# A Wavelet and Filter Bank Framework for Phonetic Classification

by

Ghinwa F. Choueiter

B.E., American University of Beirut, Lebanon (2002)

Submitted to the Department of Civil and Environmental Engineering
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August 2004



Author .....
Department of Civil and Environmental Engineering
July 30, 2004

Certified by .....
James R. Glass
Principal Research Scientist
Thesis Supervisor

Certified by .....
Ruaidhri M. O'Connor
Assistant Professor
Thesis Reader

Accepted by .....
Heidi Nepf
Chairman, Department Committee on Graduate Students

# A Wavelet and Filter Bank Framework for Phonetic Classification

by

Ghinwa F. Choueiter

Submitted to the Department of Civil and Environmental Engineering
on July 30, 2004, in partial fulfillment of the
requirements for the degree of
Master of Science

## Abstract

In this thesis, we construct a wavelet and filter bank framework for context-independent phonetic classification, with the aim of extending it towards a full speech recognition system. The framework is implemented for feature extraction, and targets the limitations of the commonly used STFT analysis. The wavelet transform allows a multiresolution analysis and subsequently a good signal representation by exploiting the time/frequency resolution trade-off. The wavelet transform can also be efficiently implemented using an adequately designed filter bank.

Most of the work reported in the literature on wavelet-based analysis for speech recognition involves off-the-shelf wavelets and dyadic filter bank implementation. The original contribution of this work lies in extending previous work in two directions: filter design and the implementation of *rational* filter banks. We adopt two filter design techniques. The first minimizes the modulus between the designed and desired filter imposing orthogonality through the lattice structure of the filter. The second method minimizes the attenuation in the filter stopband. We first use tree-structured filter banks to obtain various frequency partitions. We then adopt a method for the design of rational filter banks. The latter naturally incorporates the critical band effect.

A baseline classifier using MFCC acoustic measurement has a typical error rate of 24.6% on the TIMIT Core Test set. We match and exceed this result, as well as those reported in the literature. For example, using a rational filter bank implementation we obtain 24.0% on the Core Test Set. We also get 22.9% for the same acoustic measurement using 4-fold aggregation.

Thesis Supervisor: James R. Glass
Title: Principal Research Scientist

Thesis Reader: Ruaidhri M. O'Connor
Title: Assistant Professor

# Acknowledgments

I would first like to acknowledge my thesis advisor, Jim Glass, for his patience and guidance throughout the past year as my thesis work progressed. Equally, I thank TJ Hazen for answering my many questions on almost anything, my previous office-mates Karen Livescu and Min Tan for the fruitful conversations on various topics, as well as Han Shu for answering my questions when TJ was not there.

I am thankful to Rory O'Connor, Kate Saenko, and Ken Schutte for reading my thesis drafts and providing me with insightful feedback.

My deepest gratitude goes to Mesrob, Rory, and Rayka, my three angels who protect me from the dangers of the world — mostly myself —, and give me all the love and support.

Many thanks go to the congregation of the Jesuit Urban Center in Boston and its very special choir members. I thank, especially, Father Tom Carroll, Peter Wick, Kira Hanson, Dong-ill Shin, and Mark Brown for their unconditional love.

I would not have been who I am and where I am without the love and sacrifices of my father, mother, and sisters, Fakhry, Therese, Nadine, and Mary Choueiter. I thank my father for instilling in me the desire to be the best I can, and my mother, the eternal perfectionist, for being the best role model I can have. I thank Nadine for her never-ending patience, and for being there for me all the time. I thank Mary for her impatience, and for teaching me to stand up for myself, and fight on my own.

I thank God for the many blessings in my life.

This research was supported by the MIT/Microsoft iCampus Alliance for Educational Technology and the SLS Affiliate program.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

The theory of wavelets has been studied intensively over the past two decades. The range of their application varies from image and signal processing to geophysics. Wavelets are functions with compact support capable of representing signals with a good resolution in the time and frequency domains. The wavelet transform is well defined within the multiresolution framework which allows signal analysis at various scales and thus enables us *to see the forest and the trees*. Wavelets, like sinusoids, are basis functions that span the square-integrable space, $L_2(\mathcal{R})$, and can be used to develop series expansions of signals belonging to that space.

These attractive features led wavelet analysis to be studied as a mathematical tool for signal processing and proposed as an alternative to Fourier analysis which, in its simplest form as the *Fourier Transform* (FT) lacked any time localization. The *Short-Time Fourier Transform* (STFT) was later proposed to overcome the limitations of the FT. However, the fixed window-size in the STFT implied time-stationarity of the signal within each time frame and provided only a fixed time and frequency resolution. The *Wavelet Transform* (WT) was able to overcome this problem. Figure 1-1 illustrates the resulting time-frequency space for the Fourier and Wavelet analysis respectively. As sinusoids are basis functions in Fourier analysis, wavelets form the basis functions in Wavelet analysis. A sinusoidal function as well as a Daubechies wavelet of order 10 are illustrated in Figure 1-2. The significant advantage is that wavelets are localized in time while sinusoids are not.

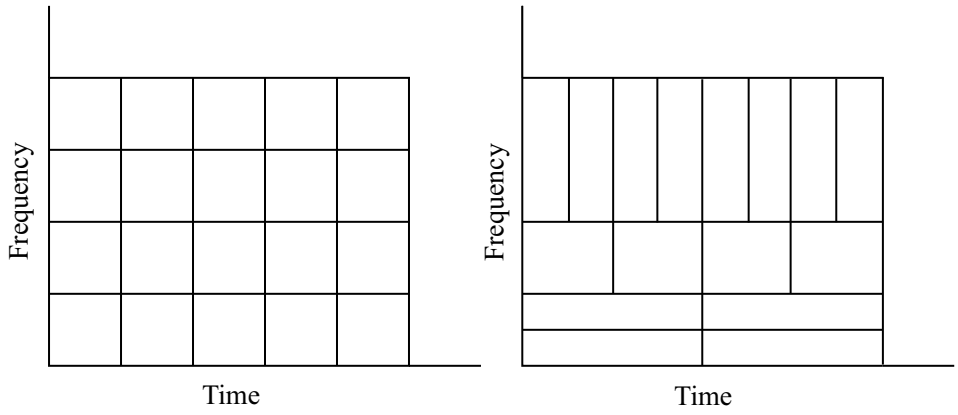


Figure 1-1: Tiling of the time-frequency space in Fourier Analysis (left) versus Wavelet Analysis (right).

Filter banks have also emerged as signal processing tools that decompose and analyze signals. A filter bank is an array of filters, which can be low-pass, band-pass, or high-pass, that are used to decompose a signal into subbands over different regions of the frequency spectrum. Such an analysis is quite useful, especially when the signal has a non-uniform spectral content as in the case of speech.

Wavelets and filter banks have evolved separately, where wavelets thrived in applications for theoretical and applied mathematics, such as the design of orthogonal bases for the $L_2(\mathcal{R})$ space[12], while filter banks have been applied successfully in subband coding for speech compression [11]. Within the multiresolution framework, continuous-time wavelets are closely connected to discrete-time filter banks where it has been proved that a wavelet transform can be implemented using filter banks [39, 42]. It is this relation that we study and exploit in the problem of phonetic classification. Given some desired filter characteristics, we design a filter and implement it in a filter bank to obtain the wavelet transform. The filter bank features that we wish to leverage are perfect reconstruction, regularity, and orthogonality. Figure 1-3 illustrates this scheme.

In Automatic Speech Recognition (ASR) systems, an acoustic-phonetic model maps the speech waveform to a string of discrete phonetic units. Acoustic-phonetic modelling requires first capturing acoustic observations necessary for phonetic distinction. This problem has always been a challenging research topic requiring efficient

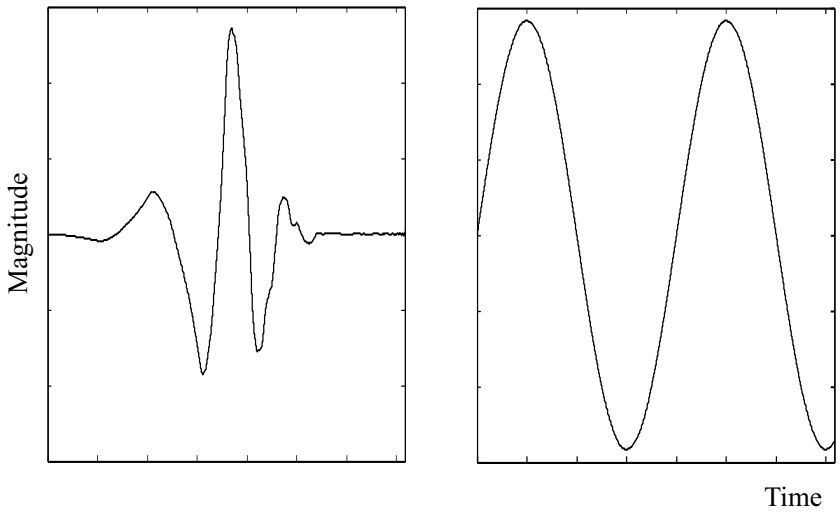


Figure 1-2: A wavelet function (left) versus a sinusoidal function (right).

extraction of acoustic features that compactly represent the speech waveform and yet preserve discriminatory information. Some of the commonly used acoustic measurements are: *perceptual linear prediction cepstral coefficients* (PLPCC) [26], *discrete cosine transform coefficients* (DCTC), *linear predictive coefficients* (LPC) [37], and the dominantly used *Mel-Frequency Cepstral Coefficients* (MFCC) [13]. MFCCs will be extensively referenced in this work and used as the reference point for comparative studies with the new acoustic measurements developed in this thesis.

As illustrated in Figure 1-1, the flexible time-frequency representation that the wavelet transform provides makes it a suitable tool for extracting speech features. This feature has been used for the past decade for various applications in signal processing such as denoising [22], compression [3], as well as speech recognition [14, 15, 16, 24]. Furthermore, within the realms of speech analysis and recognition, wavelets have had diverse applications to problems such as pitch detection [29] and formant tracking [20].

The goal of this thesis is to propose a new framework for wavelets and filter banks for feature extraction in speech recognition systems. Specifically the work pertains to context-independent phonetic classification. While a wavelet-based framework for context-independent phonetic classification is a long way from automatic speech recognition, we believe this examination will give us insight into the advantages and

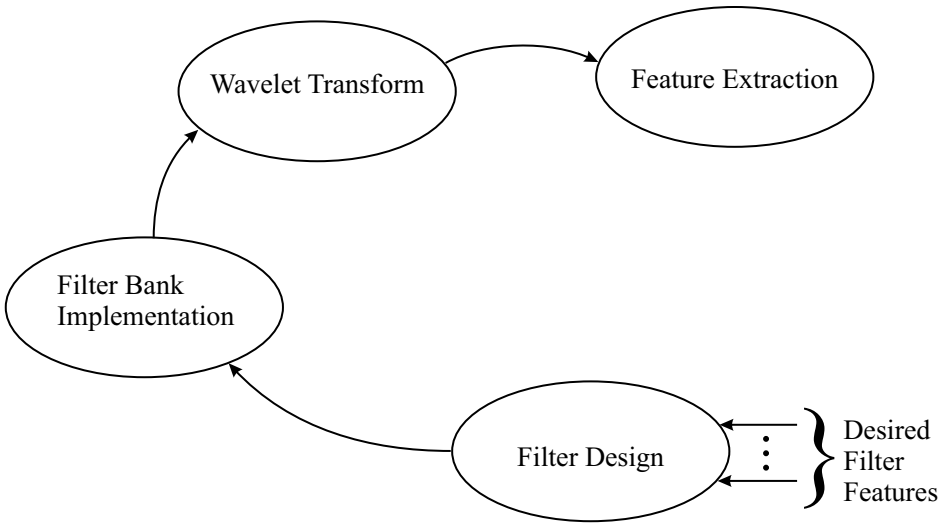


Figure 1-3: A flowchart illustrating a scheme initiated with filter design and terminated with a wavelet-based analysis for the task of feature extraction.

limitations of the proposed techniques. Through our work in deriving a problem-specific wavelet and filter bank architecture, we have successfully achieved results that match and exceed the performance of MFCCs. It has been shown that such gains in phonetic classification translate into gains in phonetic and word recognition [23, 34].

## 1.1 Previous Work

### Acoustic Observations

In the acoustic modeling literature, numerous acoustic representations have been proposed and evaluated [2, 13, 23, 26, 28, 37]. One of the earliest comparative studies on parametric representation is done by Davis and Mermelstein [13]. Several short-time spectral representations are presented and compared. Two groups of observations are studied: the first is based on Fourier analysis such as MFCC and *Linear Frequency Cepstrum Coefficients* (LFCC), and the second is based on linear prediction analysis [37] such as LPC, *Reflection coefficients* (RC) and *Linear Prediction Cepstral Coefficients* (LPCC). A significant result of this work is that the MFCC outperforms all the other acoustic representations. This can be attributed to its good capture

of perceptual information. This work was the first to set the MFCC as the most commonly used representation in speech recognition systems. It is implemented in many current state-of-the-art recognizers [18]. Appendix A includes a description of the MFCC computation scheme, and for further information, we refer the reader to [13].

In a more recent PhD Thesis, Halberstadt examined several heterogeneous acoustic measurements such as MFCC, DCTC, and PLPCC [23]. All three are, again, short-time spectral representations that aim to incorporate perceptual information. For example, the PLPCC computation, which is based on an all-pole model of the vocal tract, takes into account the critical-band spectral resolution, the equal-loudness effect, and the intensity-loudness power law. It has been conjectured that PLPCC and MFCC carry complementary information [23].

## Wavelet-based Acoustic Observations

In an attempt to investigate acoustic measurements that could outperform the MFCC or provide a more efficient representation of the speech signal, much research has been done in the field of wavelet theory [10, 15, 24, 30, 38, 40, 41, 43]. We elaborate on selected examples.

Wassner and Chollet replaced the Fourier analysis stage, required for the computation of the power spectrum, with the wavelet transform [43]. Based on empirical results, the authors also shifted the log transformation stage and proposed different energy-transformations. Their MFCC variant is evaluated on an HMM-based speaker-independent connected word recognition using three sub-databases extracted from Polyphone and Computer95[1]. The results showed improvement over the original MFCC for both databases.

An important component of the aforementioned paper is an emphasis on the need to search for a signal analysis tool different from the Fourier transform. This poses the question of whether Fourier analysis provides an optimal representation of the

[1]The two databases were collected by IDIAP and the Swiss telecom PTT from French spoken telephone speech.

signal. Given its lack of good localization in the time-frequency space and lack of flexibility in basis function design, we can say that it does not. However, this is not a new fact. As we have seen in this section, much work within the scope of wavelet analysis has been done, more than a decade ago, to overcome this limitation.

Tan et al. suggested the design of a hearing aid based on the wavelet transform [40]. Their focus was on the classification of speech into four major classes - voiced, plosives, fricatives, and silence. They proposed the use of the wavelet transform instead of the STFT to segment and classify speech signals. They used the Daubechies wavelet of order 10 [12] to decompose the signal up to 4 scales. The output of the wavelet transform is windowed using a 4ms Hamming window every 0.8ms and the RMS energy is computed for each frame. Maximum Likelihood (ML) is used for speech/non-speech segmentation. The output at each scale of the wavelet transform is used in order to classify the speech segment as voiced, plosive, or fricative.

## 1.2 Goal and Motivation

**The goal of this thesis is to present a wavelet and filter bank framework for phonetic classification with the motivation of extending the work towards a full speech recognition system implementation. Limitations of previously designed wavelet-based acoustic measurements are discussed, and a new approach, proposed as an extension to the previous work, is implemented, and evaluated.**

In particular configurations, such as the simple dyadic case, the wavelet transform corresponds to a filter bank with a logarithmic spectrum or constant relative bandwidth. This is contrasted with the linear frequency scale of the STFT. We compare with the STFT since, as mentioned earlier, the most common acoustic measurements are short-time spectral representations with an initial Fourier analysis stage. The correspondence between the discrete wavelet series and the constant-$Q$ filter bank will become more clear in Chapter 2. It is this connection that renders wavelet analysis

a natural option in audio applications since the filter bank can simulate the hearing process by taking into account psychoacoustic effects such as the critical band — filters are spaced linearly at low frequency and logarithmically at high frequency.

From a signal processing point of view, Fourier analysis is limited to a uniform time-frequency resolution while the wavelet transform has a more flexible time-scale representation. It is this flexibility that has made wavelets an appealing mathematical tool for signal analysis. The ability to switch between good frequency resolution at the low-frequency bands and good time resolution at the high-frequency bands allows us to capture localized changes in the signal as well as coarse approximations of the signal.

The wavelet transform represents a signal at several scales making it easier to separate the noisy component from the original signal. Again this contrasts with the MFCC, which has the disadvantage of smearing a noisy band-limited component across all cepstral coefficients. Hence, although MFCC is a fairly robust representation, it is not very reliable in adverse conditions such as noisy environments. We do not focus too much on this point since our data set is clean as will be seen in Section 5.1.1. It is, however, an important fact since it implies the possibility of designing a robust framework in noisy conditions.

As we have seen, wavelets are a compelling candidate for speech processing. In the hope of providing further reinforcement for the motivation of this work, we look at other examples of wavelet application in speech recognition.

Farooq and Datta used wavelet packets to obtain a 24-band filter bank that mimics the MFCC [14]. The acoustic feature is obtained by computing the energy in each frequency band, converting to a log scale, and performing a *discrete cosine transform* (DCT) ending up with 13 coefficients. The Daubechies wavelet with 6 vanishing moments is used. Phonetic classification is performed on the TIMIT corpus — over a limited subset of the data and the phonemes — and compared with the MFCC. Their results showed that the wavelet-based feature outperforms the MFCC in the case of stops and unvoiced phonemes.

Kim et al. proposed a modified octave structured 5-level filter bank for speech

recognition [31]. They experimented on Korean digit words, and various orthogonal and biorthogonal Daubechies wavelets are tested. The biorthogonal filter 9-7 tab filter (80.07%) outperformed both the best orthogonal filter (79.38%), the LPC-based feature (75.38%), and the MFCC-based feature (77.23%). The results are averaged over 13 test speakers.

Tan et al. studied and compared *discrete wavelet transform* (DWT) with *sampled continuous wavelet transform* (SCWT) and MFCC as front-end processors in a speaker-independent HMM-based phoneme recognition system [41]. The acoustic feature extracted with the SCWT used a Morlet wavelet [21] with a constant $Q$-factor of approximately 3.3. The output of the SCWT is half-wave rectified, low-pass filtered, and downsampled from 16 kHz to 100 Hz. The result is then reduced to 12 coefficients using cepstral analysis. The acoustic feature extracted with the DWT used Daubechies wavelet with 8 vanishing moments. The signal is analysed up to 6 scales and the two largest coefficients in each scale are retained giving a 12-dimensional observation vector. The experiments are performed over a subset of the TIMIT corpus indicating a better recognition rate for the SCWT over the DWT but marginal improvement over the MFCC.

Most of the work pertaining to wavelet analysis for speech recognition, suffer from two main drawbacks which we will attempt to tackle in this thesis.

- The acoustic measurements are restricted to off-the-shelf wavelets such as Daubechies, Coiflets, and Symlets wavelets depicted in Figure 1-4 with their corresponding low-pass filters. While these wavelets might have appealing features such as smoothness in the case of the Daubechies and near symmetry in the case of the Coiflet and the Symlet, they are not optimized for speech processing tasks.
  **One possible direction, that we adopt in this thesis, is to design a filter and hence a wavelet that matches desired features.**

- The computation of the wavelet transform is typically restricted to the dyadic case meaning that at each iteration of the filter bank, the spectrum is split in half. This does not give us a fine resolution of the spectrum especially at the high

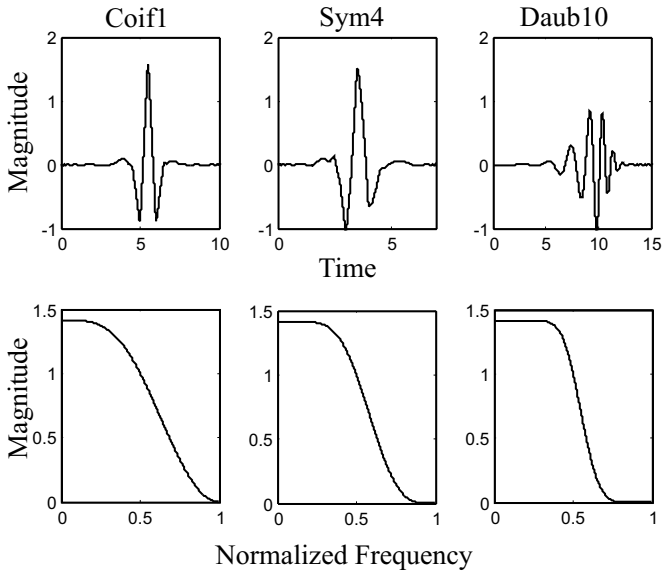


Figure 1-4: An illustration of three wavelets: the Coiflet of order 1, the Symlet of order 4, and the Daubechies of order 10 along with the corresponding low-pass filters.

frequencies. *Wavelet Packets* (WP) are a valid solution to solve the problem of frequency resolution by allowing iteration at both the high and low channels of the 2-channel filter bank. However, with WP, the constant-$Q$[2] characteristic of the filter bank is lost, and so is the ability to naturally take into account the variations of the ear's critical bandwidths with frequency.

**In this thesis, we propose to use rational sampling to obtain a finer resolution of the frequency axis and naturally simulate the critical bandwidths. The result is the ability to design a filter bank that matches desired features in the more general rational case.**

[2]The $Q$-factor of a filter is defined as the ratio of the bandwidth to the center frequency. A filter bank is constant-$Q$ when all its filter have the same $Q$ factor.

## 1.3 Thesis Structure

In this thesis we present and discuss a wavelet analysis framework for phonetic classification on the TIMIT corpus. The chapters are divided accordingly:

- **Chapter 2: Theoretical Background**

  We present the theoretical basis on which the wavelet and filter bank framework is built; we identify the basics of wavelets and filter banks required to understand the content of the thesis.

- **Chapter 3: Problem Specification**

  We define the problem of phonetic classification and the hypothesis proposing a wavelet and filter bank framework, and discussing the advantages that it provides in terms of flexibility in filter design and frequency band decomposition.

- **Chapter 4: Implementation**

  We describe an implementation of the wavelet and filter bank framework: the acoustic measurement, the initial implementation based on off-the-shelf wavelets and dyadic sampling filter banks, the filter designs, and the final implementation with 'pseudo-wavelets' and rational sampling.

- **Chapter 5: Evaluation**

  We define the experimental setup, the TIMIT corpus, the classifier, the baseline classifier, then present and evaluate our results.

- **Chapter 6: Conclusion**

  We summarize the work we have presented and propose possible extensions and improvements.

- **Appendix A:**

  We describe the MFCC computation algorithm.

- **Appendix B:**

  We describe the frequency partitions implemented and tested in the thesis.

# Chapter 2

# Theoretical Background

In this chapter, we present an overview of some of the basic concepts of wavelets and filter banks with the aim of showing the close connection between the two fields and setting up the background for filter design. We then move on to more advanced topics such as rational sampling filter banks. The chapter includes all the building blocks required to implement the wavelet and filter bank framework as well as the material needed to understand the content of the thesis. Most of the material in this chapter is based on the books by Vetterli [42] and Strang [39], where we refer the reader for detailed explanations and proofs.

## 2.1 Notations and Brief Concepts

### 2.1.1 Notations

A boldfaced uppercase/lowercase character denotes a matrix/vector. For example, $\mathbf{M}$ is a matrix and $\mathbf{v}$ is a vector. $\mathbf{I}$ denotes the identity matrix. The superscripts $^T$, and $^\dagger$ denote matrix transpose and matrix conjugate transpose respectively. The determinant of a matrix $\mathbf{M}$ is denoted $\det(\mathbf{M})$. The $N^{th}$ root of unity is denoted $W_N = \exp^{\frac{j2\pi}{N}}$. $\delta[n]$ denotes the Dirac function.

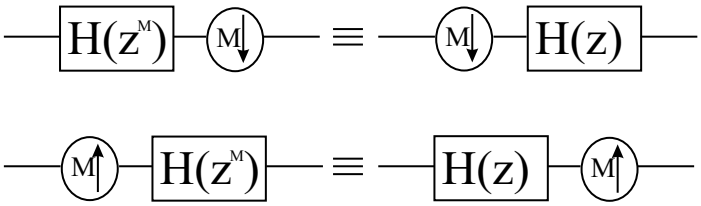


Figure 2-1: The Noble Identities.

### 2.1.2 Brief Concepts

#### Noble Identities

Figure 2-1 illustrates the Noble Identities. The top diagram shows when it is possible to reverse the order between filtering and a downsampler by a factor of M. The bottom diagram illustrates the same for filtering and an upsampler by a factor of M. Such properties will prove useful when optimizing filter bank implementation.

#### Polyphase Representation

A filter $H(z)$ has a unique polyphase representation, which for $M$ phases is:

$$H(z) = \sum_{i=0}^{M-1} H_i(z^M) z^i \quad \text{where } H_i(z) = \sum_{n=-\infty}^{\infty} h[Mn - i] z^{-n} \tag{2.1}$$

#### Unitary and Paraunitary Matrix

A matrix $\mathbf{M}$ is unitary if:

$$\mathbf{M}^\dagger \mathbf{M} = c\mathbf{I}, \qquad c \neq 0 \tag{2.2}$$

A matrix $\mathbf{H}(z)$ with real-valued coefficients is paraunitary if:

$$\mathbf{H}^T(z^{-1})\mathbf{H}(z) = c\mathbf{I}, \qquad c \neq 0 \tag{2.3}$$

## 2.2 Filter Banks

Filter banks can be efficiently implemented using discrete finite impulse response (FIR) filters, downsamplers, and upsamplers. Perfect reconstruction of the filter

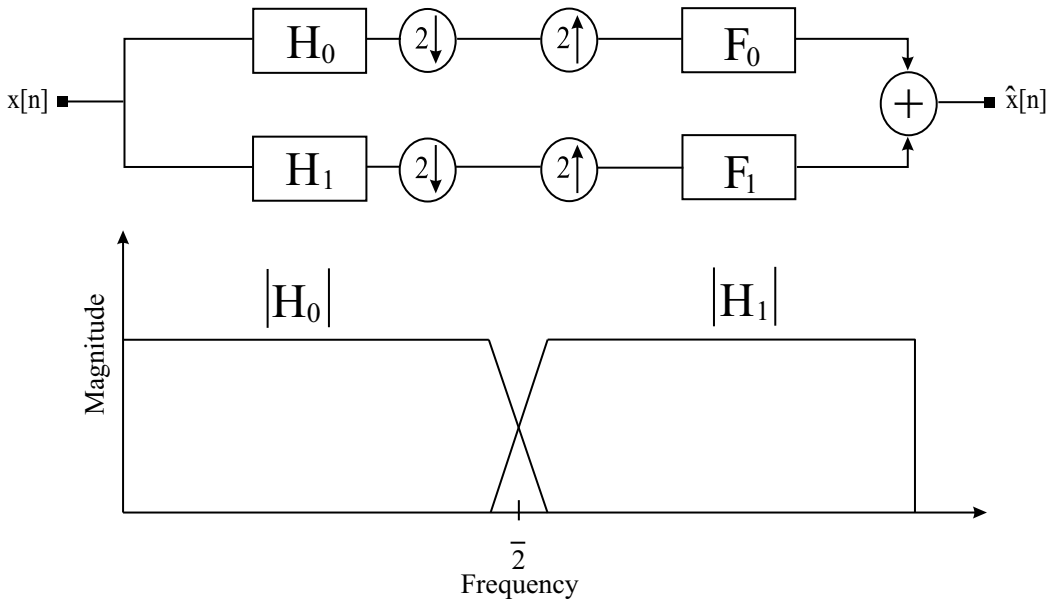


Figure 2-2: A 2-channel filter bank and the corresponding frequency spectrum partitioning.

bank is a requirement meaning that the input signal processed by analysis filters at one end of the channel should be perfectly reconstructed by some synthesis filters at the other end. Furthermore,

*Perfect reconstruction filter banks can be used to implement series expansions of discrete-time signals in the* $l_2(\mathcal{Z})$ space

Figure 2-2 shows such an implementation along with the corresponding frequency bands for the 2-channel case. In this case, $H_0(z)$ is the analysis low-pass, $H_1(z)$ is the analysis high-pass, $F_0(z)$ is the synthesis low-pass, and $F_1(z)$ is the synthesis high-pass.

### 2.2.1 The Modulation Domain

For simplicity, we study the 2-channel case for the time being. The output signal, $\hat{X}(z)$, of the filter bank is:

$$\underbrace{\frac{1}{2}[F_0(z)H_0(z) + F_1(z)H_1(z)]X(z)}_{\text{amplitude distortion}} + \underbrace{\frac{1}{2}[F_0(z)H_0(-z) + F_1(z)H_1(-z)]X(-z)}_{\text{aliasing component}} \quad (2.4)$$

For perfect reconstruction, we would like the output to be a delayed version of the input:

$$\hat{X}(z) = z^{-L} X(z) \tag{2.5}$$

In matrix form, this implies:

$$\begin{bmatrix} F_0(z) & F_1(z) \end{bmatrix} \begin{bmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{bmatrix} = \begin{bmatrix} 2z^{-L} & 0 \end{bmatrix} \tag{2.6}$$

The **analysis modulation matrix** is denoted:

$$\mathbf{H_m}(z) = \begin{bmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{bmatrix} \tag{2.7}$$

The **synthesis modulation matrix** is denoted:

$$\mathbf{F_m}(z) = \begin{bmatrix} F_0(z) & F_1(z) \\ F_0(-z) & F_1(-z) \end{bmatrix} \tag{2.8}$$

In the modulation domain, the condition for a perfect reconstruction filter bank (PRFB) (see [42] for proof) is:

$$\mathbf{H_m}(z)\mathbf{F_m}(z) = 2\mathbf{I} \tag{2.9}$$

Such a filter bank is referred to as biorthogonal. A perfect reconstruction filter bank is orthonormal if it satisfies (see [42] for proof):

$$\mathbf{H_m}(z)\mathbf{H_m}^T(z^{-1}) = 2\mathbf{I} \tag{2.10}$$

When a filter bank is orthonormal, the analysis filter and its modulated version are power complementary, meaning that:

$$|H_i(e^{j\omega})|^2 + |H_i(e^{j(\omega+\pi)})|^2 = 2, \quad i = 0, 1 \tag{2.11}$$

A similar relation holds for the synthesis filter. It is appealing to work with an orthonormal filter bank since:

*an orthonormal filter bank can implement an orthonormal expansion of discrete-time signals in the $l_2(\mathcal{Z})$ space.*

In this thesis, we will be working only with orthonormal filter banks.

One problem with the modulation implementation is its inefficiency. To understand this, we refer again to Figure 2-2. After the input signal is convolved with the low-pass and the high-pass — filtering in the frequency domain is equivalent to convolution in the time domain — the result is downsampled by 2. Half of the computed values are thrown away. On the other hand, at the synthesis side, the signals are upsampled by 2 prior to convolution and twice the amount of computations is needed although half of the values are set to 0 by the upsampler. To solve this problem, we turn to the polyphase domain.

### 2.2.2 The Polyphase Domain

The $M$-component polyphase representation of a filter is given by Equation 2.1. It shows how a system can be represented in terms of its phases. In the case of a filter bank implementation, the analysis filters, for example, are decomposed into their phases, which operate simultaneously on the phases of the input. The same can be said of the synthesis filters. For example, in the 2-channel case, the input $X(z)$ and analysis filters $H_i(z),\ i = 0, 1$, are represented in terms of their two phases. The outputs $Y_i(z),\ i = 0, 1$, of this 2-input/2-output system are given by:

$$\begin{pmatrix} Y_0(z) \\ Y_1(z) \end{pmatrix} = \begin{pmatrix} H_{00}(z) & H_{01}(z) \\ H_{10}(z) & H_{11}(z) \end{pmatrix} \begin{pmatrix} X_0(z) \\ X_1(z) \end{pmatrix} \tag{2.12}$$

where

$$H_i(z) = H_{i0}(z^2) + zH_{i1}(z^2) \tag{2.13}$$

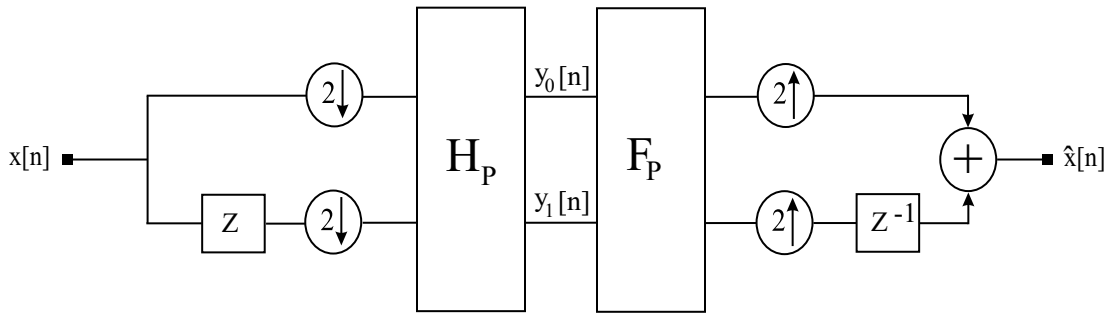


Figure 2-3: A polyphase implementation of the 2-channel filter bank.

The key point in using a polyphase implementation is that we can incorporate the Noble Identities from Section 2.1 to shift the downsamplers and upsamplers before and after the filters respectively. Figure 2-3 shows a polyphase implementation of the filter bank in Figure 2-2. Notice the locations of the upsamplers and downsamplers. The **analysis polyphase matrix** is denoted:

$$\mathbf{H_p}(z) = \begin{bmatrix} H_{00}(z) & H_{01}(z) \\ H_{10}(z) & H_{11}(z) \end{bmatrix} \tag{2.14}$$

The **synthesis polyphase matrix** is denoted:

$$\mathbf{F_p}(z) = \begin{bmatrix} F_{00}(z) & F_{10}(z) \\ F_{01}(z) & F_{11}(z) \end{bmatrix} \tag{2.15}$$

where, for example, $H_{ij}(z)$ is the $j^{th}$ polyphase component of the $i^{th}$ filter. In the polyphase domain, the condition for a perfect reconstruction filter bank (see [42] for proof) is:

$$\mathbf{H_p}(z)\mathbf{F_p}(z) = \mathbf{I} \tag{2.16}$$

In the polyphase domain, the condition for orthonormality of the filter (see [42] for proof) is:

$$\mathbf{H_p}(z)\mathbf{H_p}^T(z^{-1}) = \mathbf{I} \tag{2.17}$$

Based on the definition given in eq. 2.3:

*For an orthonormal filter bank, both modulation and polyphase matrices are paraunitary (lossless) matrices.*

In this thesis, filter banks are implemented in the polyphase domain.

### 2.2.3 Lattice Factorization

One method to design an orthogonal filter bank is based on lattice factorization. The idea behind it is that any paraunitary matrix can be factorized into basic building blocks consisting of delays and Givens rotation matrices, $G_i$

$$\mathbf{G_i} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \tag{2.18}$$

where $\mathbf{G_i}$ is unitary. The lattice structure of the polyphase analysis filter $\mathbf{H_p}(z)$ becomes:

$$\mathbf{H_p}(z;\underline{\theta}) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \prod_{i=0}^{l-1} \left( \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \right) \begin{bmatrix} \cos\theta_l & -\sin\theta_l \\ \sin\theta_l & \cos\theta_l \end{bmatrix} \tag{2.19}$$

where the parameter

$$\underline{\theta} = [\theta_0, \theta_1, \ldots, \theta_l] \tag{2.20}$$

and we have $l$ delay blocks and $l+1$ Givens rotations. Such a structure imposes orthogonality on the filter bank design. The problem is now that of solving for $\underline{\theta}$ given some desired constraints, such as matching the frequency response of a filter.

### 2.2.4 Householder Factorization

Another way to factorize a paraunitary matrix is based on the Householder factorization. The basic building blocks in this case are the Householder matrices. Hence, the lattice structure of the polyphase analysis filter $\mathbf{H_p}(z)$ becomes:

$$\mathbf{H_p}(z) = \mathbf{A_0} \prod_{i=1}^{M} \mathbf{V_i}, \quad \text{where } \mathbf{V_i} = (\mathbf{I} - (1 - z^{-1}\mathbf{v_i}\mathbf{v_i^T})) \tag{2.21}$$

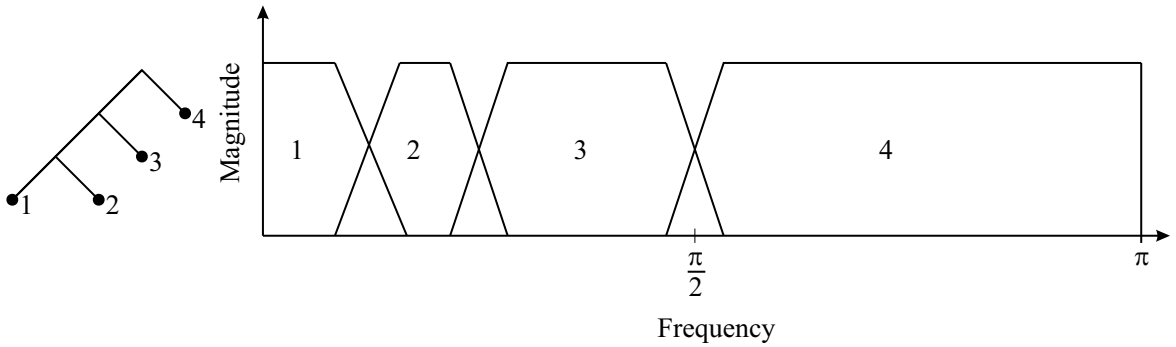


Figure 2-4: A filter bank iterated on the low-pass channel and the corresponding frequency partitioning.

$\mathbf{v_i}$, $i = 1, \ldots, M$ are unitary vectors and $\mathbf{A_0}$ is a constant orthogonal matrix.

### 2.2.5 Tree-Structured Filter Banks

The filter bank that we have seen so far is a simple 2-channel one. If one iterates it on the low-pass channel as shown in Figure 2-4, we obtain a constant-$Q$ octave band. In this simple case, the filter bank is said to have a dyadic structure meaning that at every iteration, the input spectrum is split in half.

It is fairly simple to extend this idea to arbitrary tree-structured filter banks by allowing iteration on the high-pass channel too. Figure 2-5 illustrates such an example. These structures are used to implement *wavelet packets*. An important issue worthy of mentioning is the following: when iterating on the high-pass channel, the spectrum, after downsampling, will be the mirrored version of the corresponding input spectrum. For example, when one is expecting the frequency band $[\pi/2, \pi]$ one will actually obtain $[\pi, \pi/2]$. This is something to keep in mind when extracting specific frequency bands using wavelet packets.

## 2.3 Wavelets and the Multiresolution Framework

Now that we have presented the basic concepts of filter banks, we turn our attention towards wavelets and show the relation between wavelets and filter banks within the multiresolution framework [42].

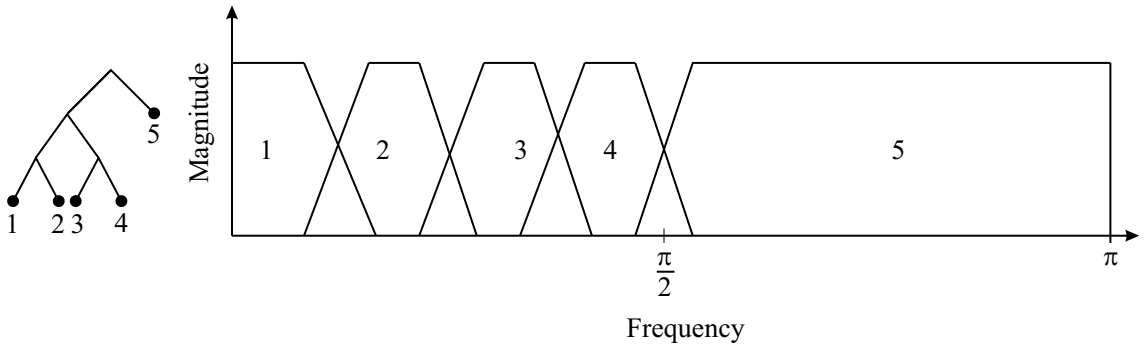


Figure 2-5: A tree-structured implementation of a filter bank and the corresponding frequency partitioning.

*In multiresolution analysis (MRA) the space $L_2(\mathcal{R})$ of signals can be spanned by successive spaces of detail at different resolutions.*

### 2.3.1 The Scaling Function

Consider the sequence of nested approximation subspaces:

$$\ldots V_1 \subset V_0 \subset V_{-1} \ldots \tag{2.22}$$

where $V_m$ are complete — span all the $L_2(\mathcal{R})$ space —, exclusive, and shift-invariant. There exists an orthonormal basis for $V_0$

$$\varphi(t-n) \quad | n \in \mathcal{Z}, \quad \text{where} \quad \varphi \in V_0 \tag{2.23}$$

more generally, there exists an orthonormal basis for $V_{-m}$

$$2^{m/2}\varphi(2^m t - n) \quad | n \in \mathcal{Z} \tag{2.24}$$

The multiresolution concept arises from the fact that the embedded spaces are scaled versions of the space $V_0$. This is also observed in the bases spanning the spaces as shown in Equations 2.23-2.24. $\varphi(t)$ is called a *scaling function.*

Dilation Equation 2.25, also known as a 2-scale equation, shows how the scaling property can be used to represent the scaling function $\varphi(t) \in V_0$ as a linear combina-

tion of the basis of $V_{-1}$, since $V_0 \subset V_{-1}$.

$$\varphi(t) = \sqrt{2} \sum_n h_0[n] \varphi(2t - n) \tag{2.25}$$

The Fourier transform of Equation 2.25 gives:

$$\Phi(\omega) = \frac{1}{\sqrt{2}} H_0(e^{j\omega/2}) \Phi(\omega/2) \tag{2.26}$$

where

$$H_0(e^{j\omega}) = \sum_{n \in \mathcal{Z}} h_0[n] e^{-j\omega n} \tag{2.27}$$

Equation 2.26 is a key one since it shows the connection between discrete-time sequences (or filters in this case) and continuous-time (scaling) functions. It gives an idea of how one can construct continuous-time wavelet bases from an iterated filter bank. We also obtain the following properties:

$$|H_0(e^{j\omega})|^2 + |H_0(e^{j(\omega+\pi)})|^2 = 2 \tag{2.28}$$

$$\sum_{k=-\infty}^{\infty} |\Phi(2\omega + 2k\pi)|^2 = 1 \tag{2.29}$$

$$|H_0(1)| = \sqrt{2} \tag{2.30}$$

$$H_0(-1) = 0 \tag{2.31}$$

### 2.3.2 The Wavelet Function

Every approximation space $V_m$ is complemented by a detail space. Hence, we consider the space $W_m$ such that it is the orthogonal complement of $V_m$ in $V_{m-1}$:

$$V_{m-1} = V_m \oplus W_m \tag{2.32}$$

By iterating this process, we obtain:

$$L_2(\mathcal{R}) = \bigoplus_{\in \mathcal{Z}} W_m \tag{2.33}$$

Equation 2.33 restates the statement that using multi-resolution analysis, the $L_2(\mathcal{R})$ is spanned by spaces of successive detail at different resolutions. Hence, we can introduce the orthonormal basis for $L_2(\mathcal{R})$:

$$\psi_{m,n}(t) = 2^{-m/2}\psi(2^{-m}t - n), \quad m, n \in \mathcal{Z} \tag{2.34}$$

and $\{\psi_{m,n}\}, n \in \mathcal{Z}$ is an orthonormal basis for $W_m$. $\psi(t)$ is called a *wavelet function* and $\psi(t) \in W_0 \subset V_{-1}$. Thus the wavelet function also satisfies a 2-scale equation:

$$\psi(t) = \sqrt{2}\sum_n h_1[n]\varphi(2t - n) \tag{2.35}$$

and the Fourier domain equivalent:

$$\Psi(\omega) = \frac{1}{\sqrt{2}}H_1(e^{j\omega/2})\Phi(\omega/2) \tag{2.36}$$

It can be shown [42] that $h_0[n]$ and $h_1[n]$ are low-pass and high-pass filters respectively of a 2-channel filter bank and the iterations of Equations 2.26 and 2.36 converge to piecewise smooth scaling and wavelet functions if the corresponding filters are *regular*.

*For a filter to be regular, it is necessary, but not sufficient, for it to have at least one zero at the aliasing frequency — $\pi$ for the dyadic 2-channel case.*

The concept of regularity is also related to *smoothness* of the scaling and wavelet functions where continuity and differentiability are desired features of the functions.

### 2.3.3 The Wavelet Series Computation

Given function $f(t) \in V_0$, we can write:

$$f(t) = \sum_{n=-\infty}^{\infty} a_0[n]\varphi(t-n) \tag{2.37}$$

The key idea is to project $f(t) \in V_0$ onto the approximation and detail spaces, $V_1$ and $W_1$ respectively, and then iterate the same process on $V_1$ and so forth. Using Equation 2.25, we obtain

$$a_1[n] = \sum_k h_0[k-2n]a_0[k] \tag{2.38}$$

$$d_1[n] = \sum_k h_1[k-2n]a_0[k] \tag{2.39}$$

Equations 2.38 and 2.39 show that the projection coefficients are obtained by filtering with $h_0[n]$ and $h_1[n]$ and downsampling by 2. This procedure is iterated to obtain the detail of the signal at several resolutions along with one coarse approximation. Using a discrete-time algorithm, we are able to implement a wavelet series expansion. The only issue is with Equation 2.37, which requires continuous-time processing. However, if the first approximation space, $V_0$, has a very fine resolution, then sampling the signal $f(t)$ is equivalent to Equation 2.37, [42]. Hence, we obtain:

$$a_0[n] \approx f[n] \tag{2.40}$$

We have previously mentioned that orthogonality of the filter bank filters leads to orthogonal transforms. This is an appealing feature since it reduces the correlation of the obtained expansion coefficients and also satisfies *Parseval's relation* where the energy of the transform is the same as that of the input signal.

## 2.4 Orthonormal Rational Filter Banks and Wavelets

The contents of this section are based on the work of Blu [5, 6, 7], where we refer the reader for further detail and proofs. The aim is to extend iterated dyadic filter banks to the more general rational case. This notion is first proposed in [32]. One motivation for such a direction is that a rational sampling factor will give a finer frequency resolution, which will be more suitable for the analysis of signals such as speech.

In the previous sections, we showed how an iterated filter bank is equivalent to a wavelet series expansion. This is an attractive feature since the wavelet transform can be implemented by a discrete-time algorithm. A rational filter bank, on the other hand, can only approximate a wavelet transform. In other words, rational sampling factors with FIR filters do not lead to a multi-resolution analysis and the iteration of a filter bank does not generate a unique limit function. This is due to the lack of the shift property:

*The 'wavelet' function corresponding to a rational filter bank is not shift-invariant. The shift error, however, can be made arbitrarily small when the function regularity increases.*

We are only concerned with the consequences, rather than the justifications, of such a statement. In [7], Blu designed an algorithm for the rational case. We briefly describe the algorithm in Section 4.4, while in this section, we restrict the discussion to the basics of orthonormal rational filter banks. We use the term *rational wavelets* as mentioned in [7], since the functions do not satisfy the shift-invariance property and effectively are not wavelets. Also, we concentrate on the rational sampling factor $M/(M-1)$, although the references study rational filter banks with general sampling factor $p/q$. Figure 2-6 illustrates a simple branch of a rational filter bank. For an input $x[n]$, the output $y[n]$ of such a branch is:

$$y[n] = \sum_k g[np - kq]x[k] \tag{2.41}$$

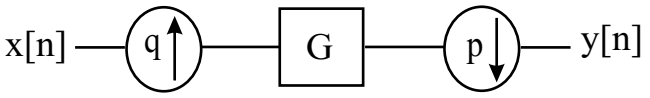


Figure 2-6: The basic branch of a rational filter bank of sampling factor $p/q$.

Figure 2-7 depicts the analysis bank of a general rational filter bank of sampling factor $p/q$ along with the corresponding frequency partitioning. The equations satisfying the orthonormality conditions for the rational filter bank illustrated in Figure 2-7 are given by:

$$\frac{1}{(M-1)M}\sum_{k=0}^{M-1} G(W_{M-1}^{s}W_{M}^{-k}z^{-1})G(W_{M}^{k}z) = \delta[s], \quad s = 0 \ldots M-2 \tag{2.42}$$

$$\frac{1}{M}\sum_{k=0}^{M-1} H(W_{1}^{s}W_{M}^{-k}z^{-1})H(W_{M}^{k}z) = \delta[s], \quad s = 0 \tag{2.43}$$

$$\sum_{k=0}^{M-1} H(W_{1}^{s}W_{M}^{k}z^{-(M-1)})G(W_{M}^{k}z) = \delta[s], \quad s = 0 \tag{2.44}$$

$$\frac{1}{(M-1)M}\sum_{l=0}^{M-2} |G(e^{j2\pi\frac{f+l}{M-1}})|^2 + \frac{1}{M}|H(e^{j2\pi f})|^2 = 1 \tag{2.45}$$

A rational filter bank that has $K$ regularity factors implies that $(\frac{z^{M}-1}{z-1}\frac{z^{M-1}-1}{z-1})^{K}$ is a factor of $G(z)$ (see [6] for a proof). This can be written as:

$$\sum_{n}(k+nM)^{r}g_{k}^{M}[n] = \sum_{n}(nM)^{r}g_{0}^{M}[n], \quad \forall 1 \le k \le M-1 \tag{2.46}$$

$$\sum_{n}(k+n(M-1))^{r}g_{k}^{M-1}[n] = \sum_{n}(n(M-1))^{r}g_{0}^{M-1}[n], \quad \forall 1 \le k \le M-2 \tag{2.47}$$

$$r = 0 \ldots K-1$$

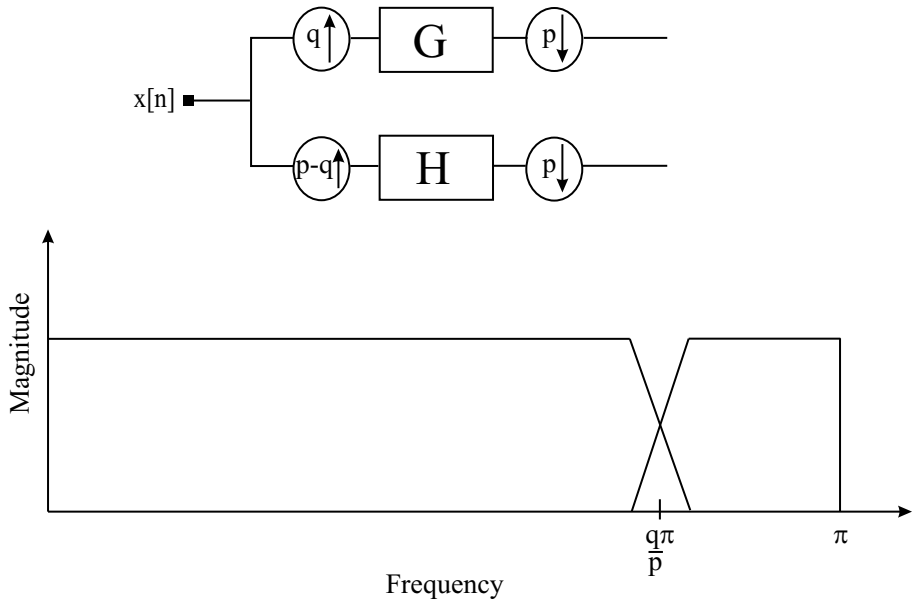


Figure 2-7: The analysis channel of a rational filter bank of sampling factor $p/q$ along with the corresponding frequency partitioning.

# Chapter 3

# Problem Specification

In this chapter, we describe the problem at hand: *Feature Extraction for Phonetic Classification*, the limitations of the commonly used MFCC, along with the suggested alternative: a *Filter Bank and Wavelet Framework*. We discuss the features studied in the thesis that make such a framework appealing for speech analysis.

## 3.1 Problem

The performance of an ASR system is heavily reliant on the adequate design of an acoustic model. Acoustic-phonetic modelling is concerned with the link between the abstract phonetic representation and the physical speech signal. An acoustic model typically attempts to model the probability of an acoustic feature given the speech signal, and hence is, in turn, dependent on adequate feature extraction. It is this specific point that we study in the thesis.

*Feature extraction, also known as front-end processing, is the process of measuring acoustic observations that compactly describe the speech segment preserving the necessary information for discriminatory analysis. The acoustic observations are also required for the complete parameterization of the acoustic models.*

An acoustic model usually takes into account the different sources of variability; those due to linguistic context such as coarticulation effects and non-linguistic con-

text such as speaker gender and dialect. However, in this thesis, we restrict the task to *context-independent phonetic classification* and assume that we are given the segmented speech — hence dealing with statistically independent phonetic models — and asked to classify individual phonetic segments.

Having said that, our aim is to investigate new acoustic observations that can be extracted from the signal. There are many approaches to feature extraction such as capturing knowledge-based or event-based acoustic parameters — cues for voicing, place and manner of articulation — where the emphasis is on acoustic-phonetics [1, 2, 4, 28]. However, the most commonly extracted measurement in ASR front-end analysis remains the MFCC. Indeed, the MFCC dominates despite its obvious limitations:

- It is inherently a short-time spectral representation based on Fourier analysis which is limited in its time-frequency representation. More specifically, the STFT is used to analyse the signal with a fixed resolution which is especially inadequate for transient signals.
- Its computation is based on the inner product of the signal power spectrum with, typically, 40 triangular band-pass filters where the selection of the triangular filter shape is quasi-arbitrary. This validates the investigation of other filter designs.
- Its performance is not robust under noisy conditions.

In this thesis, we stress and seek solutions for the two initial points. The question that naturally poses itself is:

*How do we overcome the shortcomings of the STFT-based MFCC and what do we hope to accomplish by doing this?*

## 3.2 Proposed Solution: Wavelets and Filter Banks

In answer to the questions posed in the previous section, we propose a filter bank and wavelet framework as a solution.

Over the past two decades, wavelets and filter banks have been studied as potential alternatives to STFT analysis. In Chapter 2 we gave an overview of filter bank and wavelet theory. Two important points should be emphasized:

- The wavelet transform permits a multiresolution signal analysis. By exploiting the time/frequency resolution trade-off, we are able to obtain a good signal representation that captures transients as well as coarse approximations. Wavelets form basis functions that are well localized in time and frequency unlike the sinusoidal functions in the Fourier analysis.

- A filter bank implemented with FIR filters, upsamplers, and downsamplers can be used to carry out the wavelet transform efficiently through a discrete-time algorithm.

The overview also showed how the basic filter bank with integer sampling factor is used to generate octave bands and arbitrary tree-structures that implement wavelet packets. The integer sampling factor filter bank is extended to the more general rational sampling factor. Keeping in mind that such filter banks do not lead to shift-invariant functions in the limit, an algorithm is presented [7] that designs rational filter banks and consequently *rational wavelets* with minimum shift error. We restricted the overview, for the most part, to orthonormal filter banks and provided insight on how one can design such systems imposing orthonormality through structure such as lattice and Householder factorization.

In this thesis, as in most work on the topic, wavelets and filter banks are considered to be strongly related since a wavelet transform can be implemented using filter banks. With proper design, we are able to obtain desired filter properties such as lower attenuation bands and regularity.

From the mentioned overview description, we can already see where the flexibility of such a framework illustrated in Figure 3-1 might lie.

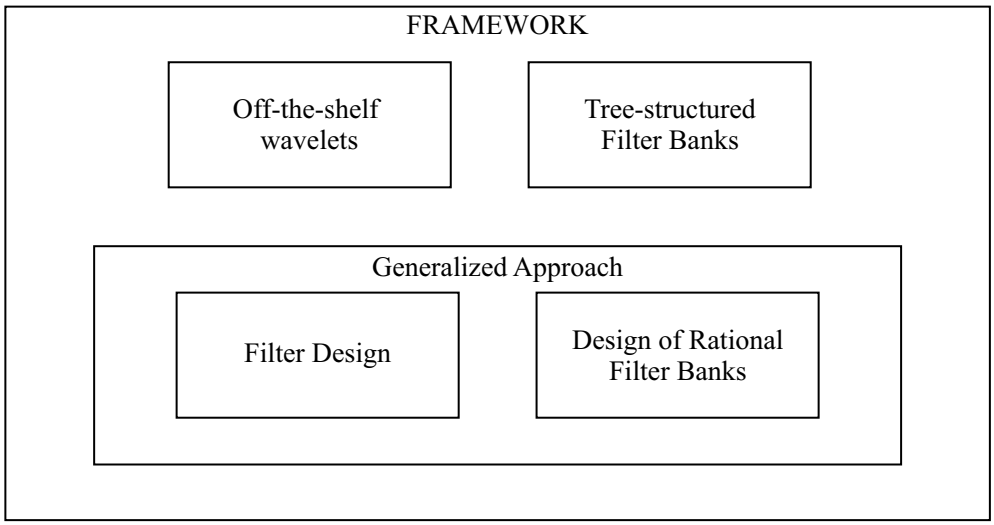


Figure 3-1: An illustration of the proposed wavelet and filter bank framework for feature extraction.

### 3.2.1 Flexibility in Filter Design

As reported in the literature, most of the filter banks implemented for speech analysis make use of off-the-shelf wavelets such as the Daubechies. While this is straightforward from a design point of view, it does not necessarily lead to adequate filters such as ones with sharp cutoff and low attenuation in the stopband. Furthermore, opportunities for filter optimization and customization arise with the flexibility of filter design.

Given constraints such as orthonormality and desired filter features such as regularity we adopt two approaches for filter design. The first method attempts to match a desired filter shape while the second minimizes the attenuation band. There are definitely limitations to the proposed algorithms and we do not claim the ability to design *any arbitrary* filter. In fact, the first algorithm is relatively simple and based on a constrained quasi-Newton method. Despite their limitations, the two methods give insight into the advantages of designing task-optimized filters.

### 3.2.2 Flexibility in Frequency Partitioning

For the most part, the literature mentions filter banks implemented for speech analysis that generate octave bands by iteration of the low-pass channel, or wavelet packets by iteration of both channels. The frequency partitions obtained in these cases, especially in the former, are not suitable for the task. This is because the octave band filter bank does not have a good frequency resolution at the high frequency bands. Trying to solve this problem by using wavelet packets will lead to a loss of the constant-$Q$ characteristics. Again this is not ideal and it motivated our interest in filter banks customized for speech analysis. The objective is to develop filter banks that have a fine frequency resolution and can mimic the auditory filters. Fairly recently, there has been work done on filter banks with rational sampling factor. Iterated rational filter banks give more flexibility in the frequency partitioning. More specifically, where the iterated dyadic filter banks are restricted to a single $Q$ factor value, iterated rational filter banks can be designed to meet a wide range of $Q$ values such as the one that matches the Bark scale — see Section 4.4.

# Chapter 4

# Implementation

In this chapter we describe the wavelet and filter bank framework proposed for the task of phonetic classification. The basic building blocks are based on the concepts presented in Chapter 2. We describe the implementation step by step starting with the acoustic measurement extracted using wavelet analysis. Then we explain the initial implementation that used off-the-shelf wavelets and a dyadic sampling. We propose filter design and rational sampling to overcome the limitations of the initial implementation.

## 4.1 The Acoustic Measurement

Figure 4-1 illustrates the stages involved in the computation of the frame-based acoustic measurement.

**Stage 1.** Computes the wavelet transform of the input speech frame. In all the experiments related to the wavelet-based acoustic measurement, the frame rate is 200 frames per second (5 ms per frame) with a frame size of 20 ms. In this stage, we also need to specify the wavelet type and the frequency decomposition, whether we are using wavelet packets or rational sampling.

**Stage 2.** Computes the $L_2(\mathcal{R})$ norm of each frequency band giving a total of $N$ energy coefficients where $N$ is the number of frequency bands analysed in

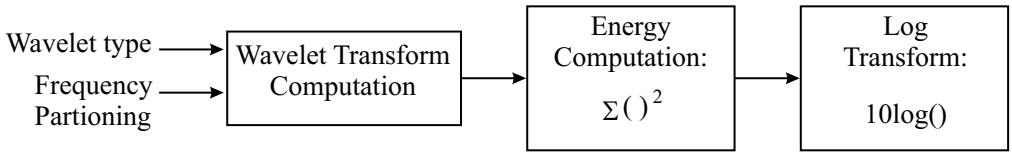


Figure 4-1: A flowchart of the computational stages for the wavelet-based acoustic measurement.

the previous stage.

**Stage 3.** Computes the log transform of the energy coefficients. We note empirical evidence that the dynamic range reduction of the coefficients also made them more robust and consistent.

The result of the three stages is an $N$-dimensional acoustic measurement. For simplicity we will refer to the acoustic measurement as wavelet-based coefficients (WBCs).

Initially, we adopted a slightly different acoustic measurement with a $DCT$ stage inserted at the end. The transform was used to whiten the feature space and decrease its dimensionality to 14, for example. However the presented measurement gave more consistent results.

This $N$-dimensional measurement is used to generate a segmental feature of dimension ($5N$+6), which is extracted over given phonetic segments. The segmental feature consisted of:

- 3 WBC averages computed over the segment in a 3-4-3 proportion
- 2 WBC derivatives computed using linear least-squared error regression over a time frame of 40 ms centered at the start and end of the segment
- 3 average energies computed over the segment similarly to the WBC averages
- 2 derivative energies computed similarly to the WBC derivatives
- a log duration

This configuration is chosen to match the one used for the baseline segmental feature. Since $N$ can range between 18 and 30 coefficients, the dimensionality of the

| Wavelet type | # Zeros at $\omega = \pi$ | Filters length |
|---|---|---|
| Haar | 1 | 2 |
| Daubechies (Daub2) | 2 | 4 |
| Daubechies (Daub4) | 4 | 8 |
| Daubechies (Daub6) | 6 | 12 |
| Daubechies (Daub10) | 10 | 20 |
| Daubechies (Daub12) | 12 | 24 |

Table 4.1: Description of the implemented off-the-shelf wavelets.

segmental feature vector can be anywhere between 96 and 156. Principal component analysis is used to project the feature space onto a lower dimension ranging between 70 and 90 as well as whiten it.

We refer the reader to Section 5.1.3 for a description of the baseline acoustic measurement.

## 4.2 Off-The-Shelf Wavelets and Tree-Structured Filter Banks

After designing the acoustic measurement, the first step was to test it on off-the-shelf wavelets. Tree-structured filter banks are used to obtain the frequency partitions. Table 4.1 lists the implemented wavelets along with a brief description. All the wavelets excluding the Haar are from the Daubechies family. The Haar and the Daub2 are implemented purely to test the algorithm on the simplest filters. The number of zeros at $\pi$ is attributed to the low-pass filter corresponding to the wavelet.

From Figure 4-2, we notice that the larger the number of zeros at $\pi$, the narrower the transition region and the sharper the filter cutoff. Sharp cutoff is a desired characteristic of filters since it implies good frequency selectivity. However, we also point out that the filter length should be increased to achieve reasonable cutoffs.

We also experimented with several frequency partitions obtained with tree-structured implementations of the filter banks. Figure 4-3 shows a tree structure that is used to

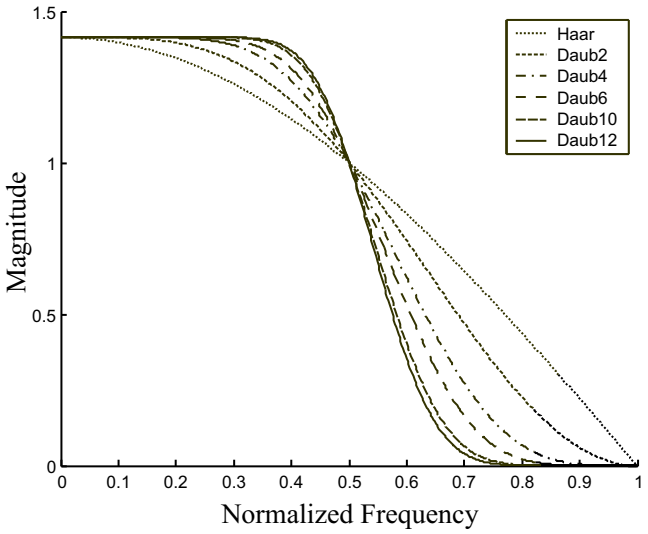


Figure 4-2: The low-pass filters corresponding to the Haar, Daub2, Daub4, Daub6, Daub10, and Daub12.

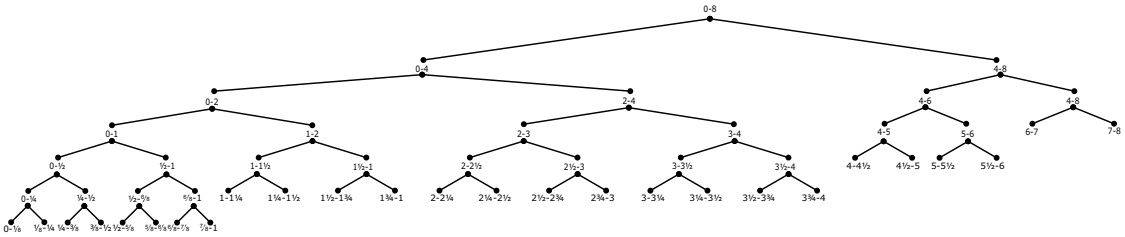

Figure 4-3: A tree structure generating a 26-band filter bank.

obtain a specific frequency partitioning of 26 bands and Table 4.2 describes, in more detail, the generated band-pass filters.

Table 4.3 shows the wavelet types and the corresponding frequency partitions — represented by the number of filters — that are implemented for each. See Appendix B for a detailed description of the rest of the frequency partitions.

## 4.3 Filter Design

In this section, we describe the two filter design techniques that are implemented. The first is based on matching a desired filter shape while the second minimizes the attenuation in the stopband. The designed filters are then tested in iterated 2-channel filter banks with a downsampling factor of 2.

| Filter # | Lower cutoff frequency (Hz) | Upper cutoff frequency (Hz) | Bandwidth (Hz) |
|---|---|---|---|
| 1 | 0 | 125 | 125 |
| 2 | 125 | 250 | 125 |
| 3 | 250 | 375 | 125 |
| 4 | 375 | 500 | 125 |
| 5 | 500 | 625 | 125 |
| 6 | 625 | 750 | 125 |
| 7 | 750 | 875 | 125 |
| 8 | 875 | 1000 | 125 |
| 9 | 1000 | 1250 | 250 |
| 10 | 1250 | 1500 | 250 |
| 11 | 1500 | 1750 | 250 |
| 12 | 1750 | 2000 | 250 |
| 13 | 2000 | 2250 | 250 |
| 14 | 2250 | 2500 | 250 |
| 15 | 2500 | 2750 | 250 |
| 16 | 2750 | 3000 | 250 |
| 17 | 3000 | 3250 | 250 |
| 18 | 3250 | 3500 | 250 |
| 19 | 3500 | 3750 | 250 |
| 20 | 3750 | 4000 | 250 |
| 21 | 4000 | 4500 | 500 |
| 22 | 4500 | 5000 | 500 |
| 23 | 5000 | 5500 | 500 |
| 24 | 5500 | 6000 | 500 |
| 25 | 6000 | 7000 | 1000 |
| 26 | 7000 | 8000 | 1000 |

Table 4.2: The frequency bands of the 26 filters obtained with the tree structure in Figure 4-3.

| Wavelets | # Filters |
|---|---|
| Haar, Daub2 | 26 |
| Daub4, Daub6, Daub10, Daub12 | 24, 26, 28, 30 |

Table 4.3: The implemented wavelets with the corresponding number of filters that are tested.

### 4.3.1 Filter Matching

Filter matching is implemented using a simple method that minimizes the difference in modulus between the designed and desired filter given some constraint. The minimization is formulated in the frequency domain. We only design orthogonal filter banks where the analysis and synthesis systems can be modeled as paraunitary matrices, $\mathbf{H_p}$ and $\mathbf{F_p}$. In Chapter 2, Section 2.2.3, we showed that a paraunitary matrix can be factorized into smaller building blocks that are a function of $\theta_i$. If we denote the magnitude response of the desired filter as $|H_d(\omega)|$, the problem becomes one of minimizing:

$$C(\underline{\theta}, \lambda) = \int_{-\infty}^{\infty} ||H_d(\omega)| - \lambda |H_p(\omega; \underline{\theta})||^2 d\omega \tag{4.1}$$

given the following constraint:

$$\frac{d^l H_0(\omega)}{d\omega^l}|_{\omega=\pi} = 0 \quad l = 0 \ldots N-1 \tag{4.2}$$

where $H_0(\omega)$ is the frequency response of the analysis low-pass filter $h_0[n]$ and $N$ is the number of desired vanishing moments which is also the number of zeros at $\pi$ for $H_0(\omega)$. Orthogonality of the filter bank is constrained by the lattice factorization. The algorithm implementation is based on a sequential quadratic programming method for constrained optimization.

We tested the algorithm by matching it to two desired signals: the Butterworth filter of order 10 and cutoff frequency $\pi/2$ and the ideal low-pass filter. The resulting filters are described in Table 4.4. They are denoted Match_{*Filter it matches*}. As in Section 4.2, each designed filter is tested with the 26-band tree-structured filter bank. Figure 4-4 shows a 30-tap filter that is designed to match the Butterworth filter given the constraints of orthogonality and having 3 zeros at $\pi$.

### 4.3.2 Attenuation Minimization

This method is fully described in Section 4.4 for the general case of orthonormal rational filter banks. Attenuation minimization is straightforward to implement for

| Filter name | # zeros at $\omega = \pi$ | Desired Filter | Filter length |
|---|---|---|---|
| Match_Butterworth | 3 | Butterworth of order 10 and cutoff frequency $\pi/2$ | 30 |
| Match_Ideal | 3 | Ideal filter | 30 |

Table 4.4: Description of the filters designed using the matching technique.

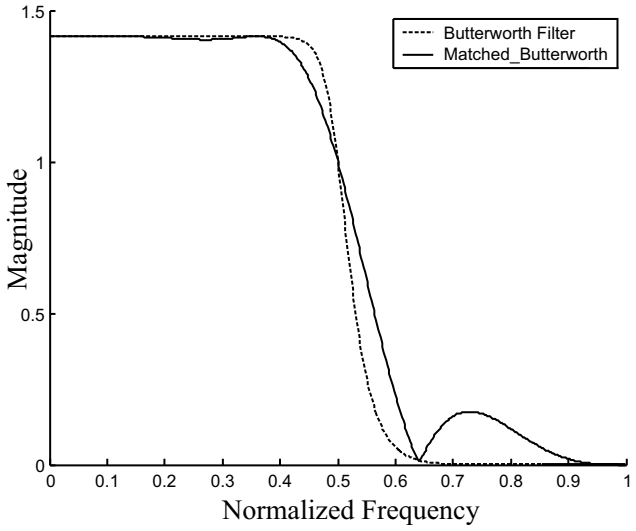


Figure 4-4: A low-pass filter designed to match the Butterworth filter of order 10.

the simple dyadic case.

The key point behind the technique is to match the filter to an ideal low-pass filter by minimizing the difference in modulus between the two filters. However, as we will see in Section 4.4, according to the power complementary Equation 4.5, setting the values of the filter in the passband will uniquely set its values in the stopband. Thus, the problem is reduced to a minimization of attenuation in the stopband although it originated as a filter matching problem also.

The filters that are designed using the attenuation minimization algorithm are listed in Table 4.5. All the filters have a regularity order set to 1 in order to guarantee convergence of the algorithm. We refer to the designed filters by the generic name Filter_#. Figure 4-5 illustrates the magnitude response of Filter_5, a 30-tap filter designed using this method. For comparison we include the ideal filter and the Daub12 filter. Unlike Daub12, Filter_5 exhibits a very good attenuation in the stopband.

All the filters, are tested with the 26-band tree-structured filter bank.

## 4.4 Orthonormal Rational Filter Banks and Wavelets

In this section, we describe the design of orthonormal rational filter banks with sampling factor $p/q$ or $M/(M-1)$ as in the case we adopt.

The filter banks that we have seen so far are of the octave band type or implement the more general arbitrary tree structures. If we define, the $Q$-factor as the ratio of

| Filter name | Regularity order | Filter length |
|---|---|---|
| Filter_1 | 1 | 10 |
| Filter_2 | 1 | 16 |
| Filter_3 | 1 | 20 |
| Filter_4 | 1 | 26 |
| Filter_5 | 1 | 30 |
| Filter_6 | 1 | 34 |

Table 4.5: Description of the filters designed using the attenuation minimization technique.

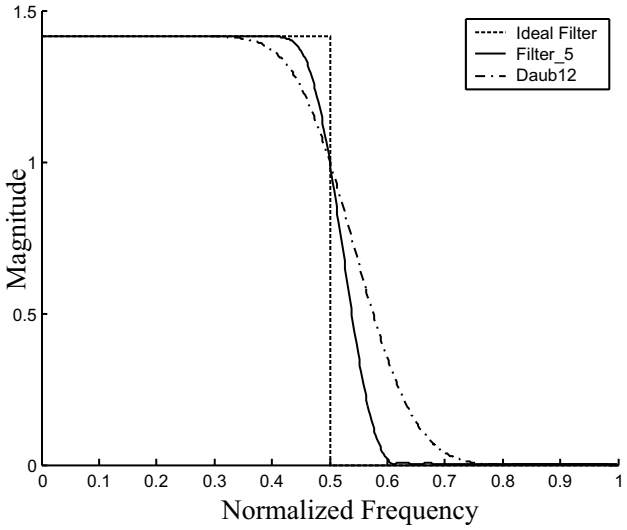


Figure 4-5: The designed low-pass filter with the corresponding ideal filter it matches to and the Daub12 filter.

the bandwidth to the center frequency of a band, then the value obtained for the octave band is 2/3. Suppose now that we want to consider a more general case, in the hope of obtaining a $Q$-factor tunable to the human auditory system, for example. As mentioned before, in the dyadic case, the input spectrum is split in half at each iteration. This can surely be extended to include a more general partitioning ratio such as $(M-1)/M$. Hence, at each iteration of the filter bank, the spectrum is split into the ratios $1/M$ and $(M-1)/M$ instead of $1/2$ and $1/2$ as in the dyadic case. The next section shows how this can be done using uniform $M$-band filter banks.

### 4.4.1 Rational Filter Banks from Uniform $M$-band Filter Banks

When we first attempted to design a filter bank with a rational sampling factor, we obtained Figure 4-6 and the corresponding frequency partitioning. At each iteration of the filter bank, we have a uniform $M$-band analysis filter bank which decomposes the spectrum into equipartitions of bandwidths $\frac{\pi}{M}$ each. Then an (*M-1*)-band synthesis filter bank generates a single output by upsampling and interpolating the *M-1* inputs. In the frequency domain, this has the effect of combining the first *M-1* bands into one, resulting finally with two frequency bands: $[0, \frac{M-1}{M}\pi]$ and $[\frac{M-1}{M}\pi, \pi]$. All of this is simply the analysis channel of the rational filter bank, and should not be confused

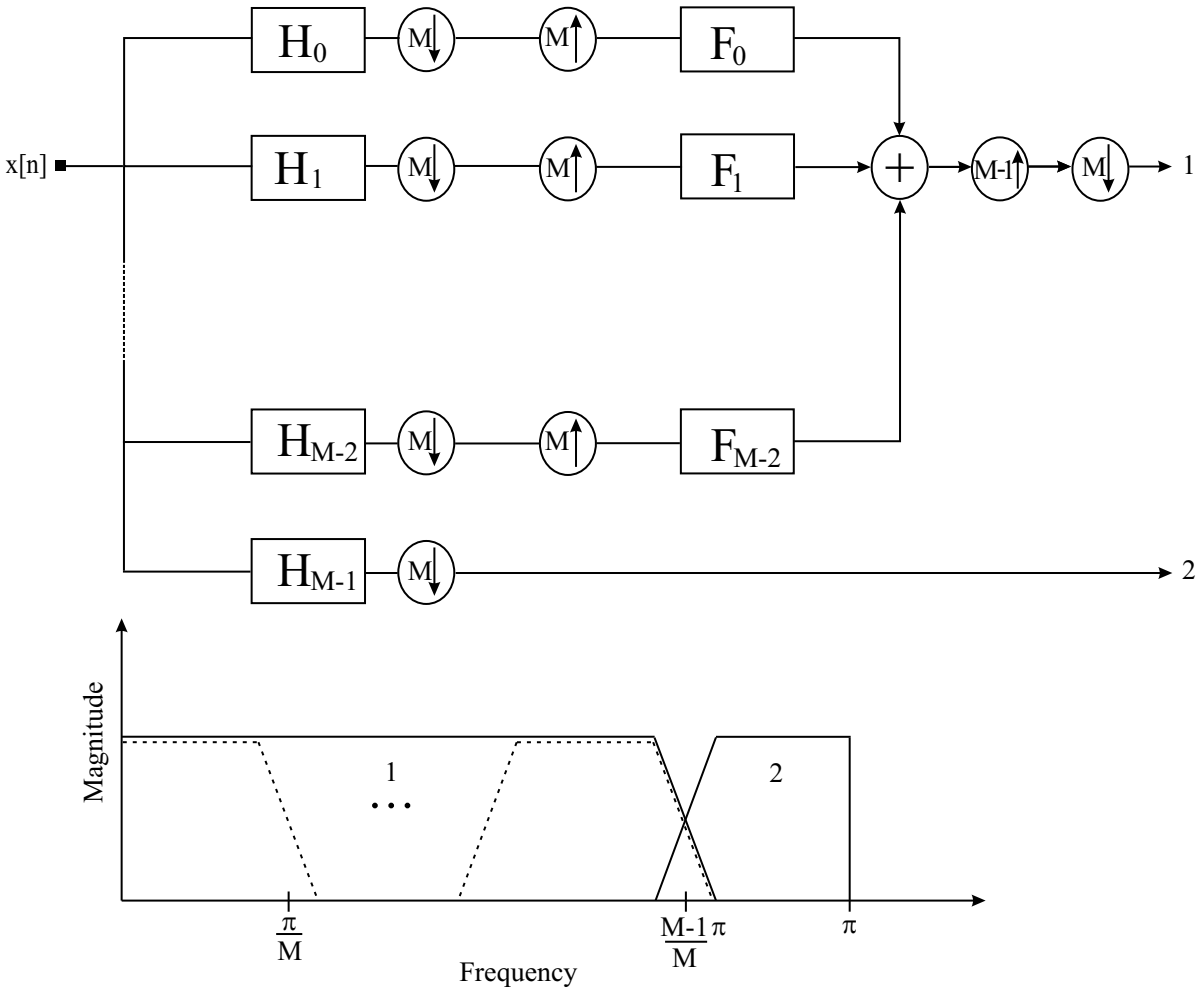


Figure 4-6: An $M$-band uniform filter bank implementing the rational sampling factor $M/(M-1)$ and the corresponding frequency partitioning after one iteration.

with a full analysis/synthesis filter bank.

To calculate the $Q$-factor corresponding to such a filter bank, we refer to the frequency partitioning after one iteration: the bandwidth of the highest frequency band is $\frac{\pi}{M}$ and the center frequency of that band is $\frac{\pi}{2M} + \frac{M-1}{M}\pi$. The expression for $Q$, in this case, becomes:

$$Q = \frac{bandwidth}{center\ frequency} = \frac{\frac{\pi}{M}}{\frac{\pi}{2M} + \frac{M-1}{M}\pi} = \frac{\frac{1}{M}}{\frac{1}{2M} + \frac{M-1}{M}} = \frac{1}{M - 1/2} \qquad (4.3)$$

With this formula, we can obtain the $M$ that matches a desired $Q$. For example, if we wanted to obtain the $Q$-factor of the MFCC which is 0.1376, Equation 4.3 gives

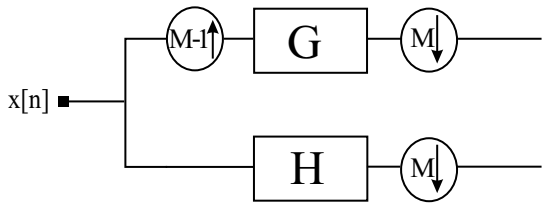


Figure 4-7: The equivalent filter bank of rational sampling factor $M/(M-1)$.

$M \approx 7.76$, which is rounded to 8. The resulting sampling ratio is 8/7. Another interesting sampling ratio is 6/5 which closely approximates the Bark scale analysis resulting in a filter bank that naturally mimics the human auditory system.

The filter bank depicted in Figure 4-6 requires the design of $M$ analysis filters and *M-1* synthesis filters solely for the analysis bank. The filter bank architecture itself seems bulky requiring a reconstruction stage at each analysis iteration in order to recombine the first *M-1* frequency bands. Hence, although our first approach gave an indication of how to obtain rational filter banks and a formula that links the number of bands $M$ to the $Q$-factor, it was cumbersome to design. It has been shown that a filter bank of the form given in Figure 4-6 can actually be put in the form illustrated in Figure 4-7 [32].

Thus, we are now dealing with a rational filter bank. Fortunately, there has been research on perfect reconstruction filter banks with rational sampling factors [32] and non-uniform multirate filter banks [27]. More specifically, we have already seen this specific structure in Section 2.4, as part of the work of Blu [7], which we describe briefly in the following sections.

### 4.4.2 Design Algorithm for the Low-Pass Filter

The motivation behind this algorithm is to find the best frequency selective low-pass filter $G(z)$ given constraints of orthogonality and regularity of the filter bank. The degree of selectivity is defined as the difference in modulus between $G(z)$ and the

ideal filter $D(z)$ whose normalized frequency response is:

$$|D(e^{j\omega})| = \begin{cases} \sqrt{(M-1)M} & \omega \in [-\frac{\pi}{M}, \frac{\pi}{M}] \\ 0 & elsewhere \end{cases} \tag{4.4}$$

This choice is such that the rational branch shown in Figure 2-6 will discard frequencies above $\pi\frac{M-1}{M}$.

In Section 2.4, we specified the conditions for orthonormality (Equations 2.42-2.44) and the implications of regularity (Equation 2.46).

The problem is reduced to a minimization with constraints. In order to further simplify it, we make the following observation: setting $s$ to zero in Equation 2.42, will give the power complementary Equation:

$$\sum_{k=0}^{M-1} |G(e^{2j\pi(f+k)/(M-1)})|^2 = (M-1)M \tag{4.5}$$

This equation shows that in order to minimize the difference between $G(z)$ and $D(z)$, we only need to minimize the attenuation band of $G(z)$ since the values of $|G(e^{j\omega})|^2$ in the attenuation band $[\frac{\pi}{M}+\epsilon, \pi]$ dictate those in the passband $[0, \frac{\pi}{M}-\epsilon]$. The filter matching problem is now reduced to minimization of the attenuation band. The $L_2(\mathcal{R})$ norm can be used to quantify the attenuation:

$$att_{f_0}(G) = \int_{f_0}^{1/2} |G(e^{2j\pi f})|^2 df \quad f_0 = \frac{\omega_0}{2\pi} = \frac{1}{2M} + \epsilon > \frac{1}{2M} \tag{4.6}$$

The minimization algorithm is formulated using the Lagrange multiplier method for constrained minimization where we minimize:

$$J(G) = function(G) - \lambda(constraints\ of\ orthonormality\ and\ regularity) \tag{4.7}$$

In order to ensure perfect reconstruction of the filter bank, Blu devised a recursive implementation of the algorithm where the condition for convergence is minimal perfect reconstruction error [7]. By doing so, the algorithm itself focuses on the at-

tenuation while the iterations minimize the reconstruction error. The algorithm is initialized with a filter $G$ of degree $N$, where it is insensitive to the initialization and its convergence, in the limit, is independent of the filter choice. The procedure described is repeated until the reconstruction error becomes smaller than a predefined value. The value selected is $10^{-11}$ similar to the choice of Blu [7].

The convergence of the algorithm is not always guaranteed unless the downsampling factor is 1. If the downsampling factor is larger than 1, then convergence is again guaranteed if the regularity order is set to 1. In our implementation, we set the regularity order to 1 since it suited the application, and it also ensured convergence.

### 4.4.3 Solution for the High-Pass Filter

We have shown how to design the low-pass filter, $G$, in the rational filter bank. We now turn our attention to the high-pass filter, $H$. It can be shown that if the difference between the upsampling and downsampling factors is 1, as in our case, then there is a unique high-pass filter corresponding to the designed low-pass filter [7]. We know that:

- $\mathbf{G}$ is designed to be paraunitary to a high accuracy where $\mathbf{G}$ is the polyphase representation of $G(z)$ of size $(M-1) \times M$.

- the rational filter bank is orthonormal and can also be represented by a paraunitary matrix $\mathbf{\Gamma_p} = [\mathbf{G}\ \ \mathbf{H}]^T$ where $\mathbf{H}$ is the polyphase representation of $H(z)$ and is of size $1 \times M$. In this case, it is a row vector.

From Section 2.2.4, we know that both $\mathbf{\Gamma_p}$ and $\mathbf{G}$ can be factorized into:

$$\mathbf{A_0} \prod_{i=1}^{M} \mathbf{V_i} = \mathbf{A_0}(\mathbf{I} - (1 - z^{-1}\mathbf{v_i}\mathbf{v_i^T}))$$

Since we are able to obtain $\mathbf{G}$, we factorize it into the form given above to obtain a rectangular constant matrix $\mathbf{A_0}$ of size $(M-1) \times M$. The key to finding $\mathbf{H}$ is to complete $\mathbf{A_0}$ so that it becomes a square orthonormal matrix, that is by adding a

| Filter name | Regularity order | Low-pass filter length | High-pass filter length | # Filters |
|---|---|---|---|---|
| Filter_6/5 | 1 | 194 | 44 | 18 |
| Filter_7/6 | 1 | 226 | 43 | 20 |
| Filter_8/7 | 1 | 226 | 41 | 22 |
| Filter_10/9 | 1 | 191 | 32 | 26 |

Table 4.6: Description of the designed filters for the rational filter banks.

single row to it, in this case. Hence $\mathbf{\Gamma_P}$ can now be written as:

$$\mathbf{\Gamma_p} = [\mathbf{A_0}\ \mathbf{H_{row}}]^T.(\mathbf{I} - (1 - z^{-1}\mathbf{v_1}\mathbf{v_1^T})) \ldots (\mathbf{I} - (1 - z^{-1}\mathbf{v_M}\mathbf{v_M^T})) \tag{4.8}$$

where $\mathbf{H_{row}}$ is chosen such that it adds one line to the rectangular matrix $\mathbf{A_0}$ to make it a square orthonormal matrix. Now that we have $\mathbf{\Gamma_P}$, we can compute $\mathbf{H}$ as:

$$\mathbf{H} = \mathbf{H_{row}}.(\mathbf{I} - (1 - z^{-1}\mathbf{v_1}\mathbf{v_1^T})) \ldots (\mathbf{I} - (1 - z^{-1}\mathbf{v_M}\mathbf{v_M^T})) \tag{4.9}$$

### 4.4.4 Implementation

Using the algorithm as described, we design the rational filter banks listed in Table 4.6. We refer to the filters as Filter_{*sampling factor*}. The regularity order is again set to 1 for all. The rational filter banks are iterated on the low-pass channel $N$ times to generate $N$ bands. We chose to iterate until the lower cutoff of the last band-pass filter obtained is at or close to 1 kHz. We then used Filter_5 designed in Section 4.3.2 to divide the 0-1 kHz region into 8 equipartitions. This is done in order to obtain a frequency partition that models the critical-band spectral resolution. Another observation is that the length of the filters is quite large. This is necessary in order to obtain filters with narrow passbands and also good frequency selectivity as is the case here.

Figure 4-8 shows the low-pass and high-pass filters for the rational filter bank of sampling factor 8/7.

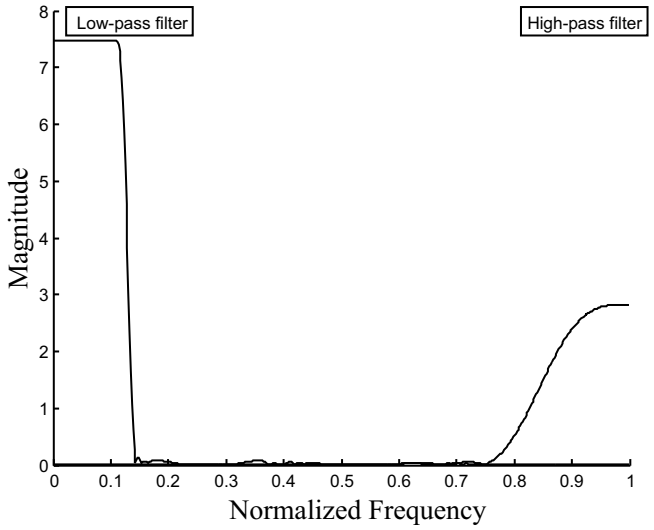


Figure 4-8: The low-pass and high-pass filters corresponding to the rational filter bank of sampling factor 8/7.

# Chapter 5

# Evaluation

## 5.1 Experimental Setup

In this chapter, we evaluate the wavelet and filter bank framework on the task of phonetic classification. First, we describe the experimental setup including the TIMIT corpus and the classifier. Then we present the baseline classifier used as a reference point in the analysis of our results. Finally we provide an evaluation of the experiments performed and a comparison with the baseline classifier and other results reported in the literature.

### 5.1.1 The TIMIT Corpus

TIMIT is a corpus of continuous speech. Its design and preparation was a joint collaboration between Texas Instruments, and the Massachusetts Institute of Technology [33]. It includes speech from 630 speakers, 438 males and 192 females, representing 8 major dialect groups of American English. Each speaker recorded 10 phonetically-rich sentences. Along with the speech waveform files, the corpus includes the corresponding time-aligned phonetic transcriptions.

There are 61 phones in the TIMIT corpus. Their IPA and ARPAbet symbols are shown in Table 5.1. We will refer to the phones through their ARPAbet symbols.

Following common practice, the 61 phone labels are collapsed into 39 labels prior

| IPA | TIMIT | Example | IPA | TIMIT | Example |
|---|---|---|---|---|---|
| [ɑ] | aa | b*o*b | [ɨ] | ix | deb*i*t |
| [æ] | ae | b*a*t | [i] | iy | b*ee*t |
| [ʌ] | ah | b*u*t | [ǰ] | jh | *j*oke |
| [ɔ] | ao | b*ough*t | [k] | k | *k*ey |
| [ɑʷ] | aw | b*ou*t | [k$^{\square}$] | kcl | k closure |
| [ə] | ax | *a*bout | [l] | l | *l*ay |
| [ə$^{h}$] | ax-h | p*o*tato | [m] | m | *mom* |
| [ɚ] | axr | butt*er* | [n] | n | *n*oo*n* |
| [ɑʸ] | ay | b*i*te | [ŋ] | ng | si*ng* |
| [b] | b | *b*ee | [ɾ̃] | nx | wi*nn*er |
| [b$^{\square}$] | bcl | b closure | [o] | ow | b*oa*t |
| [č] | ch | *ch*oke | [ɔʸ] | oy | b*oy* |
| [d] | d | *d*ay | [p] | p | *p*ea |
| [d$^{\square}$] | dcl | d closure | [▯] | pau | pause |
| [ð] | dh | *th*en | [p$^{\square}$] | pcl | p closure |
| [ɾ] | dx | mu*dd*y | [ʔ] | q | glottal stop |
| [ɛ] | eh | b*e*t | [r] | r | *r*ay |
| [l̩] | el | bott*le* | [s] | s | *s*ea |
| [m̩] | em | bott*om* | [š] | sh | *sh*e |
| [n̩] | en | butt*on* | [t] | t | *t*ea |
| [ŋ̍] | eng | Wash*ing*ton | [t$^{\square}$] | tcl | t closure |
| [▯̍] | epi | epenthetic silence | [θ] | th | *th*in |
| [ɝ] | er | b*ir*d | [ʊ] | uh | b*oo*k |
| [e] | ey | b*ai*t | [u] | uw | b*oo*t |
| [f] | f | *f*in | [ü] | ux | t*oo*t |
| [g] | g | *g*ay | [v] | v | *v*an |
| [g$^{\square}$] | gcl | g closure | [w] | w | *w*ay |
| [h] | hh | *h*ay | [y] | y | *y*acht |
| [ɦ] | hv | a*h*ead | [z] | z | *z*one |
| [ɪ] | ih | b*i*t | [ž] | zh | a*z*ure |
| - | h# | utterance initial and final silence | | | |

Table 5.1: IPA and ARPAbet symbols for the phones in the TIMIT corpus with sample occurrences.

to scoring and the glottal stops are ignored [35]. The mapping is shown in Table 5.2.

The sentences in TIMIT fall under three categories: dialect (SA), phonetically-compact (SX), and phonetically-diverse (SI).

- The SA category consists of 2 sentences read by all 630 speakers and is typically used to study the dialectical variation between speakers. It is excluded from the training, development, and test data sets.

- The SX category consists of 450 sentences that are phonetically comprehensive

| | | | |
|---|---|---|---|
| 1 | iy | 20 | n en nx |
| 2 | ih ix | 21 | ng eng |
| 3 | eh | 22 | v |
| 4 | ae | 23 | f |
| 5 | ax ah ax-ah | 24 | dh |
| 6 | uw ux | 25 | th |
| 7 | uh | 26 | z |
| 8 | ao aa | 27 | s |
| 9 | ey | 28 | zh sh |
| 10 | ay | 29 | jh |
| 11 | oy | 30 | ch |
| 12 | aw | 31 | b |
| 13 | ow | 32 | p |
| 14 | er axr | 33 | d |
| 15 | l el | 34 | dx |
| 16 | r | 35 | t |
| 17 | w | 36 | g |
| 18 | y | 37 | k |
| 19 | m em | 38 | hh hv |
| 39 | bcl pcl dcl tcl gcl kcl q epi pau h# not | | |

Table 5.2: The mapping from 61 to 39 labels prior to scoring.

| Phonetic Class | TIMIT labels |
|---|---|
| Vowels & Semi-vowels (VOW) | aa ae ah ao aw ax axh axr ay eh er ey ih ix iy ow oy uh uw ux el l r w y |
| Nasals & Flaps (NAS) | em en eng m n ng nx dx |
| Stops (STP) | b d g p t k |
| Weak Fricatives )WFR) | v f dh th hh hv |
| Strong Fricatives (SFR) | s z sh zh ch jh |
| Closure (CL) | bcl dcl gcl pcl tcl kcl epi pau h# |

Table 5.3: A list of the phonetic classes used in subsequent experiments.

and compact. Each of the 630 speakers read 5 sentences.

- The SI category consists of 1890 sentences that are selected from existing text sources and are phonetically diverse. Each of the 630 speakers read 3 sentences.

The data sets used in the classification experiments are described below and their

| Set | # Speakers | # Utterances | # Hours |
|---|---|---|---|
| Train | 462 | 3696 | 3.14 |
| Development | 50 | 400 | 0.34 |
| Core Test | 24 | 192 | 0.16 |
| Full Test | 118 | 944 | 0.81 |

Table 5.4: Number of speakers, utterances, and hours for each of the Train, Development, Core Test, and Full Test data sets.

| Dialect | Speakers |
|---|---|
| New England | mdab0 mwbt0 felc0 |
| Northern | mtas1 mwew0 fpas0 |
| South Midland | mlll0 mtls0 fjlm0 |
| Southern | mbpm0 mklt0 fnlp0 |
| New York City | mcmj0 mjdh0 fmgd0 |
| Western | mgrt0 mnjm0 fdhc0 |
| Army Brat (moved around) | mjln0 mpam0 fmld0 |

Table 5.5: The 24 speakers included in the TIMIT Core Test set.

contents are displayed in Table 5.4.

- The Train set consists of 462 speakers. It is used for training in all the experiments.

- The Development set consists of 50 speakers. It is heavily used, for classification as well as confidence scoring, in the subsequent experiments.

- The Core Test set includes 2 males and 1 female from each dialect group. Table 5.5 shows the 24 selected speakers and their corresponding dialect region.

- The Full Test set consists of 118 speakers. It is primarily used for evaluating confidence scores.

There is no overlap of speakers between any of the data sets, and the sentences in the training set are different from those in the development and test sets.

### 5.1.2 The Classifier

The classification experiments are performed using the SUMMIT segment-based speech recognizer [19]. This recognizer processes the temporal sequence of acoustic measurements (MFCC) to generate a segmentation graph. Each segment in the graph is represented by a fixed-size feature vector. In this thesis we are not concerned with recognition nor the generation of such a segmentation network. Since we only deal with phonetic classification, we obtain the segments from the phonetic transcriptions. We work only with segmental acoustic models and extract acoustic measurements over the given segments.

In all the experiments, normalization and principal component analysis (PCA) are performed on the acoustic observations in order to whiten the feature space. The measurements are then modeled using diagonal Gaussian mixture models (GMMs). $K$-means and Expectation-Maximization (EM) algorithms are used to initialize and estimate the parameters of the Gaussian models respectively. Classification is implemented using Maximum A Posteriori (MAP) incorporating the priors in the acoustic models.

### 5.1.3 The Baseline Classifier

A baseline classifier is set up to serve as a reference in the analysis of the results. The speech waveform is preemphasized by a factor of 0.97 prior to any processing. Next a Hamming window is applied to obtain speech frames and the 256-point STFT is computed for the 25.6 ms frames at a rate of 5 ms. 14 MFCCs are computed using the method described in Appendix A. A 76-dimensional observation vector is extracted for each segment in the TIMIT phonetic transcriptions. Similarly to the acoustic observation described in Section 4.1, the segmental measurements consisted of:

- 3 MFCC averages computed over the segment in a 3-4-3 proportion
- 2 MFCC derivatives computed using linear least-squared error regression over a time frame of 40 ms centered at the start and end of the segment

- 3 average energies computed over the segment in a 3-4-3 proportion
- 2 derivative energies computed similarly to the MFCC derivatives
- a log duration

Diagonal GMMs are used to model the acoustic measurements with a minimum of 61 datapoints per mixture component and a maximum of 96 mixture models per phone. The same is done in the case of the wavelet-based coefficients. With this baseline configuration, we obtain a classification error of approximately 23.9% on the Development set, 24.6% on the Core Test set, and 24.4% on the Full Test set.

## 5.2 Results

The first experiments involved off-the-shelf wavelets of the Daubechies family, where the filter banks are implemented in a tree-structured fashion. Several tree-structures are tested (See Appendix B and Section 4.2 for details). Figure 5-1 shows the classification error rate on the Development set for four Daubechies wavelets as a function of the adopted frequency partitioning. From the plot, we deduce the following:

- The higher the order of the Daubechies wavelet, the better the performance of the classifier. Figure 4-2 illustrates the low-pass filters corresponding to each of the wavelets. We recall that the higher the wavelet order, the more frequency selective the filter is and the lower its attenuation in the stop band. Hence the performance of the classifier is dependent on the filter choice.
- For each filter, there is no significant change in the error rate as a function of the frequency partitioning. We recall that the dimensionality, $N$, of the extracted acoustic observation is dictated by the number of frequency partitions. Furthermore, as seen in Section 4.1 the dimensionality of the final feature vector is $5N + 6$. Subsequent principal component analysis reduces the dimensionality of the resulting feature space to 76 in this case. We suggest that this particular scheme does not take full account of the variability induced by implementing

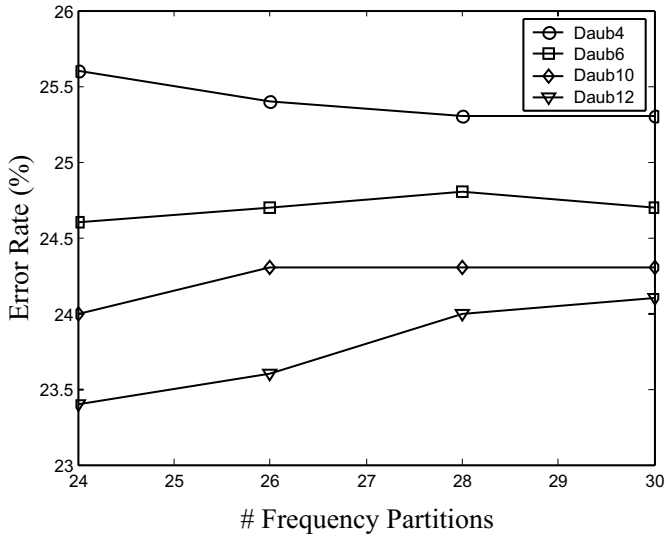


Figure 5-1: Variation of the classification error rate on the Development set as a function of frequency partitions. The acoustic measurements are extracted using the Daubechies wavelets.

| Acoustic Measurement | (%)Error rate on the dev set |
|---|---|
| Haar | 30.3 |
| Daub2 | 27.1 |

Table 5.6: Error rates on the Development set for the Haar and Daub2 wavelets.

one structure as opposed to the other. Possibly, additional processing to the feature vector or the acoustic observation itself would reflect these variations better. We adopt the 26-band structure in the following experiments since it most closely mimics Mel-spaced filters taking into account the critical band effect.

Table 5.6 lists the error rate on the Development set for the Haar and Daub2 wavelets. The results are quite good given that the filters are short — Haar is a 2-tap FIR while Daub2 is a 4-tap FIR —, and consequently they have a very bad selectivity in the frequency domain. Also, as we switch to the Daubechies wavelet, the improvement in performance is noticeable.

Using the first filter design technique described in Section 4.3.1, we obtain and test the two filters listed in Table 4.4. Table 5.7 shows the error rate on the Development

| Acoustic Measurement | (%)Error rate on the dev set |
|---|---|
| Match_Butterworth | 24.1 |
| Match_Ideal | 23.5 |

Table 5.7: Error rates on the Development set for the two filters designed to match the Butterworth and ideal filters respectively.

set for both filters with a clear improvement when the filter is designed to match an ideal filter with a sharp cutoff.

The second filter design technique described in Section 4.3.2, is used to generate six different filters of various lengths. The filters are listed in Table 4.5. Figure 5-2 shows the error rates on the Development set for the six filters. We observe the following:

- Again, the notion of frequency selectivity comes up where the longer the FIR filter, the better its selectivity and the better the performance of the classifier.
- As we saw in Figure 4-5, the filters designed using the attenuation minimization technique match the ideal filter fairly well and have a much better attenuation in the stopband than the Daub12 filter. This explains why the designed filters have the best results seen so far.

Finally, rational filter banks with sampling factors of the form $M/(M-1)$ are implemented and tested. Table 4.6 lists the four different filter banks that are analysed.

As we have previously mentioned, principal component analysis is used to project the feature space onto a lower dimension which is set to 76 for most of the experiments. For the rational filter banks, we experimented with this dimensionality and Table 5.8 gives a listing of the tested values. Figure 5-3 shows the error rate on the Development set for the four rational filter banks as a function of the feature space dimensionality. We make the following observations:

- Filter_8/7 outperforms the rest of the tested filters. We recall that the 8/7 ratio matches the $Q$-factor of Mel-spaced filters, and we suggest that this could be the reason for the better performance.

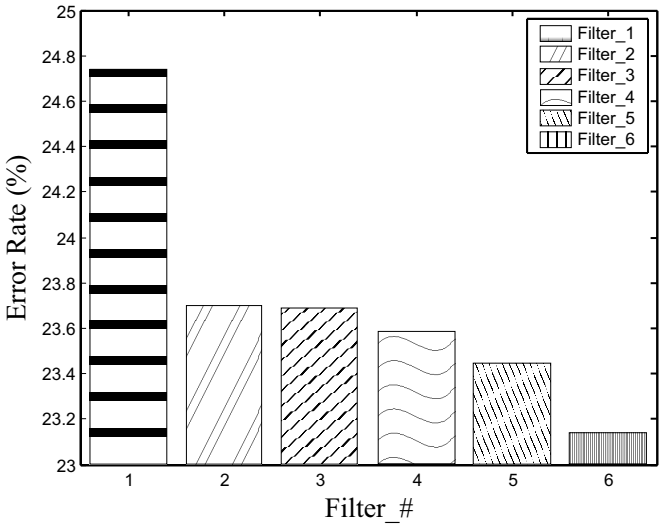


Figure 5-2: Error rates on the Development set for the six filters designed using the attenuation minimization technique.

| Dimension |
|---|
| 70 |
| 76 |
| 80 |
| 90 |

Table 5.8: Listing of the implemented feature space dimensionality.

- The dimensionality value 76 seems to result roughly in the best performance although there is not much change in error rates for each of the filters.

For further evaluation of the results we selected the five acoustic measurements described in Table 5.9.

For a qualitative comparison of the selected acoustic measurements, we include Figure 5-4 which illustrates the low-pass filters corresponding to $A_1$-$A_4$. The ideal low-pass filter with normalized cutoff frequency 0.5 is also included for reference. The low-pass filter corresponding to $A_5$ is not included here, but is illustrated in Figure 4-8, since it corresponds to the rational filter bank of sampling factor 8/7 and covers a different and narrower frequency band.

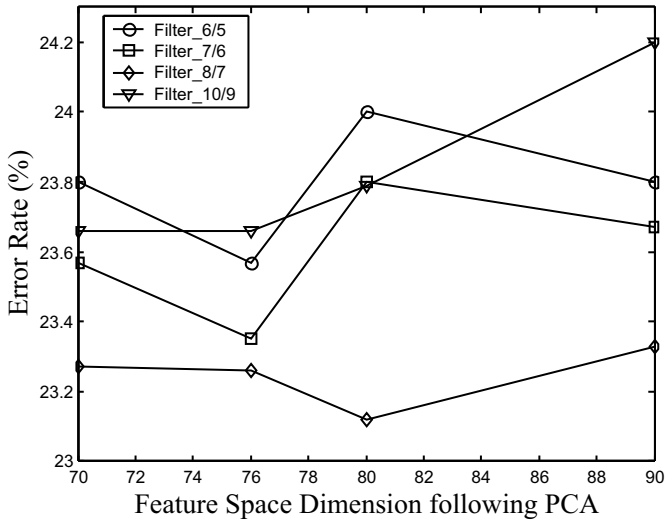


Figure 5-3: The classification error rate on the Development set for the rational filter banks as a function of the feature space dimension.

| Label | Acoustic Measurement Description | | |
|---|---|---|---|
| | Wavelet/Filter | Feature Space Dimension | # Frequency Partitions |
| $A_1$ | Daub12 | 76 | 26 |
| $A_2$ | Match_Ideal | 76 | 26 |
| $A_3$ | Filter_5 | 76 | 26 |
| $A_4$ | Filter_6 | 76 | 26 |
| $A_5$ | Filter_8/7 | 76 | 22 |

Table 5.9: Listing of the acoustic measurements $A_1$-$A_5$ with a brief description of each.

Classification results on the phonetic subclasses are listed in Table 5.10. The results for the baseline classifier are included for comparison. The performance is again evaluated on the Development set. We make the following observations:

- The classification error rates corresponding to all the acoustic measurements match or exceed that of the MFCC on the Development set.

- The results listed in Table 5.10 are reminiscent of those obtained by Halberstadt [23]. Although the overall error rates corresponding to the different acoustic measurements are close to each other, there is an obvious difference in performance over the phonetic subclasses. For example, Filter_6 gives the best

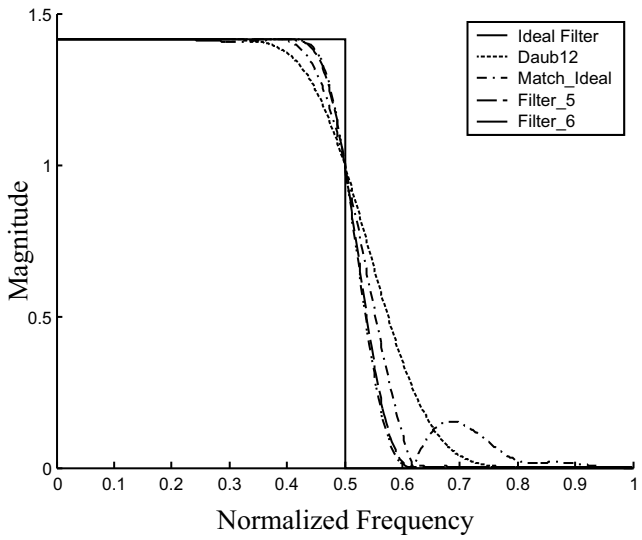


Figure 5-4: The low-pass filters corresponding to the acoustic measurements $A_1$-$A_4$. The ideal low-pass filter is included for reference.

| Acoustic Measurement | (%) Error rate on the dev set | | | | | | |
|---|---|---|---|---|---|---|---|
| | ALL | VOW | NAS | STP | WFR | SFR | CL |
| MFCC | 23.9 | 31.6 | 25.3 | 27.4 | 28.5 | 21.5 | 4.2 |
| $A_1$ | 23.6 | 30.4 | 26.5 | 28.9 | 28.1 | 21.2 | 4.2 |
| $A_2$ | 23.4 | 31.5 | 26.5 | 28.9 | 25.4 | 23.8 | 3.7 |
| $A_3$ | 23.4 | 30.7 | 23.5 | 28.7 | 26.9 | 21.2 | 4.3 |
| $A_4$ | 23.1 | 30.4 | 23.4 | 28.7 | 27.6 | 21.0 | 3.6 |
| $A_5$ | 23.2 | 30.5 | 25.5 | 26.4 | 27.7 | 22.7 | 3.3 |

Table 5.10: Classification performance (overall and phonetic subclasses) of the acoustic measurements described in Table 5.9 and the baseline (MFCC) on the Development set.

results for strong fricatives and nasals while Filter_8/7 gives the best results for stops and closures. This suggests the possibility of implementing a hierarchical architecture where filters optimized to the different subclasses are designed.

The five acoustic measurements are also evaluated on the Core Test and Full Test sets. Significance scoring is performed on the Development and the Full Test sets and not on the Core Test set since it has a small size. The McNemar significance test is used [17]. We make the following observations based on Table 5.11:

- $A_1$ which corresponds to Daub12 does not perform well on the Core Test and

| Acoustic Measurement | (%) Error rate | | McNemar significance | |
|---|---|---|---|---|
| | Core Test | Full Test | Dev | Full Test |
| MFCC | 24.6 | 24.4 | - | - |
| $A_1$ | 25.9 | 24.7 | 0.39 (N) | - |
| $A_2$ | 25.2 | 24.5 | 0.15 (N) | - |
| $A_3$ | 24.9 | 24.3 | 0.15 (N) | 0.56 (N) |
| $A_4$ | 24.6 | 24.1 | 0.019 (Y) | 0.137 (N) |
| $A_5$ | 24.0 | 23.8 | 0.045 (Y) | 0.011 (Y) |

Table 5.11: Classification performance of the acoustic measurements described in Table 5.9 and the baseline (MFCC) on the Core Test and Full Test sets. McNemar significance scores for the Development and Full Tests are also listed. (Y) or (N) indicates whether the difference in results is statistically significant at the 0.05 level.

Full Test sets in comparison to the other acoustic measurements as well as the MFCCs.

- The improvement of $A_1$-$A_3$ over the baseline on the development set is not significant.

- Acoustic measurement $A_5$, which corresponds to the rational filter bank of sampling factor 8/7, consistently outperforms the rest of the measurements and the baseline. The difference in results over the baseline is also significant.

## 5.3 Evaluation of the Results

Our results compare favorably to those mentioned in the literature as well as those of the baseline classifier. Though the results mentioned here are for context-independent phonetic classification, they should not be used entirely for direct comparison since the training and test conditions might differ from one another.

The best reported result for context-independent phonetic classification is by Halberstadt [23]. He successfully experimented with heterogeneous measurements and multiple classifiers, and obtained an error rate of 18.3% on the Core Test set. One of the issues addressed is that of aggregating several acoustic models in order to boost the performance and robustness of the models [25]. To get an idea of the extent

of expected improvement upon implementing aggregation on our acoustic measurements, 4-fold aggregation is tested on $A_5$. Error rates of 21.8% on the Development set, and 22.9% on the Core Test set are obtained. These results compare very well with the performance of the 4-fold aggregated models tested for various segmental measurements in [23]. The error rates Halberstadt reported on the Development set ranged between 21.4% and 22.7%.

Other results are provided by Clarkson and Moreno who implemented Support Vector Machines (SVM), with various kernel functions, applied to phonetic classification. They obtained error rates that range between 22.9% and 23.7% on the Core Test set [9]. Chigier et al. experimented with several signal representations and reported 22.0% using PLP features and a neural net classifier [36]. Chengalvarayan and Deng developed a new hidden Markov model that integrates generalized dynamic feature parameters into the model structure. The best result they reported is an error rate of 31.8% on a 20-speaker test set [8]. Zahorian et al. obtained 23.0% on the Core Test set using spectral/temporal features and binary-pair partitioned neural network classifier [44].

The results we obtain are generated with relatively simple acoustic measurements and classifier. However, they are comparable to those reported in the literature. This is encouraging and portends further improvement upon the combination of several generated measurements in addition to acoustic model aggregation.

# Chapter 6

# Conclusion

In this thesis we addressed the problem of feature extraction for context-independent phonetic classification. We presented a wavelet and filter bank framework in which we exploited various dimensions of the wavelet and filter bank theory such as filter design and the extension of the 2-channel dyadic case to the rational case. We now provide a brief summary of the thesis as well as possible future extensions to the current framework.

## 6.1 Summary

Seeking an alternative to the STFT and its limitations in the task of feature extraction for speech recognition, we investigated wavelet and filter bank theory. The wavelet transform allows a multiresolution analysis and subsequently a good signal representation by exploiting the time/frequency resolution trade-off. Furthermore, the wavelet transform can be efficiently implemented using an adequately designed filter bank. Taking advantage of these key points, we proposed a signal analysis framework.

The basic building block of our framework was *filter design*, where we presented two different techniques for designing orthonormal filter banks based on paraunitary matrix factorization. This came as an alternative to the use of off-the-shelf wavelets that do not always result in filters suitable for speech analysis. The first technique

| Acoustic Measurement | (%) Error rate | | |
|---|---|---|---|
| | Development | Core Test | Full Test |
| MFCC | 23.9 | 24.6 | 24.4 |
| $A_1$ | 23.6 | 25.9 | 24.7 |
| $A_2$ | 23.4 | 25.2 | 24.5 |
| $A_3$ | 23.4 | 24.9 | 24.3 |
| $A_4$ | 23.1 | 24.6 | 24.1 |
| $A_5$ | 23.2 | 24.0 | 23.8 |

Table 6.1: Summary of the classification performance of the acoustic measurements described in Table 4-1 and the baseline (MFCC) on the Development, Core Test, and Full Test sets

minimized the modulus between the designed and desired filter imposing orthogonality through the lattice structure of the designed filter. Another constraint in this method was the number of wavelet vanishing moments which ensured some degree of regularity, a desired wavelet feature. The second method minimized the attenuation in the stopband of the designed filter. The constraints are again orthonormality of the filter bank as well as a regularity of order 1.

A filter bank generates an array of filters that cover different bands of the spectrum. In the simplest case, a dyadic filter bank iterated on the low-pass channel will generate an octave band. We extended this idea in two ways. First, using tree-structured filter banks, we obtained various frequency partitions. With proper tree configuration, we obtained a frequency decomposition that mimics Mel-spacing. The second method involved the design of rational filter banks, which incorporated the critical band effect more naturally. Instead of being restricted to a sampling factor of 2, we are able to generate more general factors of the form $M/(M-1)$.

The final stage is to implement the filter bank and generate the acoustic measurement. Energy is computed over the generated bands and log-scaled. The resulting acoustic measurement is further processed to generate a large feature vector consisting of concatenated averages and derivatives of the original observation as well as energies and durations. The high dimensionality of the resulting feature vector is reduced using principal component analysis.

Table 6.1 summarizes the performance of sample acoustic measurements on the Development, Core Test, and Full Test sets. We observe a gradual improvement in the results as we progress from the first to the last measurement. $A_1$ corresponds to an off-the-shelf wavelet, $A_2$ corresponds to a designed filter using the matching technique, $A_3$ and $A_4$ are generated using the attenuation minimization technique, and $A_5$ corresponds to a designed orthonormal filter bank with sampling factor 8/7. The results compare favorably to those reported in the literature as well as the baseline classifier. We note again that, most of the research, specifically on wavelet-based analysis for speech recognition, reported in the literature involves only off-the-shelf wavelets and dyadic filter bank implementations. Unfortunately, we are unable to perform direct comparison with these studies, since the data sets were often different and smaller than ours. However, we are able to show that off-the-shelf wavelets do not necessarily give the best results, and there is a need for wavelet and, consequently, filter design. We also showed that a dyadic filter bank implementation is not optimal, and we adopted a method for the design of rational filter banks with sampling factor $M/(M-1)$. These structures naturally incorporate the critical band effect while providing a fine resolution of the spectrum. The results shown in Table 6.1 indicate that a rational filter bank consistently outperforms the rest of the acoustic measurements as well as the baseline classifier.

## 6.2 Future Work

The framework that we have presented gives insight into the effect of filter design and rational filter bank on the performance of phonetic classification. It is however, still primitive in terms of design as well as implementation. We cite some of the future challenges in its improvement:

The wavelet and filter bank framework was tested on the TIMIT corpus which is a clean data set. It would be even more challenging to implement it on a noisy data set. With the multi-scale analysis provided by wavelets, and the ability to isolate noisy components — under the assumption that they would be localized in scale —,

it is encouraging to attempt such an implementation.

The framework is also limited to the task of phonetic classification. A natural extension would be phonetic recognition taking into account linguistic context-dependency such as coarticulation. Obviously, the final aim would be to develop a full speech recognition system.

With the flexibility in filter design and frequency partitioning, the framework lends itself to hierarchical approaches where it will be possible to adaptively design filter banks that are optimized for the classification or recognition of a class or subclass of signals. Halberstadt investigated the use heterogeneous measurements and acoustic model aggregation to improve model performance [23, 25]. We are currently implementing acoustic model aggregation where our latest error rates are 21.8% on the Development set and 22.9% on the Core Test set. The results are obtained for a 4-fold aggregated model corresponding to acoustic measurement $A_5$, which is presented in this thesis. It is comparable to the error rates reported by Halberstadt for the 4-fold aggregated models corresponding to his segmental measurements [23]. Table 6.2 shows the results reported by Halberstadt in [23] on the Development set as well as those corresponding to $A_5$.This is a very promising result and we hope that the combination of various aggregated models will lead to even better performance.

The filter and filter bank design techniques that we used in this thesis are quite simple and do not always give satisfactory results or even converge. For example, the filter matching method did not give a very good attenuation in the stopband nor did it always give a good match to a wide range of filters. It would be interesting to modify the method, devise a new one, or implement different optimization techniques in order to experiment with various filter shapes.

As far as filter bank implementation, we have not mentioned the complexity of the algorithm in terms of computation and consequently processing time. We discuss it briefly here giving another example of the extensions that can be done to the framework. For example, ideally we would like our system to operate as close as possible to real-time. However, this is currently not the case. Our core *wavelet transform* algorithm, has a polyphase implementation (See Figure 2-3 and Equation 2.1 for the

| Acoustic Measurement | (%) Error rate | | | | | | |
|---|---|---|---|---|---|---|---|
| | ALL | VOW | NAS | SFR | WFR | STP | CL |
| S1 | 21.52 | 28.4 | 22.4 | 19.8 | 26.9 | 23.7 | 3.8 |
| S2 | 21.60 | 28.9 | 22.2 | 18.2 | 28.4 | 23.7 | 3.4 |
| S3 | 21.46 | 27.9 | 20.4 | 18.9 | 29.0 | 26.5 | 3.8 |
| S4 | 22.47 | 27.9 | 20.7 | 19.5 | 30.1 | 24.8 | 4.0 |
| S5 | 22.10 | 28.8 | 23.1 | 20.2 | 30.0 | 24.9 | 3.6 |
| S6 | 22.64 | 28.4 | 24.6 | 20.8 | 32.5 | 26.6 | 4.4 |
| S7 | 22.68 | 29.6 | 25.7 | 20.4 | 30.3 | 25.2 | 3.3 |
| S8 | 22.08 | 28.3 | 25.8 | 19.4 | 30.3 | 24.6 | 3.9 |
| $A_5$ | 21.79 | 28.6 | 22.7 | 20.4 | 24.2 | 26.7 | 3.6 |

Table 6.2: Classification performance (overall and phonetic subclasses) of the acoustic measurements designed in [23] and one of the acoustic measurements proposed in this thesis, $A_5$. 4-fold aggregation is performed on all models and classification is done on the Development set.

dyadic case). If we look at the 2-channel case, the computation of the output requires four convolutions of the polyphase inputs with the polyphase filters of length $L/2$ — the original filters were of length $L$. Thus the number of operations/input sample is $L$ multiplications and $L-1$ additions. On the other hand, an FFT-based convolution would require much less operations/input sample. For example, the split-radix FFT algorithm will take $\alpha.L.\log_2 L + O(\log\log L)$ [42]. Since we are dealing with large input signals (frames of 320 samples) and large FIR filters of lengths more than 100 taps, it becomes advantageous to consider optimized FFT-based algorithms for filter bank implementation.

More importantly, the proposed feature extraction framework can be pipelined. Since the underlying algorithm is frame-based, it is possible to implement it in real-time where frames are processed independently while the signal is input to the system.

Wavelets and filter banks are, indeed, compelling tools to pursue further in the field of speech analysis.

# Appendix A

# MFCC Computation

In this appendix, we describe the stages involved in the MFCC computation algorithm [13, 23]. Figure A-2 illustrates the different stages in the MFCC computation:

**Stage 1.** Compute the short-time energy spectrum by calculating the magnitude squared of the STFT over frame intervals of predefined width, for example 25 ms.

**Stage 2.** Multiply the energy spectrum by $N$ triangular band-pass filters. As illustrated in Figure A-1, 40 is a typical value for $N$, where 40 triangular filters are enough to cover the whole spectrum when the speech signal has been sampled at a rate of 16 kHz. Compute the Mel-frequency spectral coefficients (MFSC), as the energy outputs of each filter. The triangular filters are designed to incorporate a Mel-frequency warping with linear spacing below 1000 Hz and logarithmic spacing above that.

$$f' = 2595 \log_{10}(1 + \frac{f}{700})$$

**Stage 3.** Compute the log transform, $10 \log_{10}()$ of the $N$ MFSCs

**Stage 4.** Take the DCT of the logged MFSCs to whiten the MFSC space and project

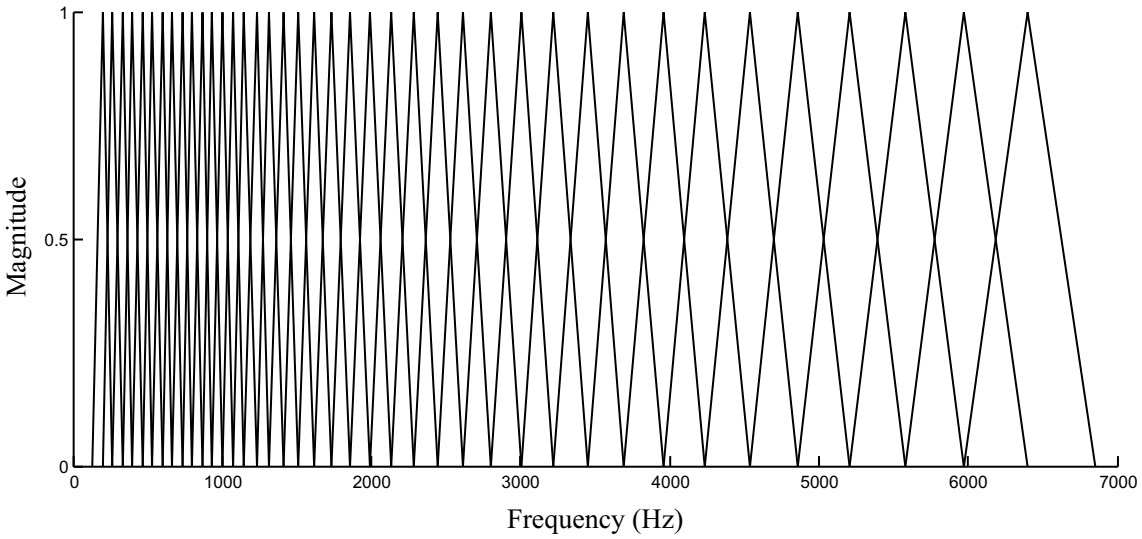


Figure A-1: 40 triangular filters used for the MFSC computation.

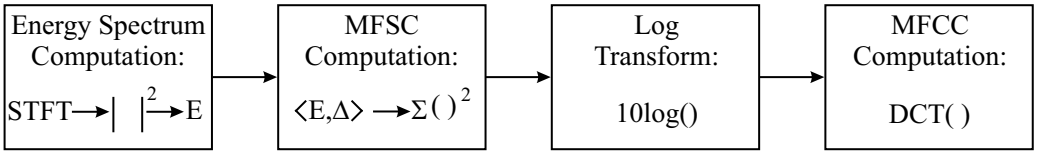


Figure A-2: Flowchart depicting the computation of the MFCCs.

it onto a lower dimension $M$, typically 14 when $N$ is 40.

$$MFCC[i] = \sum_{k=0}^{K-1} \cos \frac{\pi i(k-1/2)}{K} MFSC_{log}[k] \quad m = 0 \ldots M, K = N$$

# Appendix B

# Description of the Frequency Partitions

In this appendix, we describe the frequency partitions that are implemented using tree-structured filter banks. Tables B, B, and B list in detail the lower and upper cutoff frequencies as well as the bandwidths of the spectral bands obtained for three different tree-structured filter banks.

| Filter # | Lower cutoff frequency (Hz) | Upper cutoff frequency (Hz) | Bandwidth (Hz) |
|---|---|---|---|
| 1 | 0 | 125 | 125 |
| 2 | 125 | 250 | 125 |
| 3 | 250 | 375 | 125 |
| 4 | 375 | 500 | 125 |
| 5 | 500 | 625 | 125 |
| 6 | 625 | 750 | 125 |
| 7 | 750 | 875 | 125 |
| 8 | 875 | 1000 | 125 |
| 9 | 1000 | 1250 | 250 |
| 10 | 1250 | 1500 | 250 |
| 11 | 1500 | 1750 | 250 |
| 12 | 1750 | 2000 | 250 |
| 13 | 2000 | 2250 | 250 |
| 14 | 2250 | 2500 | 250 |
| 15 | 2500 | 2750 | 250 |
| 16 | 2750 | 3000 | 250 |
| 17 | 3000 | 3500 | 500 |
| 18 | 3500 | 4000 | 500 |
| 19 | 4000 | 4500 | 500 |
| 20 | 4500 | 5000 | 250 |
| 21 | 5000 | 5500 | 500 |
| 22 | 5500 | 6000 | 500 |
| 23 | 6000 | 7000 | 1000 |
| 24 | 7000 | 8000 | 1000 |

Table B.1: The frequency bands of the 24 filters obtained with tree-structured filter banks.

| Filter # | Lower cutoff frequency (Hz) | Upper cutoff frequency (Hz) | Bandwidth (Hz) |
|---|---|---|---|
| 1 | 0 | 125 | 125 |
| 2 | 125 | 250 | 125 |
| 3 | 250 | 375 | 125 |
| 4 | 375 | 500 | 125 |
| 5 | 500 | 625 | 125 |
| 6 | 625 | 750 | 125 |
| 7 | 750 | 875 | 125 |
| 8 | 875 | 1000 | 125 |
| 9 | 1000 | 1250 | 250 |
| 10 | 1250 | 1500 | 250 |
| 11 | 1500 | 1750 | 250 |
| 12 | 1750 | 2000 | 250 |
| 13 | 2000 | 2250 | 250 |
| 14 | 2250 | 2500 | 250 |
| 15 | 2500 | 2750 | 250 |
| 16 | 2750 | 3000 | 250 |
| 17 | 3000 | 3250 | 250 |
| 18 | 3250 | 3500 | 250 |
| 19 | 3500 | 3750 | 250 |
| 20 | 3750 | 4000 | 250 |
| 21 | 4000 | 4250 | 250 |
| 22 | 4250 | 4500 | 250 |
| 23 | 4500 | 4750 | 250 |
| 24 | 4750 | 5000 | 250 |
| 25 | 5000 | 5500 | 500 |
| 26 | 5500 | 6000 | 500 |
| 27 | 6000 | 7000 | 1000 |
| 28 | 7000 | 8000 | 1000 |

Table B.2: The frequency bands of the 28 filters obtained with tree-structured filter banks.

| Filter # | Lower cutoff frequency (Hz) | Upper cutoff frequency (Hz) | Bandwidth (Hz) |
|---|---|---|---|
| 1 | 0 | 125 | 125 |
| 2 | 125 | 250 | 125 |
| 3 | 250 | 375 | 125 |
| 4 | 375 | 500 | 125 |
| 5 | 500 | 625 | 125 |
| 6 | 625 | 750 | 125 |
| 7 | 750 | 875 | 125 |
| 8 | 875 | 1000 | 125 |
| 9 | 1000 | 1250 | 250 |
| 10 | 1250 | 1500 | 250 |
| 11 | 1500 | 1750 | 250 |
| 12 | 1750 | 2000 | 250 |
| 13 | 2000 | 2250 | 250 |
| 14 | 2250 | 2500 | 250 |
| 15 | 2500 | 2750 | 250 |
| 16 | 2750 | 3000 | 250 |
| 17 | 3000 | 3250 | 250 |
| 18 | 3250 | 3500 | 250 |
| 19 | 3500 | 3750 | 250 |
| 20 | 3750 | 4000 | 250 |
| 21 | 4000 | 4250 | 250 |
| 22 | 4250 | 4500 | 250 |
| 23 | 4500 | 4750 | 250 |
| 24 | 4750 | 5000 | 250 |
| 25 | 5000 | 5250 | 250 |
| 26 | 5250 | 5500 | 250 |
| 27 | 5500 | 5750 | 250 |
| 28 | 5750 | 6000 | 250 |
| 29 | 6000 | 7000 | 1000 |
| 30 | 8000 | 8000 | 1000 |

Table B.3: The frequency bands of the 30 filters obtained with tree-structured filter banks.

# Bibliography

[1] A.M.A. Ali, J. Van der Spiegel, and P. Mueller. An acoustic-phonetic feature-based system for the automatic recognition of fricative consonants. In *Proc. ICASSP '98*, volume 2, pages 961–964, Seattle, WA USA, May 1998.

[2] A.M.A. Ali, J. Van der Spiegel, and P. Mueller. Acoustic-phonetic features for the automatic classification of stop consonants. *IEEE Transactions on Signal Processing*, 9(8):833–841, November 2001.

[3] K. Amaratunga. Wavelet representations and their application to the modeling, compression and reduction of spatial data. In *Proc. of the 1st Intl. Conf. on New Information Technologies for Decision Making in Civil Engineering*, pages 81–92, Montreal, Canada, October 1998.

[4] N. Bitar and C. Espy-Wilson. A knowledge-based signal representation for speech recognition. In *Proc. ICASSP '96*, pages 29–32, Atlanta, GA USA, May 1996.

[5] T. Blu. Iterated filter banks with rational rate changes connection with discrete wavelet transforms. *IEEE Transactions on Signal Processing*, 41(12):3232– 3244, December 1993.

[6] T. Blu. *Bancs de filtres iteres en fraction d'octave, Application au codage de son.* PhD thesis, Ecole National Superieur des Telecommunications, Paris, France, april 1996. in French.

[7] T. Blu. A new design algorithm for two-band orthonormal rational filter banks and orthonormal rational wavelets. *IEEE Transactions on Signal Processing*, 46(6):1494 – 1504, June 1998.

[8] R. Chengalvaryan and L. Deng. Use of generalized dynamic feature parameters for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 5(3):232–242, May 1997.

[9] P. Clarkson and P.J. Moreno. On the use of support vector machines for phonetic classification. In *Proc. ICASSP '99*, volume 2, pages 585–588, Phoenix, AZ USA, March 1999.

[10] I. Cohen, S. Raz, and D. Malah. Shift invariant wavelet packet bases. In *Proc. ICASSP '95*, volume 2, pages 1081–1084, Detroit, MI USA, May 1995.

[11] A. Croisier, D. Esteban, and C. Galand. Perfect channel splitting by use of interpolation/-decimation/tree decomposition techniques. In *Intl. Conf. on Information Sciences and Systems*, pages 443–446, Patras, Greece, August 1976.

[12] I. Daubechies. *Ten Lectures on Wavelets*, volume 61. SIAM Press, Philadelphia, PA USA, 1992.

[13] S. B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. Acoustics, Speech and Signal Processing*, ASSP-28(4):357, 1980.

[14] O. Farooq and S. Datta. Mel filter-like admissible wavelet packet structure for speech recognition. In *IEEE-SP Letters*, volume 8, pages 196–198, July 2001.

[15] O. Farooq and S. Datta. Robust features for speech recognition based on admissible wavelet packets. In *Electronics Letters*, volume 37, pages 1554–1556, December 2001.

[16] R.F. Favero. Compound wavelets: wavelets for speech recognition. In *Proc. of the IEEE-SP Intl. Symp. on Time-Frequency and Time-Scale Analysis*, pages 600–603, Philadelphia, PA USA, October 1994.

[17] L. Gillick and S.J. Cox. Some statistical issues in the comparison of speech recognition algorithms. In *Proc. ICASSP '89*, volume 1, pages 532–535, Glasgow, UK, May 1989.

[18] J. Glass. A probabilistic framework for segment-based speech recognition. *Computer Speech and Language*, pages 137–152, 2003.

[19] J. Glass, J. Chang, and M. McCandless. A probabilistic framework for feature-based speech recognition. In *Proc. ICSLP '96*, volume 4, pages 2277–2280, Philadelphia, PA USA, October 1996.

[20] H. Goldstein. Formant tracking using the wavelet-based dst. In *Proc. of the IEEE South African Symposium on Communications and Signal Processing*, pages 183–189, Stellenbosch, South Africa, October 1994.

[21] P. Goupillaud, A. Grossman, and J. Morlet. Cycle-octave and related transforms in seismic signal analysis. *Geoexploration*, pages 85–102, 1984-1985.

[22] M. Gupta and A. Gilbert. Robust speech recognition using wavelet coefficient features. In *IEEE Automatic Speech Recognition and Understanding Workshop*, pages 445–448, December 2001.

[23] A.K. Halberstadt. *Heterogeneous Acoustic Measurements and Multiple Classifiers for Speech Recognition.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, November 1998.

[24] Y. Hao and X. Zhu. A new feature in speech recognition based on wavelet transform. In *Proc. of the 5th Intl. Conf. on Signal Processing*, volume 3, pages 1526–1529, Beijing China, August 2000.

[25] T. J. Hazen and A. K. Halberstadt. Using aggregation to improve the performance of mixture Gaussian acoustic models. In *Proc. ICASSP '98*, pages 653–656, Seattle, WA USA, May 1998.

[26] H. Hermansky, B. A. Hanson, and H. Wakita. Perceptually based linear predictive analysis of speech. In *Proc. ICASSP '85*, pages 509–512, Tampa, FL USA, March 1985.

[27] P.Q. Hoang and P.P. Vaidyanathan. Non-uniform multirate filter banks: theory and design. In *IEEE Intl. Symp. on Circuits and Systems*, volume 1, pages 371–374, Portland, OR, May 1989.

[28] A. Juneja and C. Espy-Wilson. An event-based acoustic-phonetic approach to speech segmentation and e-set recognition. In *Proc. ICPhs '03*, Spain, 2003.

[29] N.A. Kader. Formant tracking using the wavelet-based dst. In *Seventeenth National Radio Science Conference*, pages 1–8, Minufiya, Egypt, February 2000.

[30] Y. Kaisheng and Cao Zhigang. A wavelet filter optimization algorithm for speech recognition. In *Intl. Conf. on Communication Technology*, Beijing, China, October 1998.

[31] K. Kim, D.H. Youn, and C. Lee. Evaluation of wavelet filters for speech recognition. In *IEEE International Conference on Systems, Man, and Cybernetics*, volume 4, pages 2891–2894, Nashville, TN USA, October 2000.

[32] J. Kovacevic and M. Vetterli. Perfect reconstruction filter banks with rational sampling factors. *IEEE Transactions on Signal Processing*, 41(6):2047 – 2066, June 1993.

[33] L. Lamel, R. Kassel, and S. Seneff. Speech database development: design and analysis of the acoustic-phonetic corpus. In *Proc. DARPA Speech Recognition Workshop*, pages 100–109. Report No. SAIC-86/1546, February 1986.

[34] L.F. Lamel and J. Gauvian. High-performance speaker-independent phone recognition using cdhmm. In *European Conference Speech Communication and Technology*, pages 121–124, 1993.

[35] K. F. Lee and H. W. Hon. Speaker-independent phone recognition using hidden Markov models. *IEEE Trans. Acoustics, Speech and Signal Processing*, ASSP-37(11):1641, 1989.

[36] H.C. Leung, B. Chigier, and J.R. Glass. A comparative study of signal representations and classification techniques for speech recognition. In *Proc. ICASSP '93*, volume 2, pages 680–683, Minneapolis, MN USA, 1993.

[37] L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition.* Prentice-Hall, Englewood Cliffs, NJ, 1993.

[38] H.L. Rufiner. A method of wavelet selection in phoneme recognition. In *40th Midwest Symp. on Circuits and Systems*, volume 2, pages 889–891, Sacramento,CA USA, August 1997.

[39] G. Strang and T. Nguyen. *Wavelets and Filter Banks.* Wellesley-Cambridge Press, Wellesley, MA, 1996.

[40] B.T. Tan, R. Lang, H. Schroder, Minyue, A. Spray, and P. Dermody. Applying wavelet analysis to speech segmentation and classification. In *Wavelet Applications, Proc. SPIE 2242*, pages 750–761, 1994.

[41] B.T. Tan, F. Minyue, A. Spray, and P. Dermody. The use of wavelet transforms in phoneme recognition. In *Proc. ICSLP '96*, volume 4, pages 2431–2434, Philadelphia, PA USA, October 1996.

[42] M. Vetterli and J. Kovacevic. *Wavelets and Subband coding.* Prentice-Hall, Englewood Cliffs, NJ, 1993.

[43] H. Wassner and G. Chollet. New time-frequency derived cepstral coefficients for automatic speech recognition. In *Proc. ICSLP '96*, volume 4, pages 260–263, Philadelphia, PA USA, October 1996.

[44] S. A. Zahorian, P. L. Silsbee, and X. Wang. Phone classification with segmental features and a binary-pair partitioned neural network classifier. In *Proc. ICASSP '97*, pages 1011–1014, Munich Germany, April 1997.