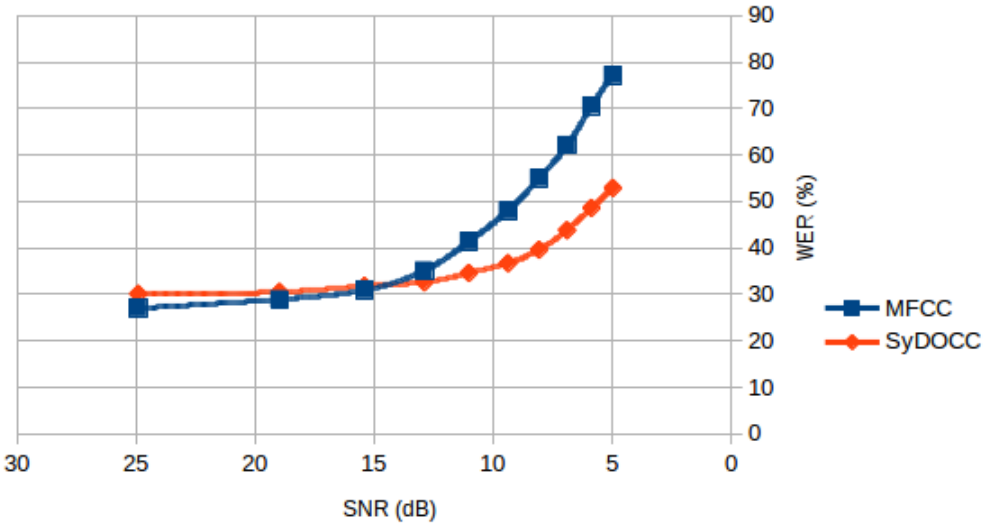

Figure 5.3: Comparison of MFCC with (MFCC,SyDOCC) for the language Italian with decreasing noise signal ratio (SNR) of the test data.

5.2 Experiments with Pashto

In this experiment, we will train a speech recognition system for the language Pashto. In addition to artificially noised data, there is genuine noise like telephone conversations in noisy environments, conversations in car, street and office. This data set is taken from IARPA BABEL program: Pashto FullLP[1], in length of 100 hours of

[1]This effort uses the IARPA Babel Program language collection release IARPA-babel104b-v0.4bY

training data. The test-set is development data with a length 10 hours from 136 speakers. The system we trained is grapheme based system. We use N-grame Language model trained on training utterances because of the lack additional resources in Pashto. The system, like the one which trained for Italian, is bootstrapped using a flatstart approach to initialize the parameters. We built context-independent system and then a context-dependent system using 8000 models because preliminary experiments have shown that this amount of models yields to the best performance. As we see in table 5.4 and its visualisation in figure 5.4, we compare MFCCs with the combination (MFCC,SyDOCC) with different delta features. The results show a difference for about 3% WER in mean for the MFCC.

MFCC	(MFCC, SyDOCC)	(MFCC, SyDOCC d3)	(MFCC d7, SyDOCC d1)
71.5 %	76%	75%	74.2 %

Table 5.4: Comparison of MFCC with (MFCC,SyDOCC) and delta features, denoted with "d", for the language Pashto. Overall, where there is no "d", delta7 is used as default. The percentage values are the WER of the best results

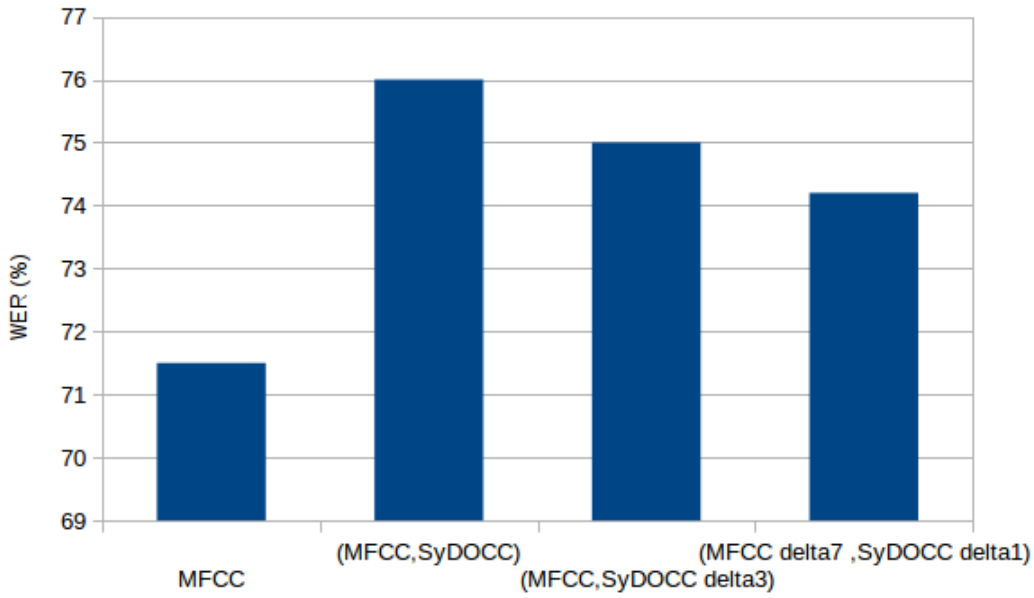

Figure 5.4: Comparison of MFCC with (MFCC,SyDOCC) and delta features, denoted with "d", for the language Pashto. Overall, where there is no "d", delta7 is used as default.

For the sake of error analysis, we show in table 5.5 and its visualization in figure 5.5 three comparisons of results of the previous experiment. We show only results of certain subsets of speakers, out of 136 speakers, on which SyDOCCs combined with MFCCs outperform MFCCs. By analysing these subsets, we could get more information about noises, with which SyDOCCs perform better.

Table 5.6 and figure 5.6 show the relative gain only for a subset of speakers.

	MFCC	SyDOCC combinations	number of speakers
(SyDOCC,MFCC)	71.86%	69.93%	38
(SyDOCC,MFCC delta3)	71.11%	69.47%	23
(SyDOCC delta1,MFCC)	70.97%	69.75%	26

Table 5.5: Comparison of MFCC with (MFCC,SyDOCC) and deltas for the language Pashto. Where no delta mentioned, delta7 is used. The percentage values are the word error rate WER of the best results of a subset of speakers, the first column describes the combination of SyDOCC, their results are in the third column

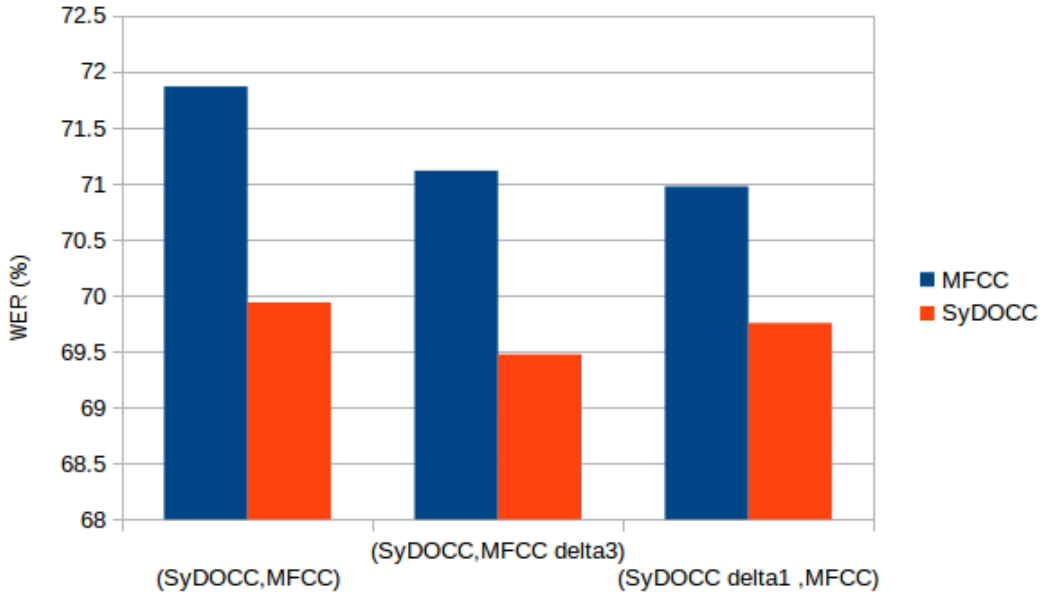

Figure 5.5: Comparison of MFCC with (MFCC,SyDOCC) and deltas for the language Pashto. Where no delta mentioned, delta7 is used. The title under the column are related to the red one, whereas the blue one is for MFCC

SyDOCC,MFCC	(SyDOCC,MFCC delta3)	(SyDOCC delta1,MFCC)
2.76	2.36	1.75

Table 5.6: Comparison of MFCC with (MFCC,SyDOCC) and deltas for the language Pashto. Where no delta mentioned, delta7 is used. the values are the relative gain for some speakers.

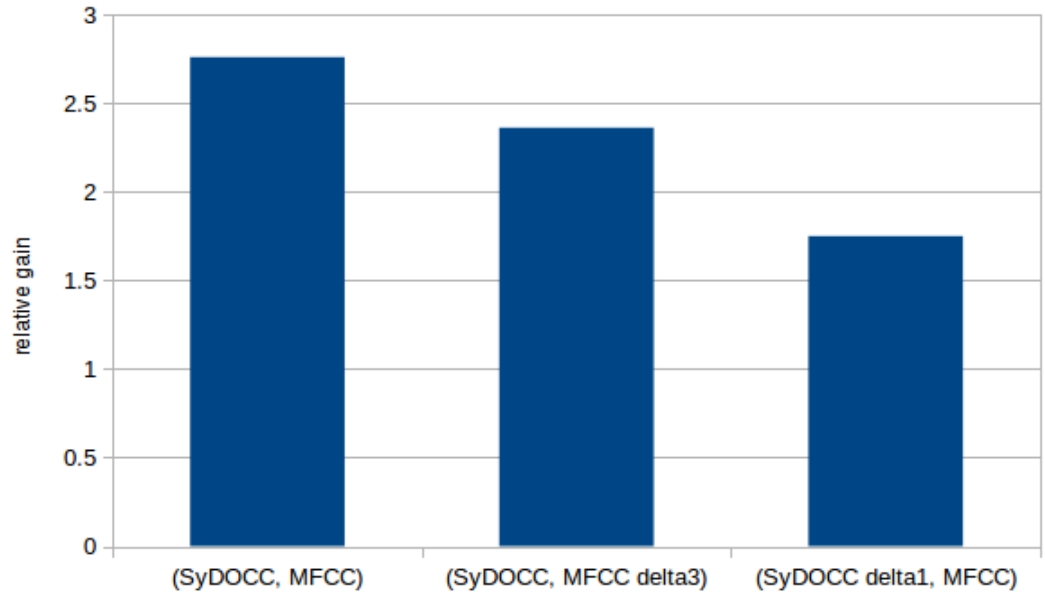

Figure 5.6: Comparison of MFCC with (MFCC,SyDOCC) and deltas for the language Pashto. Where no delta mentioned, delta7 is used. the Y axis indicate to the relative gain for some speakers.

5.3 Experiments with Tagalog

The data[2] and the setting of the system are like those we used with Pashto in the previous section. However, The Test-set are development data with length 10 hours from 146 speakers, and we build context-dependent system using 10000 models.

Table 5.7 and its visualisation in figure 5.7 show the best result among all speakers. The system with MFCCs show 2.4% less WER, For Analysing errors we did the same as with Pashto. As we see in Table 5.8 and figure 5.8, the best result of a subset of speakers (43 of 146 total speaker) on which the SyDOCCs combined with MFCCs with delta 3 outperformed the MFCCs.

MFCC	(MFCC,SyDOCC delta3)
75.5 %	77.9%

Table 5.7: Comparison of MFCC with (MFCC, SyDOCC delta3) for the language Tagalog. Where no delta mentioned, delta7 is used. The percent values are the WER of the best results

MFCC	(SyDOCC,MFCC delta3)	number of speakers
83.53%	76.55%	43

Table 5.8: Comparison of MFCC with (MFCC,SyDOCC delta3) with deltas. Where no delta mentioned, delta7 is used. The percent values are the WER of the best results of a subset of speakers

5.4 Summary

We have investigated the tuning of several free paramters of SyDOCCs; however, there are still many parameter combinations to be examined in order to reach better

[2]This effort uses the IARPA Babel Program language collection release IARPA-babel106-v0.2f

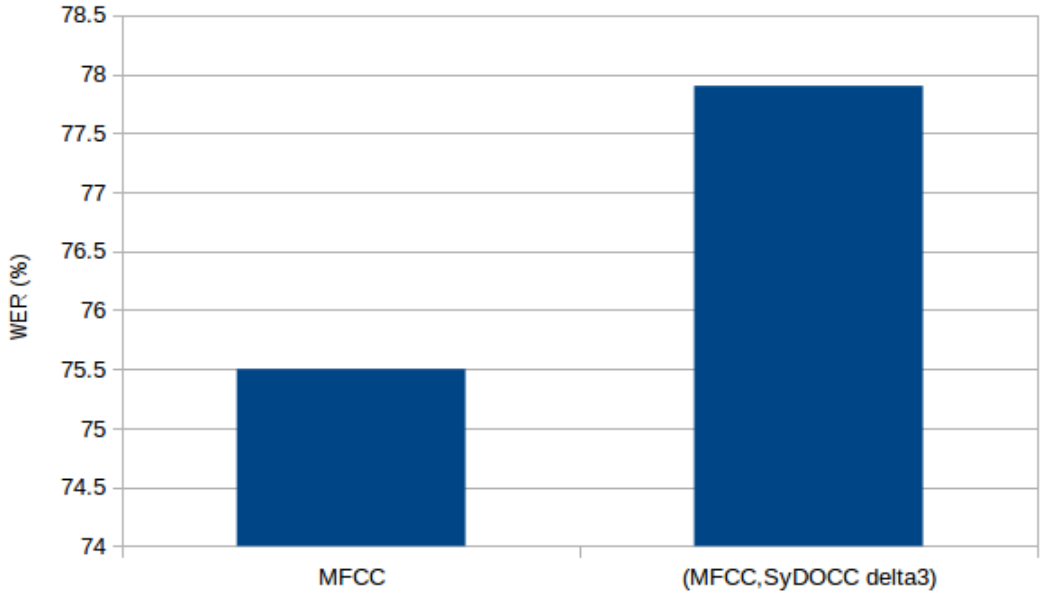

Figure 5.7: Comparison of MFCC with (MFCC, SyDOCC delta3) for the language Tagalog. Where no delta mentioned, delta7 is used.

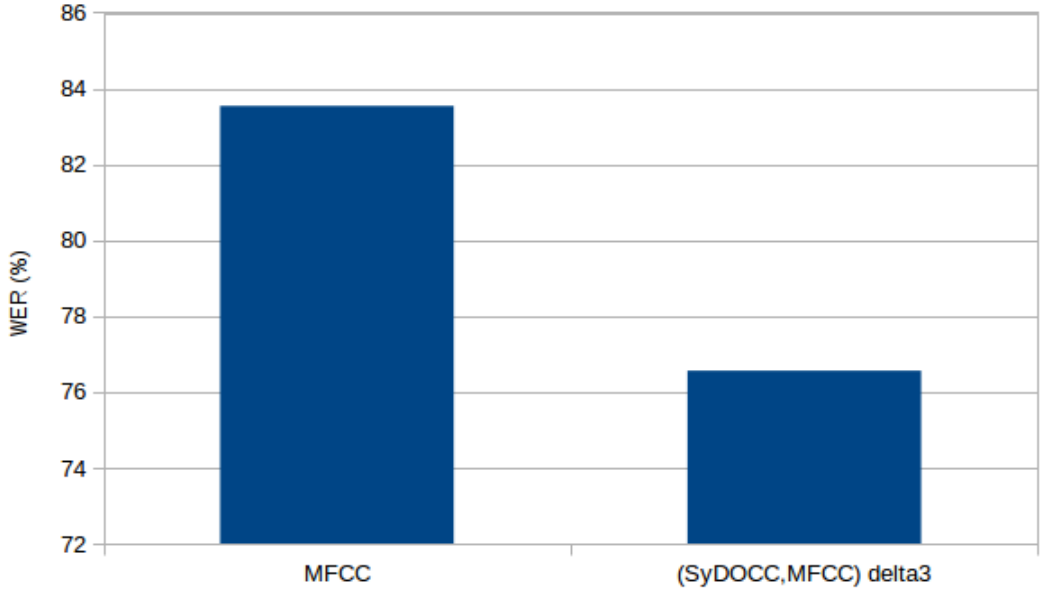

Figure 5.8: Comparison of MFCC with (MFCC,SyDOCC delta3) deltas. Where no delta mentioned, delta7 is used.

results with SyDOCCs. We found that SyDOCCs are robust towered white noise. We found also that SyDOCCs outperform MFCCs when using noisy training, where we use once white noise and once an additive street noise for the training and tested with clean and a street noised data.

6. Summary and outlook

In this thesis, we have implemented the Synchronised Damped Oscillator Cepstral Coefficients (SyDOCCs) as a new feature for robust speech recognition. These features are motivated by the hearing mechanism of the human ear, to take advantage of its insensitivity towards some kind of noises. We have simulated the function of basilar membrane with the gamma-tone filterbank, and the function of hair cells with the damped oscillators. There is still some research to be done to simulate more features like amplifying sound by the hair cells, otoacoustic emission features of outer hair cell, or synchrony of the hair cell stereocilia.
We have seen that SyDOCCs are robuster towards white noise compared to MFCC features. In a clean environment, SyDOCCs bring not much advantage and MFCCs are a tiny bit better. Many experiments have shown that SyDOCCs could improve the performance if combined with MFCCs. We have evaluated SyDOCCs with artificial noise and partly with a real word noise. Test with additional languages from Option Period 1 of the BABEL program, which have greater acoustic mismatch between speakers, can give us more informative statement about SyDOCCs. More experiments and more parameter tuning for SyDOCCs can still give an improvement on the recognition task. It would also be beneficial, to implement SyDOCCs directly as a part of the Janus Recognition Toolkit.

Literatur

[FeFu99] R. Fettiplace und P. Fuchs. Mechanisms of hair cell tuning. *Annual review of physiology* 61(1), 1999, S. 809–834.

[GlMo90] B. R. Glasberg und B. C. Moore. Derivation of auditory filter shapes from notched-noise data. *Hearing research* 47(1), 1990, S. 103–138.

[HuAH01] X. Huang, A. Acero und H.-W. Hon. *Spoken language processing : a guide to theory, algorithm, and system development*. Prentice Hall, Upper Saddle River, NJ. Includes bibliographical references and index; Geb. : DM 182.95, 2001.

[IrPa97] T. Irino und R. D. Patterson. A time-domain, level-dependent auditory filter: The gammachirp. *The Journal of the Acoustical Society of America* 101(1), 1997, S. 412–419.

[LuDa08] X. Lu und J. Dang. An investigation of dependencies between frequency components and speaker characteristics for text-independent speaker identification. *Speech communication* 50(4), 2008, S. 312–322.

[Ma] N. Ma. An Efficient Implementation of Gammatone Filters. http://staffwww.dcs.shef.ac.uk/people/N.Ma/resources/gammatone/. [Online; accessed 16-December-2014].

[MiFG13] V. Mitra, H. Franco und M. Graciarena. Damped oscillator cepstral coefficients for robust speech recognition. In *INTERSPEECH*, 2013, S. 886–890.

[MuBE10] L. Muda, M. Begam und I. Elamvazuthi. Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. *arXiv preprint arXiv:1003.4083*, 2010.

[PNSHR87] R. Patterson, I. Nimmo-Smith, J. Holdsworth und P. Rice. An efficient auditory filterbank based on the gammatone function. In *a meeting of the IOC Speech Group on Auditory Modelling at RSRE*, Band 2, 1987.

[RaAS] S. Ravindran, D. V. Anderson und M. Slaney. IMPROVING THE NOISE-ROBUSTNESS OF MEL-FREQUENCY CEPSTRAL COEFFICIENTS FOR SPEECH DISCRIMINATION.

[Schu95] E. G. Schukat-Talamazzini. *Automatische Spracherkennung : Grundlagen, statistische Modelle und effiziente Algorithmen*. Künstliche Intelligenz. Vieweg, Braunschweig [u.a.]. kart. : DM 98.00, 1995.

[ScNK11] J. Schnupp, I. Nelken und A. King. Auditory neuroscience : making sense of sound, c2011. Includes bibliographical references and index. - Description based on print version record.

[ShPa03] B. J. Shannon und K. K. Paliwal. A comparative study of filter bank spacing for speech recognition. In *Microelectronic engineering research conference*, Band 41, 2003.

[Smit99] J. Smith. Bark and ERB bilinear transforms. *IEEE Transactions on Speech and Audio Processing, 7(6):697, 1999* 7(6), November 1999, S. 697 – 708.