An Architecture for Low-Power Voice-Command Recognition Systems

by

Qing He

B.A.Sc, University of Waterloo (2009)
S. M, Massachusetts Institute of Technology (2012)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2016

Author .....
Department of Electrical Engineering and Computer Science
May 20, 2016

Certified by .....
Gregory W. Wornell
Sumitomo Professor of Engineering
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Chair, Department Committee on Graduate Theses

An Architecture for Low-Power Voice-Command Recognition Systems

by

Qing He

Submitted to the Department of Electrical Engineering and Computer Science
on May 20, 2016, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

The advancements in fields such as machine-learning have allowed for a growing number of applications seeking to exploit learning methods. Many such applications involve complex algorithms working over high-dimensional features and are implemented in large scale systems where power and other resources are abundant. With emerging interest in embedded applications, nano-scale systems, and mobile devices, which are power and computation constrained, there is a rising need to find simple, low-power solutions for common applications such as voice activation.

This thesis develops an ultra-low-power system architecture for voice-command recognition applications. It optimizes system resources by exploiting compact representations of the signal features and extracting them with efficient analog front-ends. The front-end performs feature pre-selection such that only a subset of all available features are chosen and extracted. Two variations of front-end feature extraction design are developed, for the applications of text-dependent speaker-verification and user-independent command recognition, respectively.

For speaker-verification, the features are selected with knowledge of the speaker's fundamental frequency and are adapted based on the noise spectrum. The back-end algorithm, supporting adaptive feature selection, is a weighted dynamic time warping algorithm that removes signal misalignments and mitigates speech rate variations while preserving the signal envelope.

In the case of user-independent command recognition, a universal set of features are selected without using speaker-specific information. The back-end classifier is enabled by a novel multi-band deep neural network model that processes only the selected features at each decision.

In experiments, the proposed systems achieve improved accuracy with noise robustness using significantly less power consumption and computation than existing systems. Components of the front- and back-ends have been implemented in hardware, and the end-to-end system power consumption is kept under a few hundred μWs.

Thesis Supervisor: Gregory W. Wornell
Title: Sumitomo Professor of Engineering

Acknowledgments

The past few years of training and experience at MIT have transformed me and my thinking. It has been an amazing adventure!

First and foremost, I would like to offer my sincerest gratitude to my advisor Prof. Gregory Wornell for his mentoring, for being open-minded and for supporting me to work on such an interesting problem that allowed me to get exposure to many new fields. Throughout the past few years, I've learned a lot from him, both about work and about life. I always remember, during my preparation for a meeting with my project sponsor, I was not taking it very seriously at first. Then, Greg told me 'Every meeting you drive and every presentation you give, no matter how short it is, it contributes to your reputation.' This really reflects his approach to research. I admire how he always seeks to understand the fundamental truth and deliver the best quality work.

I would also like to thank my committee members Dr. James Glass and Prof. Vivienne Sze. When I started working on this project, I had very little background in the field of speech recognition. Jim has been very patient with me. He discussed possible opportunities for the project with me and pointed me to related references. As a leading researcher in the field of ASIC design, Vivienne has given me a lot of help in designing a practical system. She helped me to identify the key elements for hardware implementation and power measurements. She always gave me detailed feedback and timely responses to my questions. Without my strong thesis committee, whose combined expertise cover the fields of signal processing, inference, speech recognition and multi-media application hardware design, this work would not been possible.

It's my great pleasure to be part of the Signals, Information and Algorithms Lab. I'm so grateful that I had the opportunity to interact with such truly talented, kind, cheerful and passionate colleagues: Gauri Joshi, Ying-zong Huang, Lisa Zhang, Atulya Yellepeddi, Gal Shulkind, Ganesh Ajjanagadde, Mo Deng, Joshua Lee, Da Wang and James Krieger. We've discuss about work, about life, about fun stuff to do and good restaurants to try.

This work was supported in part by Texas Instruments, by NSF under Grant No. CCF-1319828, by Systems on Nanoscale Information fabriCs (SONIC), an SRC STARnet Center sponsored by MARCO and by DARPA.

Through interacting with you guys, I was able to see a more colorful world. I would also like thank our lab assistant Tricia O'Donnell for always helping and taking care of us.

This project stemmed from my internship project at Texas Instruments, Kilby Labs, Santa Clara. Without the help from my manager Dr. Wei Ma, and my mentor Dr. Bozhao Tan, I would not have had the opportunity to work on this problem. They gave me tremendous help throughout the years by giving me feedback on my research and providing me the data I needed for testing. I also would like to thank my colleagues at TI: Dr. Zhenyong Zhang, Lin Sun, Dr. Yunhong Li and Siddharth Joshi who have taught me a lot about hardware design and gave me a lot of mentoring and help on the hardware implementation aspects of the project.

I also owe many thanks to my MIT colleagues outside of my group: Stephen Shum, Jackie Lee, Ekapol Chuangsuwanich from the Spoken Language Systems Group and Michael Price and Frank Yaul from the Energy-Efficient Circuits and Systems group, who have given me beginner level tutorials and technical advices on topics related to speech recognition and hardware.

In addition, I would like to thank my friends around the Bay Area: Yili Huang, Miles Malerba, Mark Rushby, Sandeep Wali, Christos Hristovski, Dali Wu and Fred Dupere, who kept me company and gave me a lot of rides during my three internships at TI. You guys made the dark days much sunnier.

It is such an honor to have met so many dear friends at MIT, who have grown with me, laughed with me, fought and made up with me, listened to me, helped me, and shared many more precious moments in my life with me. I'm so grateful for having met you guys: Henna Huang, Lei Zhang, Ying Yin, Ermin Wei, Mitra Osqui, Shen Shen, Bonnie Lam, Mina Karzand , Di Chen, Yan Chen, Sha Huang, Min Ding, Ying Liu, Dheera Venkatraman and Yinqing Li.

Last but not least, I would like to thank my parents and my grandma for raising me, supporting me, encouraging me and just simply wishing me to be happy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The future of computing

With increasingly capable hardware and low-cost computing resources, current engineering systems are becoming more complex and general purpose. Common functions such as classification and detection usually involve many stages of processing. In these systems, raw data are usually acquired at rates much higher than the Nyquist rate and with rich details in order to minimize information loss at an early stage. This approach of first acquiring all possible information and then performing data reduction afterward offers the benefits of blind signal acquisition and flexible downstream processing. However, this comes at the costs of high-rate data conversion, high-speed processing, and high computation complexity due to the large input signal dimension. All of these factors can contribute to the system power consumption. Therefore, when power and computation complexity are strictly limited, the conventional approach needs to be reconsidered.

One special opportunity lies in the area of application-specific designs. When the end-application is limited to one or a set of well-defined tasks, the entire system architecture can be optimized to directly exploit relevant information, thus achieving faster, more power-efficient, more accurate and more robust performances. In particular, the following techniques are applied to system design:

(1) Low computation complexity through early stage feature pre-selection: when the essential features for the end application reside in a space whose dimension is much lower

than the dimension of the raw data, performing signal dimension reduction at the front of the processing pipeline would potentially reduce the data-conversion rate and the computation complexity for all downstream processors. For example, consider the human visual system, which has evolved to be exceptionally efficient for targeted tasks such as identifying a dangerous situation from complex natural images. Instead of capturing and processing a high resolution image of the entire scene, the visual sensory system ignores details in the information-rich images and only extracts a few key features such as speed, color, size and shape to enable time-critical decision making [1, 2]. Similarly, for speech recognition, the essential speech features lie in the modulated harmonics of a person's voice, which occupy a small portion of the full speech spectrum. This property can be utilized to enable low complexity processing through early stage feature extraction.

(2) Improved noise robustness through feature pre-selection: sometimes, less is more. For tasks such as speech recognition, higher accuracy can be achieved by discarding noisy contents and making decisions based on a smaller set of high quality features [3–5]. Hence, by disregarding noisy information, not only can we save the effort on acquiring and processing corrupted data, but the system recognition accuracy can also be improved.

(3) Adaptive feature extraction and processing: most classification systems acquire and process information (e.g., features) in a fixed manner. When there is a constraint on power consumption, a more efficient approach is to achieve the desirable performance through adaptive processing using minimum efforts. For example, a smaller number of high quality features are needed to yield the same performance as a larger number of low quality features. In other words, system efficiency can be improved by enabling adaptive feature selection and adaptive processing such that the most appropriate feature selection choices can be made in real-time based on factors such as the background noise level.

We propose an architecture that incorporates these design concepts and demonstrate that it achieves fast, accurate, low-power consumption and low-complexity performances. The biggest challenges to building such a system involve identifying the key information bearing features, finding an efficient method to extract the features and back-end algorithms that support adaptive feature processing. The specifics of the system are developed based on considerations for low-power voice command recognition applica-

tions. Nevertheless, this architecture is applicable to a wider range of signals whose features lie in a low-dimensional space compared to the raw data. The design concepts of such an architecture can be applied to a variety of systems where power-consumption and computation complexity limit system design.

1.2 Speech recognition applications

With the increasing popularity of mobile devices and wearable electronics, it is of interest to enable full voice interaction beginning with low-power, voice wake-up. Existing command recognition algorithms achieve excellent accuracy. Nevertheless, they are prohibitively power-consuming for standalone devices such as smartwatches or smartphones. In this thesis, we propose an architecture for voice wake-up systems, which, in addition to being low-power and low-complexity, achieves high recognition accuracy and improved noise robustness compared to existing systems.

More specifically, our task is to design a system that continuously listens to the surrounding environment and recognizes a small set of short commands such as 'hi, galaxy' in order to wake up the host device. With this staged gating approach, the overall system power consumption for the mobile device can be kept low while staying perceptually always-on.

1.3 Research scope

The problem of automatic speech recognition (ASR) has been widely studied and researched for decades. Recent research in speech recognition systems are achieving unprecedented accuracies [6,7]. Even though the goal is simple and clear —to decode speech sounds into text —the problem is quite complex and can be formulated in many different ways for a wide range of applications from speech dictation, speech-to-text-processing, translation, voice control to voice dialing. There are also many variations of the speech recognition problem including keyword recognition, interactive control systems, large vocabulary speech understanding, and speaker-verification, etc. For a comprehensive introduction on ASR history and an in-depth discussion on the details of the technology, one may refer to popular texts such as [8] and [9].

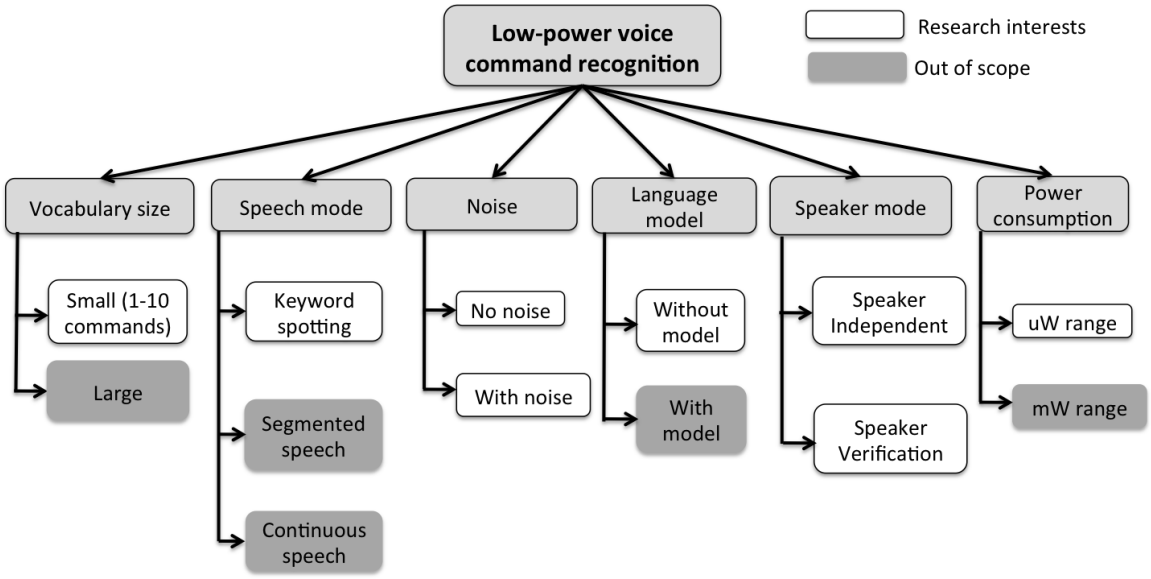


Figure 1-1: Research scope: the unshaded blocks indicate our research focus in the general field of speech recognition

Figure 1-1 outlines where our research lies within the broader field of speech recognition. The unshaded blocks indicate the region of our research interests and the shaded blocks correspond to the topics outside our research scope. Each column corresponds to a different aspect of the system.

General speech recognition systems are designed to recognize arbitrary sentences from a large vocabulary. In contrast, our research falls under the umbrella of small-vocabulary speech recognition, where the goal is to detect a small set of voice commands. The voice commands are short and are expected to be less than one second. In the context of conventional speech recognition technologies, this problem is also known as keyword spotting (KWS) [10] or spoken-term detection [11].

Instead of translating continuous speech to text (e.g., automatic speech dictation) or recognizing phrases from segmented speech that has been preprocessed and trimmed (e.g., automatic menu-selection over phone), our research aims at spotting short commands from unconstrained continuous speech. To enable system design for practical applications, one aspect of our research focus is noise robustness so that the recognition system achieves high accuracy under both quiet and noisy conditions. Most existing speech recognizers take advantage of pre-trained language models, such as the popular

tri-phone model [12]. To build a robust system that accepts speech in any language, we will not constrain our system to any language-specific models.

Our research studies two categories of speaker modes: speaker-independent recognition and text-dependent speaker-verification (SV). Speaker-independent speech recognition systems are trained with thousands of samples from a prescribed list of keywords, uttered by a variety of users, such that the system can then recognize the keywords from arbitrary users, including new users that it has never seen before. On the other hand, the text-dependent SV system does not require pre-training with a large amount of data, even though model pre-training can still be applied as prior knowledge. It takes in a few enrollment samples from the user when the system is first initiated and then recognizes the voice-command and the speaker in a joint manner. This text-dependent SV system allows the users to define their own keywords and enables customizable voice-authenticated wake-up.

There is a wide range of existing systems that offer good solutions to the command recognition problem. What makes them unfit for our application is their large power-consumption and heavy computation. The biggest challenge to our research is to design a system such that its end-to-end power consumption is confined to the μW range. Potential applications of the voice-command recognition system include activation and authentication for cellphones, google-glass, smart watches, etc.

1.4 Prior work on speech recognition hardware

Most speech recognition systems are implemented on general purpose computing hardware. The small number of application specific speech recognition systems are implemented on either field-programmable gate arrays (FPGAs) or application-specific integrated circuits (ASICs) mostly for the application of large vocabulary speech recognition.

Application specific large vocabulary speech recognition hardware offers the benefits of faster processing speed and lower power consumption. Some of these systems are implemented on FPGAs for vocabulary sizes ranging from 5K to 92K words. These systems deliver faster processing speeds by optimizing resource allocation and memory

bandwidth utilization [13–15]. FPGA designs do not predominate in low-power speech recognition hardware designs.

For the purpose of low-power large vocabulary speech recognition and low-power voice activity detection (VAD), ASIC technologies are achieving incomparable power efficiencies. For vocabulary sizes ranging from 5K to 60K words, power consumption can be kept under a few hundred mW [16–18] to a few watts [19]. A recent research reports a power consumption of 5mW for a 5K word recognizer [20]. Tradeoffs between power consumption and recognition accuracy can be made depending on design requirements. In addition, average system power consumption for continuous time speech recognition can be reduced by combining the large vocabulary speech recognizer with a front-end VAD. A wide range of algorithms are proposed for VAD design and existing ASIC designs have brought down the VAD power consumption to as little as 6μWs [21].

The field of μW voice-command recognition hardware for small vocabulary sizes remains unexplored.

1.5 Contributions

In this thesis, we develop an architecture for voice-command recognition systems. The proposed system achieves ultra-low power consumption by performing early stage signal dimension reduction and adaptive signal acquisition and processing. The system consists of a feature extraction front-end and a recognition back-end. We propose new techniques for both components.

Through cepstral analysis, we first show that the most essential information for speech recognition can be captured within a few judiciously selected spectral features. By pre-selecting a small subset of features, our system allows for lower-rate signal acquisition and simpler down-stream processing. Using the technique of multi-coset sampling, we propose an efficient procedure for extracting the pre-selected spectral features by sampling the signal at its Landau rate (i.e., the minimum rate required to sample the signal).

We propose two variations for the front-end design, which are separately applied to the applications of text-dependent speaker-verification and user-independent command recognition. We develop low-complexity and adaptive back-end feature processing al-

gorithms for both applications. A weighted dynamic time warping algorithm is developed for the speaker verification application. It removes signal misalignments and mitigates speech rate variations while preserving the shapes of the signal envelope. For user-independent command recognition, we propose a sparsely-connected multi-band neural networks model that enables adaptive feature selection and low-complexity computing.

We also propose a hardware design for the proposed system using analog components combined with a digital processor. Parts of the system are implemented in hardware. The end-to-end system power consumption is estimated to be under a few hundred μWs. The proposed system demonstrates comparable accuracy, improved noise robustness and much lower power consumption compared to conventional systems.

1.6 Thesis outline

In Chapter 2, we introduce the architecture of the proposed system and its components. We compare and differentiate the proposed system with the conventional speech recognition systems by examining their high-level architecture as well as individual components. The proposed system can be broken down into the feature-extraction front-end (Chapters 3 and 4) and the recognition back-end (Chapters 5 and 6).

In Chapter 3, we show that, by utilizing the special structures present in speech sounds, we can capture the most essential information for speech recognition within handful of narrowbands using an analog filterbank. In Chapter 4, we then propose an efficient procedure for extracting the narrowband spectral features using the method of multi-coset sampling.

The narrowband feature selection and extraction procedure has two variations. In Chapter 5, we explore applications in speaker-verification with narrowband features selected around the harmonics of the speaker's fundamental frequency. We introduce the weighted dynamic time warping algorithm, which achieves improved recognition accuracy and reduced signal envelope mutation through adaptive warping path constraining.

In Chapter 6, we integrate the front-end design to a user-independent command recognition back-end. In this case, the features are universal for all users and are selected without the knowledge of the speaker's fundamental frequency. For command recogni-

tion, we propose a low-complexity multi-band neural network model that is compatible with the feature adaptive front-end and that achieves comparable recognition accuracy as the state-of-the art techniques.

A hardware implementation of the system is presented in Chapter 7. Power consumption of the analog components are estimated using existing technologies. The digital components of the system are implemented on a low-power micro-controller and their power consumptions are then measured. Finally, we conclude our studies and discuss future research directions in Chapter 8.

Chapter 2

An architecture for low-power voice-command recognition

A voice-command speech recognition system can be broken down into two stages: an acoustic feature extraction front-end maps the complex and information-rich speech waveform into a low-dimensional feature space, and a speech recognition back-end identifies whether a candidate command was uttered by performing computation on the features. In conventional systems, the feature extraction unit is common to most applications, including large vocabulary continuous speech recognition as well as KWS. On the other hand, there is a large number of variations available for the recognition stage to serve different applications. What is common among these system is that, they usually require multiple stage processing and complex computation on high dimensional features.

In this chapter, we describe the architecture proposed for low-power voice-command recognition, and compare it with the architecture of a conventional speech recognition system. We show that, by using new types of front- and back-ends, the proposed system architecture enables significantly lower-complexity computation and improved system power efficiency for the application of voice-command recognition.

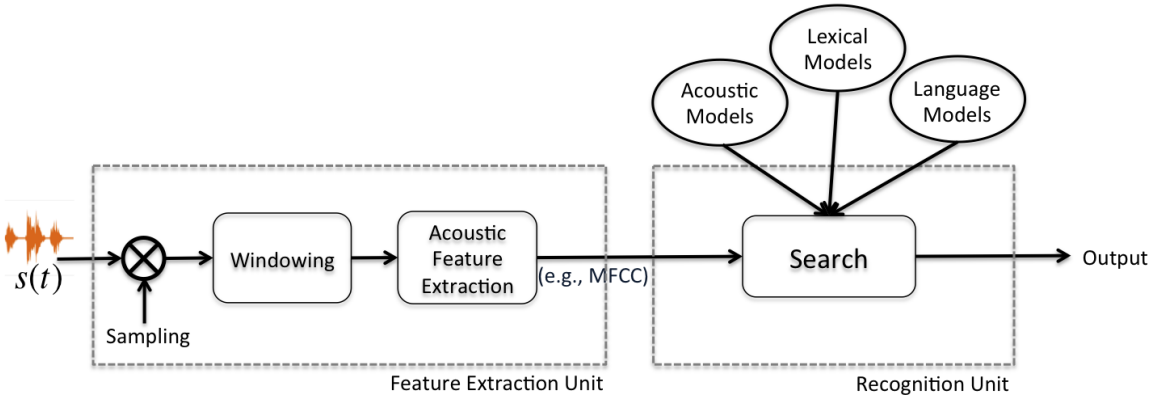


Figure 2-1: Architecture of a conventional speech recognition system, which can be broken down into a feature extraction front-end and a recognition back-end. The steps taken to extract the acoustic features, including sampling, windowing and feature computation are common in most ASR systems. The speech recognition unit can be considered as searching for the word(s) that best matches the input features and are designed differently for different applications.

2.1 A conventional speech recognition system

Figure 2-1 shows the system architecture that is widely adopted in conventional speech recognition systems. The system is composed of the feature extraction front-end and the recognition back-end. The acoustic feature extraction process is completed in three stages. First, analog-to-digital (ADC) conversion quantizes the analog acoustic signal into digital samples. Then, the digital samples are segmented into overlapping window frames. Each frame is processed through an acoustic feature extraction unit, which transforms the raw speech frame into a vector of acoustic features. Some of the most popular acoustic feature representations include the Mel-frequency cepstral coefficients (MFCCs) [9] and the perceptual linear predictive (PLP) coefficients [9, 22]. The acoustic feature extraction unit aims at condensing the high-rate raw samples into a low-dimensional feature vector, which provides a compact representation of the speech content.

The recognition step can be considered as searching for the word(s) that best matches the input features. Depending on the specific application, the recognition unit often incorporates prior knowledge to guide the search process. For example, acoustic models, lexical models and languages models trained off-line with a pool of training data are often used in speech recognition to provide a statistical representation of spoken lan-

guage [8,12]. Many core technologies such as the Gaussian mixture models (GMMs), hidden Markov models (HMMs) and n-gram language models are widely applied to model different aspects of speech.

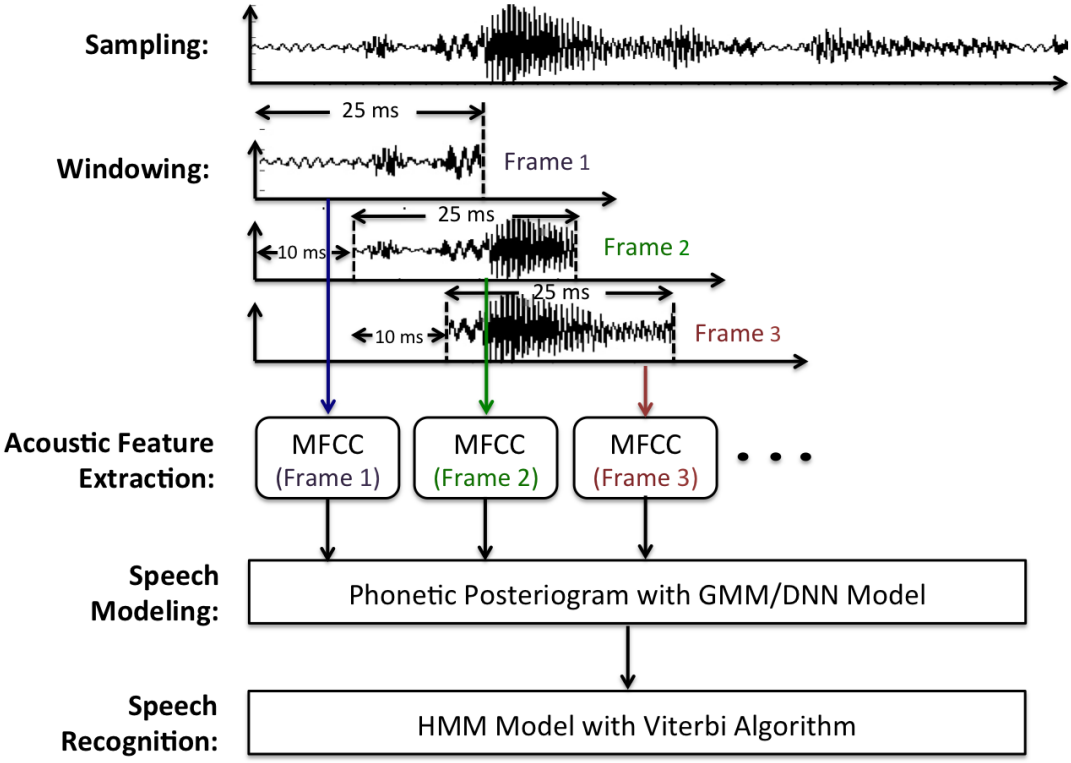


Figure 2-2: An example of the conventional ASR process: the raw signal is sampled at 16kHz and then framed with 25ms windows with 10ms spacing. The MFCC acoustic features are extracted for each frame. Then, the raw acoustic features are applied to the widely used GMM models, which generates a phonetic posterior given the acoustic features. The sequence of phonetic posteriogram is then fed through a HMM model to complete the word search.

Figure 2-2 provides an example of the speech recognition process with parameters typically used in conventional systems. Usually, wide-band speech signals are sampled at around 16kHz. The samples are then segmented into overlapping frames with 25ms window length and 10ms spacing. In this case, each frame consists of 400 samples and every frame is then mapped to a 13-dim MFCC feature vector (often augmented with first and second MFCC derivatives) through a 40 band digital filter-bank. These signal pre-processing steps, including sampling, windowing and acoustic feature extraction, are

common in most ASR systems. The subsequent recognition component can be configured in many different variations depending on the specific application.

In the example illustrated in Figure 2-2, the recognition system first performs acoustic modeling with GMMs or artificial neural networks (ANNs) [7, 23–25] to generate a phonetic posteriogram of the speech input. The time-sequence of phonetic posteriograms are then fed through a HMM to complete the word search.

There are many alternative algorithms designed for the recognition task. For small vocabulary KWS, there are template based algorithms [26, 27] and model based algorithms that use HMM models [28–30] or DNNs [31] to model the targeted keywords. Some apply the large-vocabulary continuous speech system for the application of KWS [32]. Other options include distance based methods such as nearest neighbors [33] and support vector machines (SVMs) [34].

Systems based on existing feature extraction and recognition algorithms are prohibitively power-consuming for stand-alone hand-held devices due to several reasons. First, speech signals are sampled at a rate that is equal to or above the signal Nyquist rate. The task of removing irrelevant information such as environment information, artifacts due to speech variation, speaker information and background noise, is left to the acoustic feature extraction algorithm and later processing steps. This approach of first digitizing data at a high rate and then performing dimension reduction is quite power expensive because power consumption of ADCs scale linearly with its sampling rate [35] and the complexity of downstream processing also scales proportionally with the input dimension. In other words, if we could perform early stage signal dimension reduction directly on the analog signal, computation complexity of the overall system may be reduced.

Secondly, without constraints on power consumption or computation complexity, conventional systems acquire and process the maximal amount of information at all times, aiming to achieve the optimal accuracy. This blind approach is not desirable under system power constraints. Depending on the signal quality, the marginal gain in accuracy by using more data may be diminishing. As a result, system power efficiency can be improved if feature extraction and processing are adaptively optimized based on the input signal quality.

2.2 Architecture for low-power voice command recognition

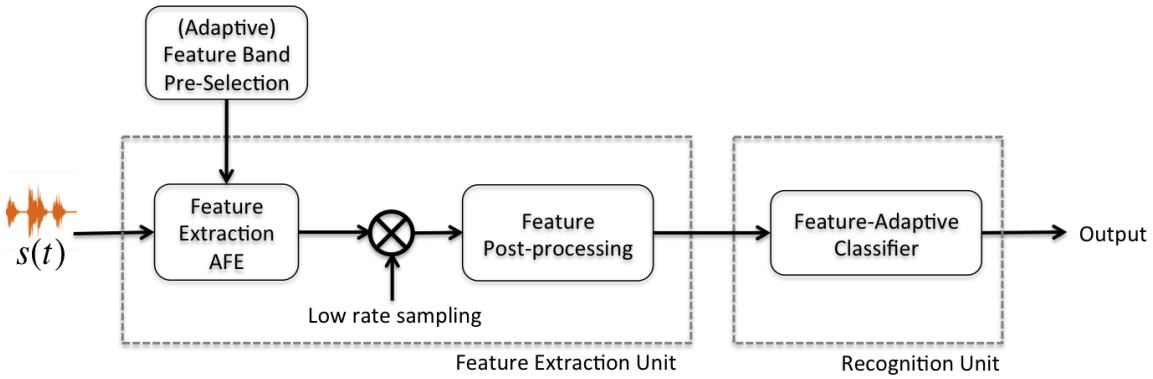


Figure 2-3: Architecture of the proposed low-power command recognition system: it performs spectral domain feature extraction directly on the analog waveform before being converted to digital samples. No prior model is included and classification is conducted directly on the acoustic features.

On the other hand, Figure 2-3 presents the high-level architecture of the proposed system which consists of an adaptive feature pre-selection unit, a mixed-signal feature extraction front-end and a low-complexity speech recognition back-end.

Desirable recognition accuracy can be achieved without acquiring and processing the full speech spectrum. The challenge mainly involves determining a set of compact features that are easy to extract with simple circuitry while still retaining sufficient information for command classification. Exploiting the structure in speech and compact representations of speech features in the spectral domain, our system reduces system complexity by pre-selecting and processing only a subset of all available features.

2.2.1 Feature pre-selection

The feature pre-selection unit adaptively selects the 'best' features in real time depending on factors such as the background noise and engineering design criteria. The feature extraction unit consists of a combination of analog filterbanks, ADCs and low-complexity digital processing. Instead of sampling the signal at its Nyquist rate and then performing feature extraction on the digitized signal, our system filters out redundant information and acquires only the pre-selected features with efficient analog front-ends (AFEs) at the beginning of the processing pipeline. Signal sampling is performed on the sparse spectral

features with a few uniform low-rate samplers, each running at a few hundred hertz. To fully exploit the benefits of feature pre-selection, we propose efficient sampling and feature extraction algorithms such that the total sampling rate and processing complexity are proportional to the amount of information content extracted by the AFE.

2.2.2 Feature adaptive recognition

The back-end recognition unit utilizes different decision making algorithms depending on the specific application. A special characteristic that distinguishes our recognition back-ends from conventional systems is its adaptivity on feature inputs. Our system back-ends are designed to be compatible with the adaptive feature extraction front-end such that the recognition decision can be made with an arbitrary set of input features. In the sequel, we propose multiple command recognition algorithms for different applications including text-dependent speaker-verification and user-independent command recognition.

System support for adaptive processing is highly beneficial because speech content for recognition is redundant in the spectral sub-bands, and a subset of all the available bands can be sufficient for the recognition task. Hence, the proposed system scales processing power by using fewer sub-band features when there is no background noise and more when there is noise. In addition, conventional systems concatenate all sub-band features into a super-vector regardless of noise conditions, which could result in poor recognition even when a single band is corrupted. In contrast, we can mitigate the loss of granular SNR by actively discarding the noisy features and retaining only the high quality ones [5, 36, 37]

2.3 Summary

In this chapter, we introduced conventional ASR architectures and proposed a new architecture for low-power voice-command recognition systems. In a conventional ASR system, the speech signal is first sampled at rates equal to or higher than the Nyquist rate. The high-dimensional digital signal is then processed with a large number of sub-

sequent steps to extract its low-dimensional features. The recognition unit analyzes the features to make the final recognition decision through multi-stage processing.

In contrast, the proposed system performs signal dimension reduction directly on the analog signal before it is digitized. First, spectral domain features are pre-selected based on information such as the fundamental frequencies of speech and the noise spectrum. The selected features are acquired by filtering the speech signal with an analog filterbank and then digitized at a much lower rate than the signal Nyquist rate. The recognition back-ends are designed to be compatible with the adaptive feature selection front-end such that a recognition decision can be made using an arbitrary subset of features.

By using feature pre-selection, early-stage feature extraction with an AFE and feature-adaptive recognition, the proposed system operates at much lower sampling and processing rates than conventional systems, thus enabling a low-power system designs.

Chapter 3

The narrowband spectral features

In speech processing, the purpose of the acoustic feature extraction step is to map the complex and information-rich raw speech waveform into a low-dimensional representation, which is used as inputs to the back-end pattern recognizer to distinguish examples of different classes (i.e., identifying the uttered word). This feature extraction step not only transforms the input into a more compact form, but it also helps to reduce the variabilities of examples of the same class. In the case of speech signals, the acoustic features usually consist of some evaluation of the signal power spectrum. The accuracy of the final recognition decision heavily depends on the effectiveness of the feature extraction step.

Current research mainly focuses on the recognition unit instead of the feature extraction unit. This is because popular acoustic features such as the MFCCs, Mel-frequency spectral coefficients (MFSCs) (i.e., the filterbank features [38]) and the PLP coefficients have demonstrated unparalleled experimental performance. Even though these conventional features yield excellent recognition accuracy, the feature extraction procedure for these features generally requires high fidelity raw speech data and a multi-stage processing pipeline, which involves high-complexity processing. Therefore, they are not directly applicable to our low-power system design.

In this chapter, we first review the basics of speech generation and how speech content is embedded in the speech spectrum. Then, we study existing speech feature extraction schemes to better understand how they are designed and why they are overly complex for our system. Through cepstral analysis, we propose a new set of acoustic features.

These features consist of spectral contents extracted from a small subset of the full speech spectrum. We show through analysis that these selected features contain the most essential information for speech recognition and can be efficiently extracted in the analog domain. Through early-stage signal feature pre-selection using an analog front-end (AFE), the sampling and processing rates for all downsteam components can be reduced.

3.1 Background on speech sound generation

Before we start to analyze the feature extraction process, it is helpful to first review how speech signals are generated and their representations in terms of spectrograms. A spectrogram is a visual representation of the speech power across the frequency spectrum as time varies. It is obtained by first taking the short-time Fourier transform (STFT) of the speech signal and then computing its power.

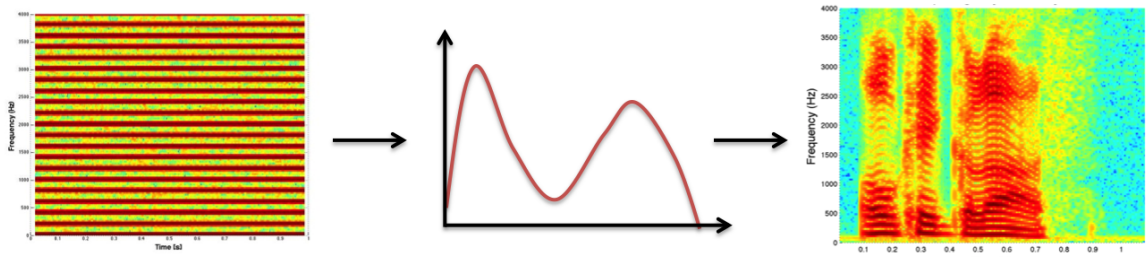


Figure 3-1: Spectrogram illustration of the speech sound generation process.

Figure 3-1 shows the spectrograms of the speech generation process. The horizontal axis of the spectrogram corresponds to time; the vertical axis corresponds to frequency; and the color temperature at a point indicates the amount of power within its corresponding time-frame and frequency. First, the vocal fold vibration generates a glottal pulse, whose form is usually represented by a sinusoidal wave at the fundamental frequency. As a result of the signal periodicity, harmonics of the glottal pulse are created as it travels through the speaker's vocal tract. In Figure 3-1-(a), the harmonics of the fundamental frequency are shown as evenly spaced horizontal stripes with spacings equal to the fundamental frequency. Speeches with low fundamental frequency have closely packed harmonics; whereas, in the high fundamental frequency case, the harmonics are more spaced out. Usually, the low frequency harmonics carry more energy than higher

frequency harmonics, and men's voice tend to have lower fundamental frequency than women's. As the glottal pulse travels through the speaker's vocal tract, it is modulated by the vocal tract transfer function. Figure 3-1-(b) shows an example of the vocal-tract modulation function. Since we are constantly changing the shape of the vocal tract when we speak, the vocal tract can be considered as a time-varying filter. Figure 3-1-(c) shows the spectrogram of the final speech signal.

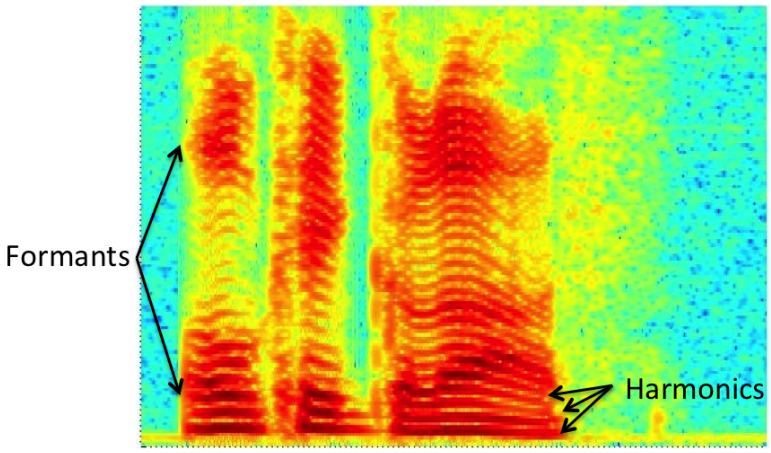


Figure 3-2: An example of the spectrogram of the speech command 'OK Glass'. The horizontal strips correspond to the harmonics. The high energy spectral components correspond to the formants, which are the resonances of the vocal tract. They contain important speech information.

Figure 3-2 corresponds to the spectrogram of the utterance 'OK Glass'. The horizontal strips are located at multiples of the fundamental frequency. These are the harmonics of the glottal pulse. If we look across the frequency spectrum, we will notice that the energy at some frequency bands are suppressed while the energy at other frequency bands are emphasized. This is a result of the vocal tract modulation. The frequencies with relatively high energy concentration correspond to the formants. They are also called the resonance of the vocal tract and they carry important speech information. The entire spectrogram contains a lot of extra information that is not useful for speech recognition. Therefore, the feature extraction process aims at condensing the information-rich raw speech data and extracting a compact representation of the essential speech information (e.g., the vocal-tract modulation function).

As we will see, cepstral analysis offers an efficient representation of the speech signal as speech is sparse in the cepstral domain. For example, the widely used MFCC features correspond to a speech signal representation when it is transformed to the cepstral domain [9]. In addition, the homomorphic property of the cepstrum allows us to easily separate different components of speech and extract the useful features for a dedicated application [39–41].

3.2 Cepstral analysis of speech

In this section, we review the basis of cepstral analysis. Referring to Section 3.1, a short segment of speech signal, denoted by $s(t)$, can be modeled as a time-domain convolution between the excitation signal, $e(t)$, and a time-invariant vocal tract modulation function, $h(t)$ for the given speech segment:

$$s(t) = e(t) * h(t). \tag{3.1}$$

For voiced sounds, $e(t)$ is a periodic glottal pulse with fundamental frequency f_0. For unvoiced sounds, $e(t)$ can be modeled as a stochastic noise sequence. It is understood that most of the speech information is embedded in the time-varying vocal tract modulation function $h(t)$ [8,9,42].

Taking Fourier transforms, the convolution relationship in (3.1) becomes multiplication in the frequency domain:

$$S(f) = E(f) \cdot H(f).$$

Taking the logarithm of the power spectral density (PSD), the multiplication operation is converted to summation:

$$\hat{S}(f) = \hat{E}(f) + \hat{H}(f), \tag{3.2}$$

where, $\hat{S}(f)$, $\hat{E}(f)$ and $\hat{H}(f)$ denote $\log|S(f)|$, $\log|E(f)|$ and $\log|H(f)|$, respectively. By taking the inverse Fourier-transform (IFT) of the logarithm of the PSD, the signal is transformed to the cepstral domain. Let us use $\hat{s}(\tau)$, $\hat{e}(\tau)$ and $\hat{h}(\tau)$ to denote $\text{IFT}(\hat{S}(f))$, $\text{IFT}(\hat{E}(f))$ and $\text{IFT}(\hat{H}(f))$, respectively. Then, it follows from the linearity of IFT and (3.2)

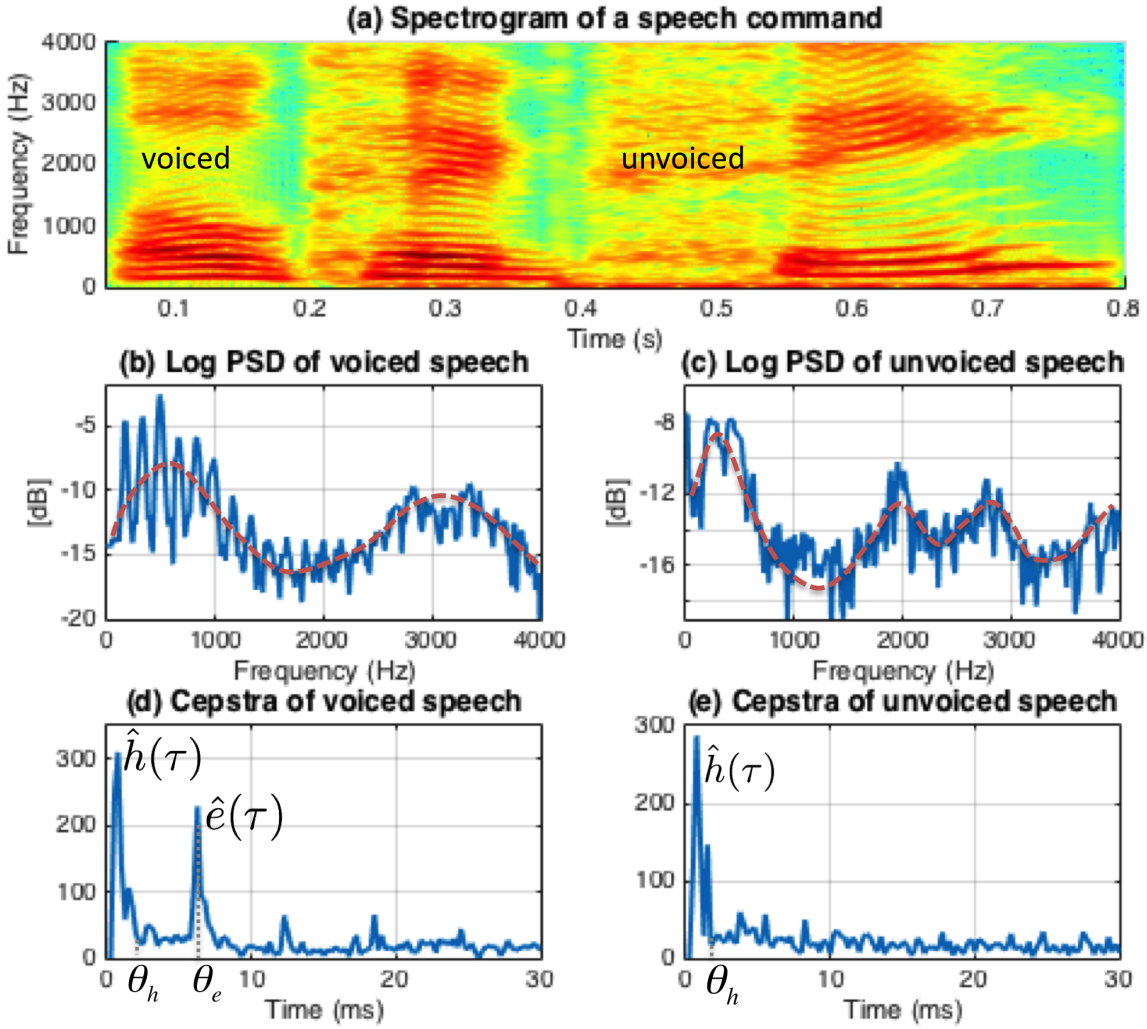


Figure 3-3: Cepstral analysis of a speech sample. (a): the spectrogram of a speech command. (b) and (c) show the logarithm of the PSD of the speech segment. The solid lines correspond to the PSD and the dashed lines corresponds to the envelope of the PSD. (d): the signal cepstrum is sparse and consists of two components: $\hat{h}(\tau)$ and $\hat{e}(\tau)$. (e) shows the cepstra of the unvoiced frame, where only the $\hat{h}(\tau)$ component is present.

that:

$$\hat{s}(\tau) = \hat{e}(\tau) + \hat{h}(\tau).$$

Figure 3-3 illustrates the process of cepstral analysis. Figure 3-3-(a) shows the spectrogram of a speech signal command over a 1s duration. If we fix a time and take a slice of the spectrogram, we obtain a vector of numbers representing the signal power across the spectrum at that specific time. Let us take the logarithm of the power values (like what our auditory system would do) and plot it, we then get the logarithms of the power

spectral density of the signal at a fixed time frame, as shown in Figure 3-3-(b) and (c). The narrow spikes (solid lines) are due to the excitation component $\hat{E}(f)$ and the signal envelopes (dashed lines) correspond to the modulation function $\hat{H}(f)$ and the high frequency falloff of speech. The excitation components contain important information about the user, but do not contain much useful information about the speech content except for tonal languages. Most of the useful speech information is embedded in the slow-varying envelope of the spectrum [42].

Figure 3-3-(d) and (e) show the signal cepstrum. The horizontal axis of the cepstrum diagram corresponds to quefrency, which has the unit of ms (i.e., cycle/kHz) and is an indication of how fast the power spectrum varies with frequency. The vertical axis corresponds to the amplitude of the cepstral component at a certain quefrency. When transformed to the cepstral domain, the speech signal becomes sparse. The low-quefrency component corresponds to vocal-tract modulation: $\hat{h}(\tau)$, and the higher-quefrency component corresponds to the excitation signal, $\hat{e}(\tau)$. Usually, the location, θ_e, of the excitation component is much higher than the cutoff quefrency, θ_h, of $\hat{H}(f)$. The location of the excitation component θ_e is related to the speaker's fundamental frequency f_0 by: $\theta_e = 1/f_0$ s (i.e., cycle/Hz). The low-quefrency components contain most of the information for speech recognition and are often extracted as acoustic features in conventional systems [39–41].

Figure 3-4 plots the cepstral coefficients throughout a one second speech utterance. The horizontal axis corresponds to time and the vertical axis corresponds to quefrency. The amplitude of the cepstral coefficient at a specific time and quefrency is indicated by the color temperature. As show in the figure, the vocal tract modulation component of the signal stays small throughout the one second speech utterance and it is a result of the physical limitation of the vocal tract and it holds for general speech [42]. As a result, a conventional approach for extracting the $\hat{h}(\tau)$ components and discarding the redundant and irrelevant information is to perform 'liftering', which is equivalent to low-pass filtering in the cepstral domain [39]. The widely used MFCCs are obtained in this manner.

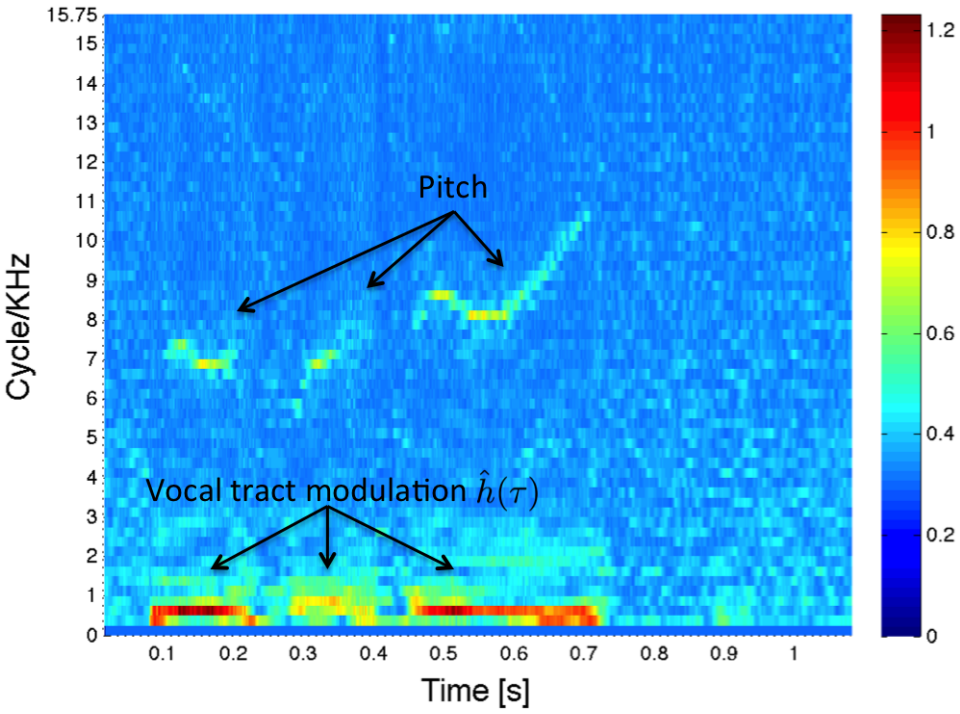


Figure 3-4: Cepstral coefficients of a speech utterance with 1s duration. It indicates that the formant quefrency stays low throughout the speech segment.

3.3 Mel-frequency cepstral coefficients

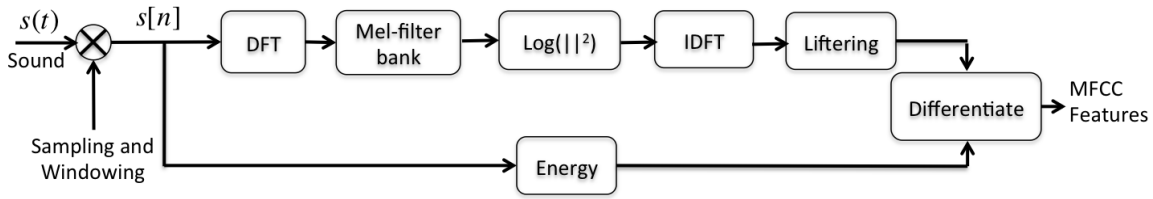


Figure 3-5: Block diagram of the MFCC feature extraction process. The computation is performed at every frame and a MFCC feature vector of length around 13 is computed for each frame.

MFCC features are widely used in speech recognition systems. Figure 3-5 shows the block diagram of the MFCC feature extraction system. First, the analog speech sound is converted to digital samples at a rate of around 16 kilo-samples per second. To obtain the MFCC coefficients, each speech frame is transformed to the frequency domain by taking its STFT. Then, the frequency axis is rescaled based on the Mel-frequency scale, which was developed experimentally to approximate human auditory perception system. The

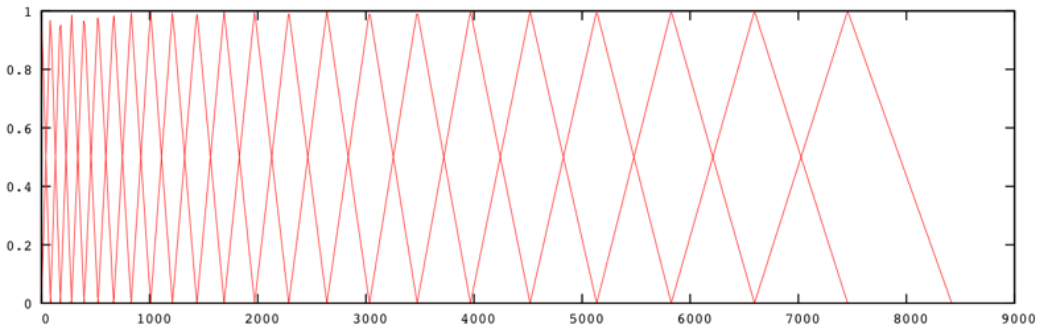


Figure 3-6: Frequency response of a 23-band Mel-frequency filter bank example

frequency response of the Mel-scale filter bank is shown in Figure 3-6. For frequencies below 1kHz, the filter bands are placed with linear spacing. Above 1kHz, the scale becomes logarithmic and the Mel-frequencies are approximated with the equation:

$$F_{\text{mel}} = \frac{1000}{\log_{10}(2)} \cdot \left[1 + \frac{F_{\text{Hz}}}{1000}\right].$$

To compress the signal dynamic range, the logarithm of the power amplitude for each band is taken to obtain the Mel-frequency spectral coefficients (MFSCs). Finally, the MFSCs are transformed to the cepstral domain by taking the IDFT. As described in Section 3.2, the signal is sparse in the cepstral domain and only the coefficients corresponding to the low quefrency contents are useful for speech recognition. As a result, we perform liftering to obtain the low-quefrency components of the cepstral coefficients (e.g., the first 13 coefficients). These are the MFCCs. The first and second derivatives of the MFCCs represent the 'velocity' and the 'acceleration' of the power evolution with respect to time. They are combined with the first-order MFCCs, as well as total energy, to form a feature super-vector, whose dimension is usually 40. For example, if speech signal is processed at 100 frames per second. A one second speech segment will be represented by a time sequence of 100 MFCC vectors, each having a dimension of 40.

MFCC features have yielded good performance for speech recognition purposes as (1) they are informative of speech content; (2) elements of the MFCC feature are approximately uncorrelated and, thus can be well modeled by the GMMs; and (3) speech can be represented with a small number of coefficients due to signal sparsity in the cepstral

domain. Nevertheless, they are not suitable for our low-power system because the feature extraction process itself is complex. It requires sampling the full signal spectrum and transforming the time-domain samples to the cepstral domain before discarding the redundant information. Hence, due to sampling and processing of the high-dimensional raw speech signal and the large number of steps involved in feature extraction, it is highly desirable to seek an alternative when power consumption is a constraint.

3.4 Narrowband features for speech recognition

Given that a major driver of power consumption in cepstral domain feature extraction is the high-rate sampling and the pre-processing required to transform the signal to the cepstral domain, we propose a feature extraction method that performs dimensional reduction directly on the time-domain signal, using a small number of analog narrowband filters. We will show that substantially the same features can be extracted through analog filtering of the raw speech waveform by exploiting certain properties of speech.

Figure 3-7-(a) depicts the logarithm of the PSD of a typical speech frame. The fast fluctuation corresponds to the glottal pulse excitation $\hat{E}(f)$ at the fundamental frequency f_0 and its harmonics, and the envelope (dashed line in Figure 3-7-(c)) outlines $\hat{H}(f)$, the vocal tract modulation function. The cepstral domain also shows these two components: $\hat{e}(\tau)$ represented by a delta function at θ_e and $\hat{h}(\tau)$ represented by a narrow triangle (Figure 3-7-(b)). Since the most essential information of $\hat{h}(\tau)$ is concentrated at the low-quefrencies (typically under 2-3 cycle/kHz [39, 42]), the $\hat{h}(\tau)$ component is shown with a cutoff at θ_h in Figure 3-7-(b).

The constraint that $\hat{h}(\tau)$ is assumed to be (cepstrally) band-limited to low quefrencies allows the opportunity to 'under-sample' the spectral domain signal. Consider the case where we sample $\hat{S}(f)$ at a set of evenly-spaced points (dots in Figure 3-7(c)). The point sampling function is defined by $P(f)$:

$$P(f) = \sum_{k \in \mathbb{Z}} \delta(f - k\Delta_p),$$

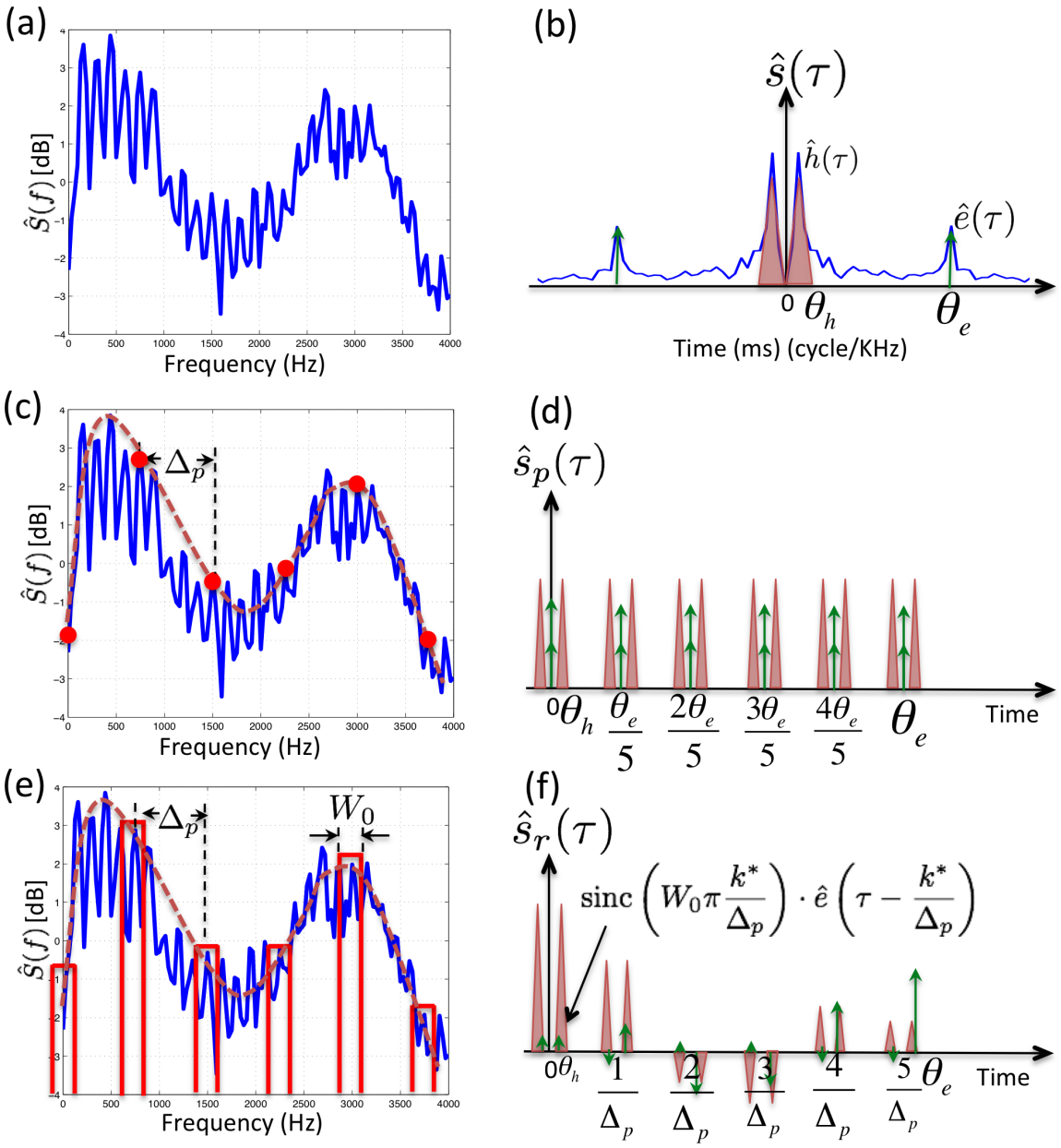


Figure 3-7: Narrowband feature extraction: (a) and (b) show the PSD and the cepstrum of a speech segment. The cepstrum is simplified as the summation of $\hat{h}(\tau)$ (triangle shape) and $\hat{e}(\tau)$ (delta function). In (c), $\hat{S}(f)$ is measured at evenly spaced points (denoted by $\hat{S}_p(f)$). Δ_p is an integer multiple of f_0. In (d), $\hat{s}_p(\tau)$ (cepstrum of $\hat{S}_p(f)$) is an aliased version of $\hat{s}(\tau)$. In (e), $\hat{S}(f)$ is measured with evenly spaced rectangular functions with arbitrary spacing, Δ_p. Aliasing between $\hat{h}(\tau)$ and $\hat{e}(\tau)$ occurs in (f) and $\hat{e}(\tau)$ is attenuated with the sinc function.

where $\Delta_p = \beta f_0$ is an integer multiple of the fundamental frequency. In the example in Figure 3-7(c), $\beta = 5$.

The sampled PSD, $\hat{S}_p(f)$, can be expressed as the product of $\hat{S}(f)$ and the sampling function $P(f)$:

$$\hat{S}_p(f) = \hat{S}(f) \times \sum_{k \in \mathbb{Z}} \delta(f - k\Delta_p).$$

The cepstrum of $P(f)$ is another set of delta functions spaced by $1/\Delta_p$. Since multiplication becomes convolution in the cepstral domain, the cepstrum of $\hat{S}_p(f)$, denoted by $\hat{s}_p(\tau)$, is an aliased version of $\hat{s}(\tau)$ (Figure 3-7(d)):

$$\hat{s}_p(\tau) = \sum_{k \in \mathbb{Z}} (\hat{e}(\tau - \frac{k}{\Delta_p}) + \hat{h}(\tau - \frac{k}{\Delta_p})).$$

As long as we choose $\Delta_p < \frac{1}{2\theta_h}$, repetitions of $\hat{h}(\tau)$ and $\hat{e}(\tau)$ will not overlap. With $\Delta_p = \beta f_0 = \beta/\theta_e$, copies of $\hat{e}(\tau)$ occur at 0 and multiples of θ_e/β (Figure 3-7-(d)). Hence, the vocal tract modulation components, $\hat{h}(\tau)$, are not corrupted by aliasing and are preserved in the 'sampled' spectrum $S_p(f)$. What this implies is that if we have a good estimation of the fundamental frequency, f_0, a few judiciously selected points from the signal PSD can capture most of the essential speech information $\hat{h}(\tau)$.

What if the estimation of the fundamental frequency f_0 is not accurate? In this case, $\hat{e}(\tau)$ is not centered around 0 and may be aliased with $\hat{h}(\tau)$. This problem can be mitigated by 'sampling' $\hat{S}(f)$ with rectangular windows instead of delta functions. As shown in Figure 3-7-(e), we measure $\hat{S}(f)$ using a set of evenly spaced rectangular windows (implemented as a set of narrowband filters). The rectangular window train can be expressed as the convolution of the point sampling function $S_p(f)$ and a scaled rectangular function of width W_0:

$$\begin{aligned} G(f) &= P(f) * \text{rect}_{W_0}(f), \\ &= \sum_{k \in \mathbb{Z}} \text{rect}_{W_0}(f - k\Delta_p), \end{aligned}$$

where,

$$\text{rect}_{W_0}(f) = \begin{cases} \frac{1}{W_0}, & \text{if } \frac{-W_0}{2} < f < \frac{W_0}{2} \\ 0, & \text{otherwise.} \end{cases}$$

Since the cepstrum of the rectangular function is a sinc function and convolution in the frequency domain becomes multiplication in the cepstral domain, the cepstrum of $G(f)$ is an impulse train whose amplitudes are scaled by the sinc function:

$$\hat{g}(\tau) = \sum_{k \in \mathbb{Z}} \text{sinc}(W_0 \pi \frac{k}{\Delta_p}) \delta(\tau - \frac{k}{\Delta_p}).$$

Therefore, the filtered spectrum, $\hat{s}_r(\tau)$, is an aliased version of $\hat{s}(\tau)$ where the amplitudes of the aliased copies are scaled by the amplitude of a sinc function as follows:

$$\begin{aligned} \hat{s}_r &= (\hat{h}(\tau) + \hat{e}(\tau)) * \hat{g}(\tau), \\ &= (\hat{h}(\tau) + \hat{e}(\tau)) * \left(\sum_{k \in \mathbb{Z}} \text{sinc}(W_0 \pi \frac{k}{\Delta_p}) \delta(\tau - \frac{k}{\Delta_p}) \right), \\ &= \sum_{k \in \mathbb{Z}} \text{sinc}\left(W_0 \pi \frac{k}{\Delta_p} \right) \left(\hat{e}(\tau - \frac{k}{\Delta_p}) + \hat{h}(\tau - \frac{k}{\Delta_p}) \right). \end{aligned}$$

This is illustrated in Figures 3-7-(e) and 3-7-(f). The modulation function $\hat{h}(\tau)$ is now aliased with $\hat{e}(\tau - k^*/\Delta_p)$, where,

$$k^* = \left\lfloor \frac{\theta_e}{1/\Delta_p} \right\rfloor, \tag{3.3}$$

and the location of aliasing is offset from 0 at $(\theta_e - k^*/\Delta_p)$. When $\Delta_p = \beta f_0$, this offset is equal to 0. As indicated in Figure 3-7-(f), the amplitude of the aliasing component is scaled by a sinc function:

$$\text{sinc}\left(W_0 \pi \frac{k^*}{\Delta_p} \right) \cdot \hat{e}\left(\tau - \frac{k^*}{\Delta_p} \right).$$

As a result, the wider the filter bandwidth W_0, the more attenuation there is on $\hat{e}(\tau - k^*/\Delta_p)$, and hence, the less $\hat{h}(\tau)$ will suffer from aliasing with $\hat{e}(\tau)$. As long as $\Delta_p < \frac{1}{2\theta_h}$ is still satisfied, $\hat{h}(\tau)$ will not be corrupted by its own aliases.

For example, with a filter bank spacing of $\Delta_p = 800\text{Hz} = 0.8\text{kHz}$, filter bandwidth $W_0 = 0.2\text{kHz}$ and speech fundamental frequency $f_0 = 100\text{Hz} = 0.1\text{kHz}$, the low quefrency corruption from the component of $\hat{e}(\tau)$ is approximately $-0.04\hat{e}(\tau - k^*)$. In short, the information captured with the set of narrowbands retains the vocal tract modulation component, $\hat{h}(\tau)$, and can be used as features for downstream voice-command recognition.

3.5 Applications of the narrow-band spectral features

The narrow-band spectral features can be used when we have knowledge of the fundamental frequency, f_0, as well as, when we do not know f_0. When f_0 is known, the bandwidths of the narrowband features can be narrower and they are centered around the harmonics of f_0. We call this the 'f_0 dependent narrow-band spectral coefficients' (NBSCs). For a detailed implementation guide, see Appendix D.

When f_0 is unknown, the narrow-bands are chosen to be evenly spaced across the frequency spectrum, as described in Section 3.4. We call this the 'universal NBSCs'.

3.6 Summary

In this chapter, we have shown that by filtering the signal with a set of narrowband filters, which are centered around the harmonics of the speech signal and are evenly-spaced across the frequency spectrum, essential speech information for speech recognition is preserved.

When we have an accurate estimate of f_0, the features are selected to narrowly center around the harmonics of speech, which possess high signal energy concentration. With these high in-band SNR features, the system can potential achieve better accuracy than using the general Mel-frequency band features.

It is important to point out that even when the narrowband features are extracted around the harmonics, which uses the information of f_0, the exact value of f_0 may be lost.

For example, if two speakers have very similar vocal tract characteristics $\hat{h}(\tau)$, but one person's fundamental frequency is an exact multiple of the other person's, narrowband features from these two speakers may be indistinguishable.

When f_0 is unknown or inaccurate, the insufficiency in fundamental frequency estimation can be compensated by increasing the bandwidth of the narrowband filters.

The narrowband spectral features are preferable because they require a smaller number of filters (e.g., ~ 10) in comparison to a set of $26 - 40$ filters used in conventional systems. More importantly, signal dimension reduction and feature extraction are performed directly on the time-domain signal without transformation to the cepstral domain, which reduces the processing complexity for feature extraction.

In Chapter 5, we explore applications in speaker-verification with narrowband features selected around the harmonics of the speaker's fundamental frequency (i.e., f_0 dependent NBSCs). In Chapter 6, we explore applications in user-independent command recognition with universal narrowband features that are common for all users (i.e., universal NBSCs).

Chapter 4

Narrowband acoustic feature extraction

The conventional method of acoustic feature extraction involves high-rate sampling and spectral content acquisition using a large number of filters (26 to 40). In Chapter 3, we have shown that most essential speech information can be captured within a small number of narrowbands. Hence, we propose a feature pre-selection approach in which only a subset of the full-spectrum is acquired for downstream processing and decision making. More specifically, our feature extraction front-end performs dimension reduction using an analog filterbank before digitization and processing even begin.

Figure 4-1 shows the spectrogram and the PSD of the speech signal before and after the bandpass filtering process. After bandpass filtering, the signal is sparse in the frequency domain and is concentrated in a few narrowbands. Even though the bandwidth of the signal may be unchanged by filtering, the total number of samples needed to represent the multi-band signal is significantly fewer than that dictated by usual Nyquist rate of the original speech signal. The minimum sampling rate is equal to the total bandwidth of the occupied narrowbands [43].

In this chapter, we first review existing low-rate sampling methods for extracting features from the multi-band signal and propose an efficient sampling scheme to transform the multi-band analog signal to feature vectors such that the minimum number of samples are obtained without the requirement of additional analog components beside filters.

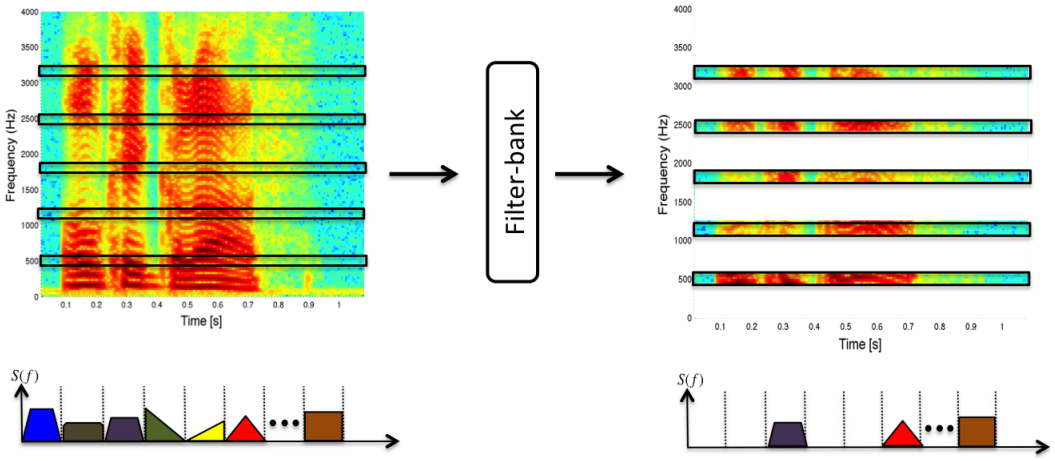


Figure 4-1: Spectrogram and PSD of a short speech sample before and after band-pass filtering.

4.1 Conventional methods for multi-band feature extraction

Figure 4-2 shows the standard approach for extracting contents from a multi-band signal through sampling. After band-pass filtering, signals in each of the sub-bands are shifted to the baseband by multiplying the signal with a sinusoidal signal. Then, each narrowband signal is filtered with a low-pass filter and digitized to discrete samples. Each ADC is operating at the Nyquist rate of the baseband signal (i.e. 2 times the bandwidth of the narrowband). In the end, logarithms of the sub-band power amplitudes are used as feature vectors. This approach requires additional analog components such as the frequency shifters and analog low-pass filters. The frequency shifters need to be tuned in real-time according to the center frequencies of the narrowband signals. The additional hardware required for this processing increases with the maximum number of narrowbands, and making it costly and infeasible for our application.

A second method for extracting the sub-band features is shown in Figure 4-3 [44, 45]. This method is widely used for analog domain acoustic feature extractions (e.g. MFCC). With this implementation, the signal in each sub-band is passed through a rectifier, which is a non-linear operator that follows the envelope of the input signal. The outputs of the rectifiers are then filtered with low-pass filters to remove the fluctuating residuals and sampled at the Nyquist rate of the sub-bands. The rectification operation has the short-

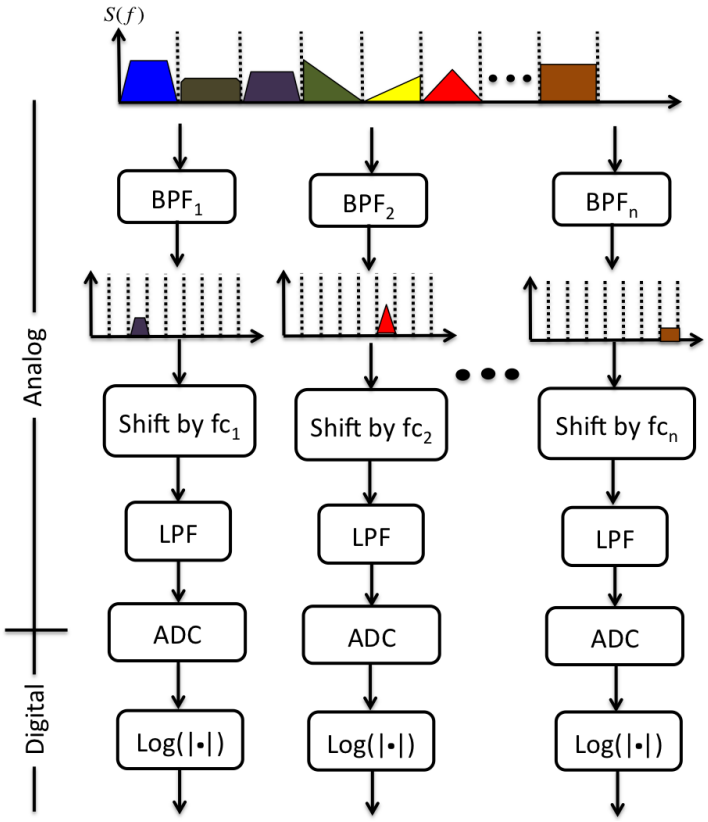


Figure 4-2: A common approach to sample a bandpass signal: each narrowband signal is shifted to the baseband, low-pass filtered and digitized separately. The additional hardware required for this operation increases with the number of non-zero narrows bands of the signal output at the bandpass filter banks.

coming of being inexact. The discrepancy between the rectified signal and the actual sub-band signal increases as the sub-band signal center frequency decreases. In addition, a rectifier is an analog component that consists of arrays of capacitors, which occupies large areas (meaning high costs) and are susceptible to process variations.

In order to simplify the sampling process and reduce the amount of the hardware required for feature extraction, we propose a method that directly sample the multi-band band-pass signal with a few low-rate uniform samplers using the method of multi-coset sampling.

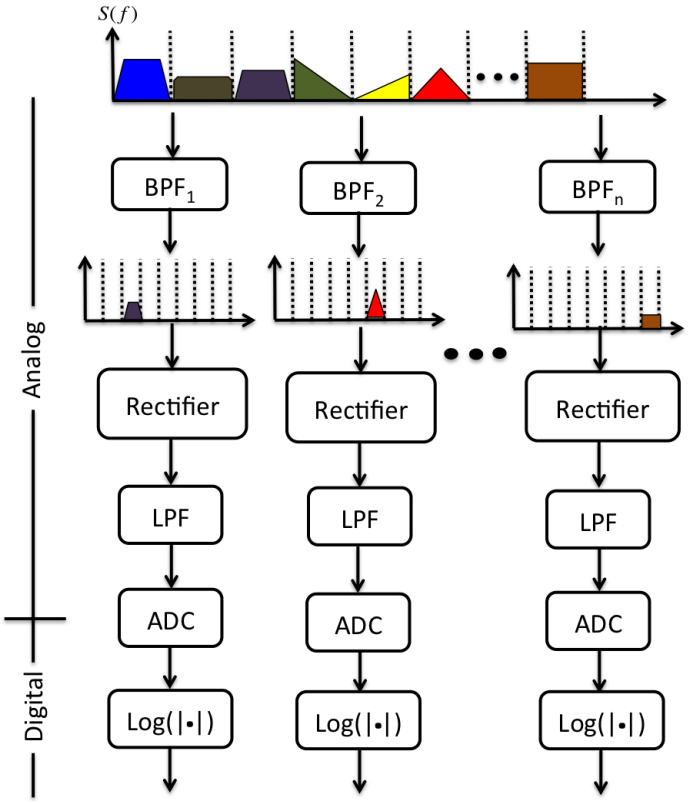


Figure 4-3: A common approach for acoustic feature extraction. Each narrowband signal is passed through a rectifier to extract its power envelope. The rectification operation is non-linear and it is followed by low-pass filtering to remove high frequency residual.

4.2 Feature extraction with bandpass filtering and multi-coset sampling

Figure 4-4 shows the flow diagram of the proposed feature extraction front-end, which samples the bandpass signal using the technique of multi-coset sampling [46, 47]. The sampling unit consists of a set of low-rate uniform samplers. The number of samplers is equal to the total number of occupied narrowbands of the signal (counting both the positive and the negative frequency spectra). The samplers are operating at the same rate, which is equal to the bandwidth of the narrowbands. Therefore, the total sampling rate is equal to the sum of bandwidths of the occupied narrowbands or the Landau rate. Since the samplers are sampling the multi-band signal at sub-Nyquist rate, the samplers' outputs are aliased versions of the multi-band signal [46]. These samplers are designed

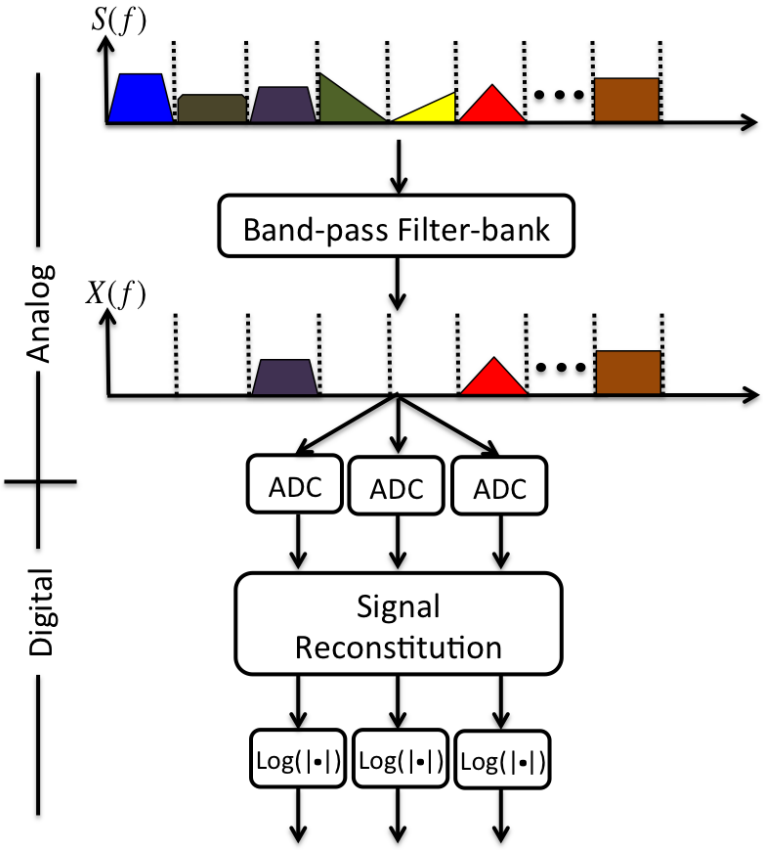


Figure 4-4: Proposed architecture for low-power feature extraction: spectral domain dimension reduction with band-pass filterbank and robust low-rate sampling with multi-coset samplers (i.e., ADCs). Implementation of the signal reconstitution module is given in Section 4.3.

to differ by a time delay from each other, which enables reconstruction of the multi-band signal from the aliased copies. A benefit of the multi-coset sampling approach is that no matter which narrow bands are active, the low-rate samplers always sample in the same manner without knowing any particular band occupation information.

The signal reconstitution module, following the coset samplers, is the core of the feature extraction front-end. It untangles the aliased low-rate samples to recover signals corresponding to each sub-band. The details of the signal reconstitution process are discussed in Section 4.2.1. After signal reconstitution, to reduce the dynamic range of the signal, we take the logarithms of the sub-band signal magnitudes to represent the speech features. A feature vector is produced at every frame and each element of the vector represents the logarithm of the signal magnitude in its corresponding narrowband at a specific

time-frame. With continuous input, the feature extraction unit outputs a time-sequence of feature vectors.

Unlike the existing methods shown in Figures 4-2 and 4-3, the proposed method does not require additional analog components such as frequency shifters, rectifiers or analog low-pass filters. Next, we show how signal reconstruction is possible and how it can be realized in systems.

4.2.1 Multi-coset sampling

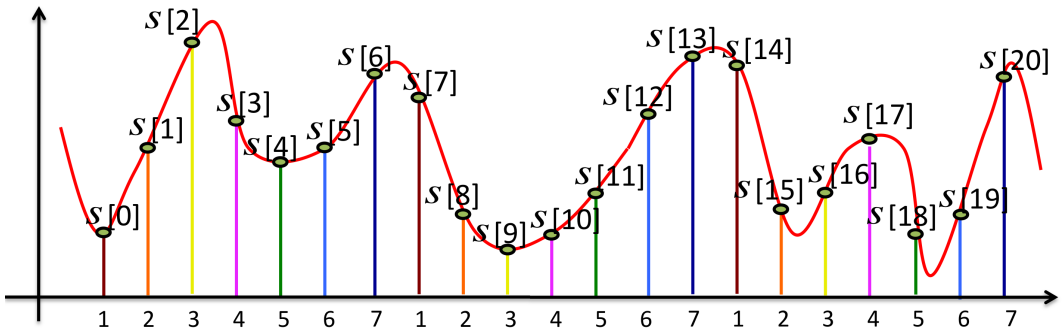


Figure 4-5: Sampling the speech signal $s(t)$ with multi-coset sampling. The number of cosets $M = 7$. The horizontal axis labels correspond to the coset index of each sample. The samples with the same color belong to the same coset.

For simplicity, let $s(t)$ denote the multi-band speech waveform (i.e., the output signal of the narrowband filterbank). The time-domain signal is shown with the red continuous curve in Figure 4-5. Let $s[n]$, $n \in \mathbb{Z}$, denote the discrete samples of the speech signal:

$$s[n] = s[0], s[1], s[2], s[3], \ldots, s[n], s[n+1], \ldots \tag{4.1}$$

As long as the sampling rate is equal to or greater than the Nyquist rate of this signal, the samples $s[n]$ provide an exact reconstruction of the continuous time multi-band signal, $s(t)$. The direct way to obtain the discrete samples, $s[n]$, is to sample with a single sampler at the desired sampling rate (i.e., a rate higher than or equal to the Nyquist rate).

Alternatively, the same samples, $s[n]$, can also be obtained by using a set of lower rate samplers with different time delays. We call the samples obtained from each sampler a coset and denote it by $s^{(m)}[n]$, where the superscript m denotes the coset index and n

denotes the index of the sample sequence. In Figure 4-5, samples from the same coset are highlighted with the same color and labeled with the same coset number along the horizontal axis. In this example, the total number of cosets is equal to 7, hence the coset index $m \in \{1,2,3,4,5,6,7\}$. Let N denote the total length of the signal segment $s[n]$ and M denote the number of cosets. Then, the relation between the m^{th} coset, $s^{(m)}[n]$, and the Nyquist rate sequence $s[n]$ is given by:

$$s^{(m)}[n] = 0,\ldots,s[m-1],0,\ldots,0,s[M+m-1],0,\ldots,0,s[2M+m-1],0,0,\ldots.$$

Let us denote the discrete Fourier transform (DFT) of $s[n]$ and $s^{(m)}[n]$ by $S[k]$ and $S^{(m)}[k]$, respectively. By definition:

$$S[k] \triangleq \sum_{n=0}^{N-1} s[n]e^{-2\pi jkn/N}, \qquad 0 \le k \le N-1,$$

and similarly,

$$S^{(m)}[k] \triangleq \sum_{n=0}^{N-1} s^{(m)}[n]e^{-2\pi jkn/N}, \qquad 0 \le k \le N-1.$$

Since each coset $s^{(m)}[n]$ is a sub-Nyquist sample sequence of $s(t)$, $S^{(m)}[k]$ is an aliased version of $S[k]$. Proposition 1 shows the relation between $S[k]$ and $S^{(m)}[k]$.

Let the $M \times N$ matrix $\mathbf{S}$ represent the coset spectra, where the m^{th} row of $\mathbf{S}$ is equal to $S^{(m)}[k]$, and, $\mathbf{S}(m,n)$ corresponds to the n^{th} element of $S^{(m)}[k]$. In particular:

$$\begin{aligned}\mathbf{S}(m,n) &\triangleq S^{(m)}[n-1] \\ &= \sum_{r=0}^{N-1} s^{(m)}[r]e^{-j\frac{2\pi}{N}(n-1)r} \\ &= \sum_{r=0}^{L-1} S[rM+m-1]e^{-j\frac{2\pi}{N}(n-1)(rM+m-1)}, \end{aligned} \tag{4.2}$$

where $1 \leq m \leq M$ and $1 \leq n \leq N$.

Next, divide the Nyquist rate signal spectrum, $S[k]$, into M narrowbands (including both the positive side and the negative side of the spectrum) of equal bandwidth: L. Let the $M \times N$ matrix $\mathbf{B}$ represent shifted versions of the signal spectrum, $S[k]$, where each row of $\mathbf{B}$ is equal to the signal spectrum circularly shifted by $(m-1)L$ (i.e., $S[k + (m-1)L \bmod N]$), where $0 \leq k \leq N-1$. . The elements of $\mathbf{B}$ is then given by,

$$\begin{aligned}\mathbf{B}(m,n) &\triangleq \mathbf{B}^{(m)}[n] \\ &\triangleq S[(m-1)L + n - 1 \bmod N] \\ &= \sum_{r=0}^{N-1} s[r] e^{-j\frac{2\pi}{N}((m-1)L+n-1)r}.\end{aligned}$$

Proposition 1 (Multi-coset sampling)**.** *The coset spectra matrix* $\mathbf{S}$ *and the signal spectrum matrix* $\mathbf{B}$ *are related by:*

$$\mathbf{S} = \frac{1}{M}\mathbf{A}\mathbf{B}, \tag{4.3}$$

where $\mathbf{A}$ *is the* $M \times M$ *DFT matrix with* $\mathbf{A}(m,n) = e^{j\frac{2\pi}{M}(m-1)(n-1)}$.

Proof. Let $\mathbf{D}$ denote the product of $\mathbf{A}$ and $\mathbf{B}$. Then, the elements of $\mathbf{D}$ are given as:

$$\begin{aligned}
\mathbf{D}(m,n) &= \sum_{l=1}^{M} A_{ml} B_{ln} \\
&= \sum_{l=1}^{M} e^{j\frac{2\pi}{M}(m-1)(l-1)} \sum_{r=0}^{N-1} \mathbf{s}[r] e^{-j\frac{2\pi}{N}[(l-1)L+n-1]r} \\
&= \sum_{l'=0}^{M-1} e^{j\frac{2\pi}{M}} \sum_{r_2=0}^{M-1} \sum_{r_1=0}^{L-1} \mathbf{s}[r_1 M + r_2] e^{-j\frac{2\pi}{N}[l'L+n-1](r_1 M+r_2)} \\
&= \sum_{r_1=0}^{L-1} \sum_{r_2=0}^{M-1} \mathbf{s}[r_1 M + r_2] e^{-j\frac{2\pi}{N}(n-1)(r_1 M+r_2)} \sum_{l'=0}^{M-1} e^{j\frac{2\pi}{M}(m-1)l'} e^{-j\frac{2\pi}{M}(r_1 M+r_2)l'} \\
&= \sum_{r_1=0}^{L-1} \sum_{r_2=0}^{M-1} \mathbf{s}[r_1 M + r_2] e^{-j\frac{2\pi}{N}(n-1)(r_1 M+r_2)} \sum_{l'=0}^{M-1} e^{j\frac{2\pi}{M}(m-1)l'} e^{-j\frac{2\pi}{M}r_2 l'} e^{-j2\pi r_1 l'} \qquad (4.4) \\
&= M \sum_{r_1=0}^{L-1} \mathbf{s}[r_1 M + m - 1] e^{-j\frac{2\pi}{N}(n-1)(r_1 M+m-1)} \\
&= M\mathbf{S}
\end{aligned}$$

where Equation (4.4) follows because $e^{-j2\pi r_1 l'} = 1$, and

$$\sum_{l'=0}^{M-1} e^{j\frac{2\pi}{M}(m-1)l'} e^{-j\frac{2\pi}{M}r_2 l'} = \begin{cases} 0, & \text{when } r_2 \neq m-1, \\ M, & \text{when } r_2 = m-1. \end{cases} \qquad (4.5)$$

□

As shown in Proposition 1, the Nyquist rate signal spectrum $\mathbf{B}$ and the coset spectrum $\mathbf{S}$ are related by multiplication with $\mathbf{A}$. The DFT matrix $\mathbf{A}$ has dimension equal to the number of cosets M. In other words, the frequency content of the m^{th} coset, $S^{(m)}[k]$, is a weighted sum of the sub-bands of the original signal spectrum. The weightings are given by the m^{th} row of $\mathbf{A}$.

Let us now focus on the first sub-band (whose width is L) of the coset spectra, i.e., $S_{\text{b}}^{(m)} = S^{(m)}[n]$ for $0 \leq n \leq L-1$. We call this the baseband signal. Then, the baseband

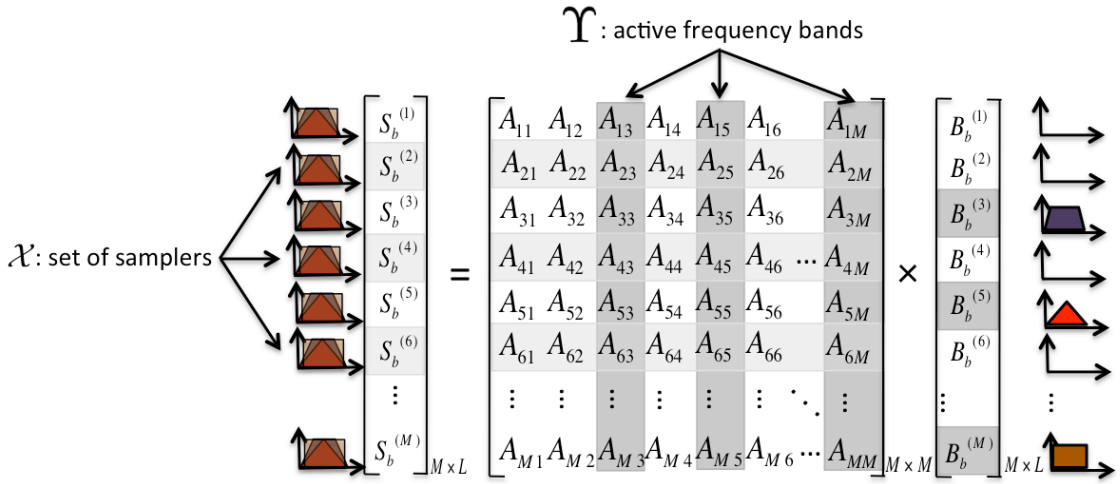


Figure 4-6: Relation between the coset spectrum and the spectrum of the multi-band signal.

signal has the same relation as (4.3). More specifically,

$$S_{\mathrm{b}}^{(m)} = \frac{1}{M}\mathbf{A}^{(m)}\mathbf{B}_{\mathrm{b}}, \tag{4.6}$$

where $A^{(m)}$ denotes the m^{th} row of $\mathbf{A}$ and $\mathbf{B}_{\mathrm{b}}(m,n) = \mathbf{B}(m,n)$ for $1 \leq m \leq M$ and $1 \leq n \leq L$ (i.e., individual sub-bands as shown in Figure 4-6). The relation in (4.6) holds for all cosets $1 \leq m \leq M$ and is illustrated in Figure 4-6. As shown on the right side of Figure 4-6, the m^{th} row of $\mathbf{B}_{\mathrm{b}}$ has width L and is equal to the m^{th} sub-band of $S[k]$ because by definition:

$$\begin{aligned}\mathbf{B}_{\mathrm{b}}^{(m)}[n] &= \mathbf{B}^{(m)}[m], \quad \text{for } 1 \leq n \leq L \\ &= S[(m-1)L + n - 1 \bmod N], \quad \text{for } 1 \leq n \leq L.\end{aligned}$$

As shown in Figures 4-6 and 4-7, since $s(t)$ is a multi-band signal that is sparse in the frequency domain and only a subset of the narrowbands are non-zero, $S_{\mathrm{b}}^{(m)}$ can be represented as a weighted sum of only the few non-zero bands.

More specifically, let $\mathrm{Y} \subseteq \{1, 2, \ldots, M\}$ denote the set of active (i.e., non-zero) narrowbands and let P denote the cardinality of Y (i.e., $P = |\mathrm{Y}|$). Let

$$\tilde{\mathbf{B}}_{\mathrm{b}} = \begin{bmatrix} B_{\mathrm{b}}^{(v_1)} \\ B_{\mathrm{b}}^{(v_2)} \\ \vdots \\ B_{\mathrm{b}}^{(v_P)} \end{bmatrix}_{P \times L}$$

denote the active narrowbands, where $v_i \in \mathrm{Y}, \forall i \in \{1, 2, \ldots, P\}$. Then, $S_{\mathrm{b}}^{(m)}$ is a weighted sum of only the non-zero sub-bands:

$$\begin{aligned} S_{\mathrm{b}}^{(m)} &= \frac{1}{M} A^{(m)} \mathbf{B}_{\mathrm{b}} \\ &= \frac{1}{M} \tilde{A}^{(m)} \tilde{\mathbf{B}}_{\mathrm{b}}, \end{aligned} \tag{4.7}$$

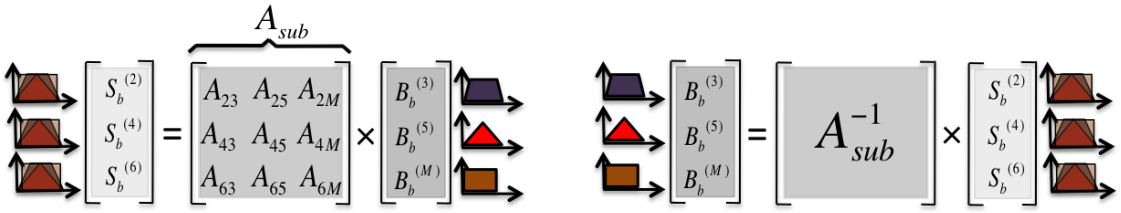
where $\tilde{A}^{(m)}$ is a sub-vector of $A^{(m)}$ and $\tilde{\mathbf{B}}_{\mathrm{b}}$ is a sub-matrix of $\mathbf{B}_{\mathrm{b}}$ such that only the elements corresponding to non-zero bands are included (i.e., $\tilde{A}^{(m)}[i] = A^{(m)}[v_i]$ and $\tilde{B}_{\mathrm{b}}^{(i)} = B_{\mathrm{b}}^{(v_i)}$ for $i \in \{1, 2, \ldots, P\}$).

Let $\mathcal{X} \subseteq \{1, 2, \ldots, M\}, |\mathcal{X}| \geq P$, denote the set of coset samplers (how they are selected is discussed in Section 4.2.2). Let

$$\tilde{\mathbf{S}}_{\mathrm{b}} = \begin{bmatrix} S_{\mathrm{b}}^{(\chi_1)} \\ S_{\mathrm{b}}^{(\chi_2)} \\ \vdots \\ S_{\mathrm{b}}^{(\chi_P)} \end{bmatrix}_{P \times L}$$

denote the outputs from the subset of active samplers, where $\chi_i \in \mathcal{X}, \forall i \in \{1, 2, \ldots, P\}$. Combining the narrowband selection Y and the coset selection $\mathcal{X}$, we get $\mathbf{A}_{\mathrm{sub}}$ such that $\mathbf{A}_{\mathrm{sub}}(m, n) = \mathbf{A}(\chi_m, v_n)$. In other words, $\mathbf{A}_{\mathrm{sub}}$ is a sub-matrix $\mathbf{A}$ that includes only the active narrowbands and the active cosets. Then, $\tilde{\mathbf{S}}_{\mathrm{b}}$ and $\tilde{\mathbf{B}}_{\mathrm{b}}$ are related by:

$$\tilde{\mathbf{S}}_{\mathrm{b}} = \frac{1}{M} \mathbf{A}_{\mathrm{sub}} \tilde{\mathbf{B}}_{\mathrm{b}}, \tag{4.8}$$



(a) Active band (b) Active band inversion

Figure 4-7: Relation between the coset spectrum and the spectrum of the active sub-bands of the multi-band signal.

and

$$\tilde{\mathbf{B}}_{\mathrm{b}} = M\mathbf{A}_{\mathrm{sub}}^{-1}\tilde{\mathbf{S}}_b. \tag{4.9}$$

The relations in (4.8) and (4.9) are illustrated in Figures 4-7(a) and 4-7(b), respectively. In the example in Figure 4-7(a), the sampler set is $\mathcal{X} = \{2, 4, 6\}$ and the active band set is $\mathrm{Y} = \{3, 5, M\}$. In this setting, $\mathbf{A}_{\mathrm{sub}}$ is a 3×3 matrix whose elements correspond to rows 2, 4, 6 and columns 3, 5, M of $\mathbf{A}_{M\times M}$. As long as this $\mathbf{A}_{\mathrm{sub}}$ is invertible, we can recover all spectral contents of the multi-band signal using as few as 3 cosets.

In summary, using the relation given in (4.9), the filtered signal spectrum can be recovered using as few as P of the M cosets, where $P \leq M$, is the number of occupied (non-zero) sub-bands. Since each coset sampler is running at $1/M$ of the Nyquist rate, the total sampling rate using multi-coset sampling is equal to P/M of the Nyquist rate and it is also equal to twice the total bandwidth of all non-zero narrowbands. This is the minimum rate required to sample a bandpass signal [43]. In the special case where the signal spectrum is full (i.e., there is no empty sub-band), we have $P = M$ and the multi-coset sampling rate is equal to the Nyquist sampling rate.

4.2.2 Coset sampler selection

Knowing the narrowband contents are theoretically reconstructible from multiple coset samples, how do we select which cosets to sample? Recall that the relation between the sub-band spectrum and the coset sample spectrum is given in (4.8), where the matrix $\mathbf{A}_{\mathrm{sub}}$ is a sub-matrix of the $M \times M$ DFT matrix. As shown in Figure 4-6, the columns

of $\mathbf{A}_{\text{sub}}$ correspond to the active narrow-bands, which may have been determined from the spectrum of the background noise or other design criteria. Then, given the column selections, we have the freedom to choose the sampler indexes. When there is no noise (neither background noise nor noise generated through the sampling process), any sampling sequences with an invertible $\mathbf{A}_{\text{sub}}$ will provide a re-construction of the narrowband contents. In fact, if we simply choose $\mathcal{X}$ to be the first P rows of the $M \times M$ DFT matrix (i.e., $\mathcal{X} = \{1, 2, \ldots, P\}$), then the sub-matrix $\mathbf{A}_{\text{sub}}$ is invertible for all possible column selections. We call this sampler $\mathcal{X}$ the bunched sampler and the invertibility property is shown in Lemma 4.1. More detailed studies of the bunched sampler can be found in [48, 49].

Lemma 4.1. *Let $\mathbf{A}$ be an $M \times M$ DFT matrix. Let $\mathbf{A}_{sub}$ be a square matrix of dimension P. The matrix $\mathbf{A}_{sub}$ is a sub-matrix of $\mathbf{A}$ such that $\mathbf{A}_{sub}(m, n) = \mathbf{A}(m, v_n)$, $v_n \in \mathrm{Y}$. Here, Y denotes the column selections such that $\mathrm{Y} \subset \{1, 2, \ldots, M\}$ and $|\mathrm{Y}| = P$. Then, $\mathbf{A}_{sub}$ is invertible.*

Proof. By construction, $\mathbf{A}_{\text{sub}}$ is a square Vandermonde matrix with the form:

$$\mathbf{A}_{\text{sub}} = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ e^{j\frac{2\pi}{M}(v_1-1)} & e^{j\frac{2\pi}{M}(v_2-1)} & \ldots & e^{j\frac{2\pi}{M}(v_n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ e^{j\frac{2\pi}{M}(P-1)(v_1-1)} & e^{j\frac{2\pi}{M}(P-1)(v_2-1)} & \ldots & e^{j\frac{2\pi}{M}(P-1)(v_n-1)} \end{bmatrix}_{P\times P}. \tag{4.10}$$

It follows from the property of square Vandermonde matrices [50] that the determinant of $\mathbf{A}_{\text{sub}}$ is given by:

$$\det(\mathbf{A}_{\text{sub}}) = \prod_{1\le m\le n\le P} (e^{j\frac{2\pi}{M}(v_m-1)} - e^{j\frac{2\pi}{M}(v_n-1)}) \tag{4.11}$$

$$\neq 0. \tag{4.12}$$

Eq. (4.12) follows because $e^{j\frac{2\pi}{M}(v_m-1)}$ are points taken from the M point complex unit circle and $v_m \le M$, $\forall v_m$. Since a square matrix is invertible if and only if its determinant is nonzero, $\mathbf{A}_{\text{sub}}$ is invertible. □

When there is noise, the selection of the samplers needs to be considered more carefully. Let $\mathbf{B}_{\text{clean}}$ denote the clean speech spectrum matrix after band-pass filtering (i.e., $\mathbf{B} = \mathbf{B}_{\text{clean}}$ when there is no noise). There are two types of noises to be considered: (1) ambient noise that is added to the speech signal before it is filtered and sampled; and (2) sampling noise that is introduced by that multi-coset feature extraction process.

Let $N_{\text{a}}[k]$ and $N_{\text{s}}[k]$ denote the spectra of ambient noise and sampling noise, respectively. Similar to how $\mathbf{B}$ is related to $S[k]$, let us define the $M \times N$ ambient noise spectrum matrix, $\mathbf{N}_{\text{a}}$, and the $M \times N$ sampling noise spectrum matrix, $\mathbf{N}_{\text{s}}$ as:

$$\mathbf{N}_{\text{a}}(m,n) \triangleq N_{\text{a}}[(m-1)L + n - 1 \bmod N],$$

and

$$\mathbf{N}_{\text{s}}(m,n) \triangleq N_{\text{s}}[(m-1)L + n - 1 \bmod N].$$

Note that, since $\mathbf{N}_{\text{s}}$ represents the noise introduced due to the multi-coset sampling processing (e.g., quantization noise, inaccuracies of BPF, etc), its spectrum generally depends on the input signal, the sampler set, $\mathcal{X}$, and the active narrowbands, Y. With

$$\mathbf{B} = \mathbf{B}_{\text{clean}} + \mathbf{N}_{\text{a}},$$

the relation in (4.3) becomes:

$$\begin{aligned} \mathbf{S} &= \frac{1}{M}\mathbf{A}(\mathbf{B}_{\text{clean}} + \mathbf{N}_{\text{a}}) + \mathbf{N}_{\text{s}} \\ &= \frac{1}{M}\mathbf{A}\mathbf{B} + \mathbf{N}_{\text{s}}. \end{aligned} \tag{4.13}$$

In this case, when we try to recover $\mathbf{B}$ from the coset samples, the noise due to sampling (i.e., $\mathbf{N}_{\text{s}}$), is also operated by the matrix $\mathbf{A}^{-1}$:

$$\begin{aligned} \mathbf{A}^{-1}\mathbf{S} &= \frac{1}{M}\mathbf{A}^{-1}\mathbf{A}(\mathbf{B}) + \mathbf{A}^{-1}\mathbf{N}_{\text{s}} \\ \mathbf{B} &= M\mathbf{A}^{-1}(\mathbf{S} - \mathbf{N}_{\text{s}}) \end{aligned} \tag{4.14}$$

Similarly, when only a subset of the full spectrum is active, we have:

$$\tilde{\mathbf{B}}_{\mathrm{b}} = M\mathbf{A}_{\mathrm{sub}}^{-1}(\tilde{\mathbf{S}}_{\mathrm{b}} - \tilde{\mathbf{N}}_{\mathrm{s}}). \tag{4.15}$$

What (4.15) tells us is that when only the ambient noise is under consideration (i.e., $\tilde{\mathbf{N}}_{\mathrm{s}} = 0$), Eq. (4.15) is equivalent to (4.9). In this case, we only need to select the sampler set such that $\mathbf{A}_{\mathrm{sub}}$ is invertible. On the other hand, when sampling noise is present, the statistics of the noise $\tilde{\mathbf{N}}_{\mathrm{s}}$ is projected by $\mathbf{A}_{\mathrm{sub}}^{-1}$. In order to avoid excessive noise amplification, a general rule of thumb is to choose the cosets such that the condition number of $\mathbf{A}_{\mathrm{sub}}^{-1}$, denoted by κ, is as small as possible. Since the condition number of any invertible matrix is equal to the condition number of its inverse, we just need to select $\mathcal{X}$ such that the condition number of $\mathbf{A}_{\mathrm{sub}}$ is kept small.

Recall that the filter band selection set Y corresponds to the column selections of $\mathbf{A}$ and it is selected based on characteristics of the speech signal and the noise spectrum. The sampler set $\mathcal{X}$ corresponds to the row selections of $\mathbf{A}$. In general, the method of choosing the row selection to yield low condition number is not obvious. It turns out that, with the evenly-spaced out filter-band selection scheme as proposed in Section 3.4, the bunched sampler selection yields the optimal condition number. This can be seen easily from the following lemma.

Lemma 4.2. *Let $\mathbf{A}$ be an $M \times M$ DFT matrix. Let $\mathbf{A}_{sub}$ be a square matrix of dimension $P \times P$. $\mathbf{A}_{sub}$ is a sub-matrix of $\mathbf{A}$ such that $\mathbf{A}_{sub}(m,n) = \mathbf{A}(m,(n-1)\beta+1)$, where β is an integer such that $\beta P = M$. Then, $\mathbf{A}_{sub}$ is a DFT matrix of dimension P.*

Proof. By definition, $\mathbf{A}(m,n) = e^{j\frac{2\pi}{M}(m-1)(n-1)}$. Then,

$$\begin{aligned}
\mathbf{A}_{\mathrm{sub}}(m,n) &= e^{j\frac{2\pi}{M}(m-1)(((n-1)\beta+1)-1)} \\
&= e^{\frac{2\pi}{M}(m-1)(((n-1)\beta+1)-1)} \\
&= e^{\frac{2\pi}{M}(m-1)((n-1)(M/P))} \\
&= e^{\frac{2\pi}{P}(m-1)(n-1)}.
\end{aligned} \tag{4.16}$$

Hence, $\mathbf{A}_{\text{sub}}$ is a DFT matrix. Since the condition number of any orthogonal matrix is 1, $\mathbf{A}_{\text{sub}}$ yields the minimum condition number. □

In contrast, when Y is not chosen to be evenly-spaced out across the frequency spectrum, the bunched samplers may result in a $\mathbf{A}_{\text{sub}}$ that is ill-conditioned. This may happen when the ambient noise is strong. Heavily polluted narrowbands are discarded, which breaks the evenly spaced structure of Y. If the sampling noise, $\mathbf{N}_{\text{s}}$, is not negligible, a different set of samplers needs to be chosen such that $\mathbf{A}_{\text{sub}}$ is well-conditioned.

The optimal $\mathcal{X}$ given a specific Y can be found by an exhaustive search. Such a search at run time is impractical for our low-power system design, while the storage of a pre-computed table of $\mathcal{X}$ for each Y may also be impractical due to table size. An alternative is to use the method of co-array sampler selection [51]. Given the matrix dimension, M, and the sampler cardinality, P, the co-array method finds $\mathcal{X}$ that yields close to the minimum worst case condition number among methods universal to Y selections of cardinality P.

The co-array algorithm is a generalization of the minimum redundancy linear array algorithm (MRLA) [52] and $\mathcal{X}$ is selected in a way that promotes different spacings for sampler pairs.

Appendix C provides a comparison of three sampler selection schemes: optimal sampler selection through exhaustive search, co-array sampler selection, and the bunched sampler selection. It is shown that the co-array method yields a much better condition number than the bunched sampler approach under the worst-case Y selection. In practice, the co-array samplers can be selected off-line per each P value and stored in a look-up table for real-time processing [1].

4.3 Feature extraction front-end implementation

In this section, we describe the system implementation details and steps taken to simplify the feature extraction procedure.

[1] The reconstituted signal can be expressed as the summation of three components: $\mathbf{B}_{\text{clean}} + \mathbf{N}_{\text{a}} + M\mathbf{A}^{-1}\mathbf{N}_{\text{s}}$. In our experiments, due to high quantization accuracy and sharp front-end filter design, the sampling noise, $\mathbf{N}_{\text{s}}$, is negligible compared to the ambient noise, $\mathbf{N}_{\text{a}}$, under noisy conditions. The optimal and the bunched sampler selection schemes yielded comparable recognition accuracy. In our experiments, the samplers are chosen based on the bunched scheme. In practical system designs, the co-array sampler selection scheme may be adopted to avoid excessive amplification on $\mathbf{N}_{\text{s}}$

4.3.1 Multi-coset sampling and reconstruction

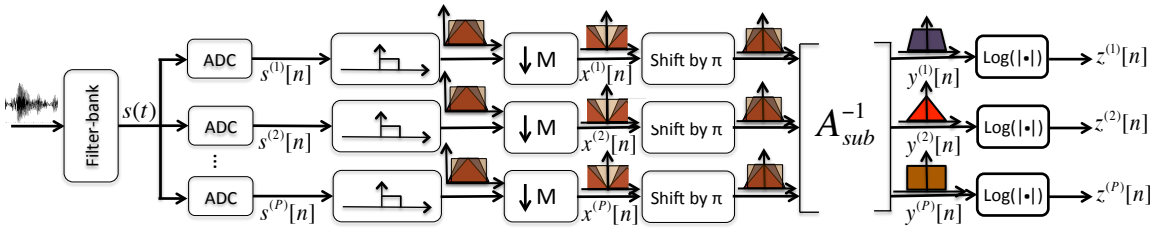
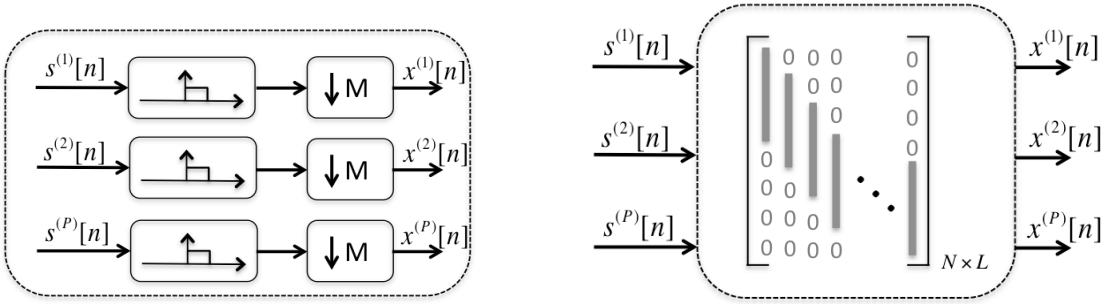


Figure 4-8: Sampling the speech signal $s(t)$ with multi-coset sampling. The number of cosets $M = 7$. The horizontal axis labels correspond to the coset index of each sample. The samples with the same color belong to the same coset.

In section 4.2.1, we looked into the mathematical relationship between the multi-band signal, $s(t)$ and its coset samples, $s^{(m)}[n]$. Now, we discuss the digital circuit realization of combined sampling and feature reconstruction. Figure 4-8 shows the flow diagram of the feature extraction process. After band-pass filtering, the multi-band signal is sampled with P low-rate samplers, which gives us the coset samples $s^{(\chi_i)}[n]$, for $\chi_i \in \mathcal{X}$ (i.e., Eq. (4.2)). We digitally filter each coset sample sequence with a shifted low-pass filter whose pass-band is designed to be between 0 and $(2\pi)/M$. As explained in Section 4.2.1, each coset at the output of the baseband low-pass filter is an aliased version of all the signal sub-bands (illustrated in Figure 4-8). We down-sample the signal by M to obtain the base-band samples. The resulting signal is a weighted sum of the sub-band signals shifted by π. We shift the signal by π in the frequency domain so the signal is properly centered and multiply the sequence by the inverse of the matrix $\mathbf{A}_{\text{sub}}$. The output of the inversion matrix corresponds to the signal in each narrow sub-band, which is now shifted to the baseband and down-sampled (e.g., outputs of $\mathbf{A}_{\text{sub}}$ in Figure 4-8). Lastly, we take the logarithm of the signal amplitudes to obtain the acoustic features for downstream processing. This feature extraction procedure follows directly from the relationship given in (4.9). We next show that this process can be simplified by combining digital low-pass filtering with sub-sequent down-sampling and by removing the frequency shifting step.

Figure 4-9(a) shows the low-pass filtering and down-sampling sub-components of the system in Figure 4-8. Recall that filtering can be expressed as convolution between the signal and the filter impulse response. To illustrate the computation complexity involved



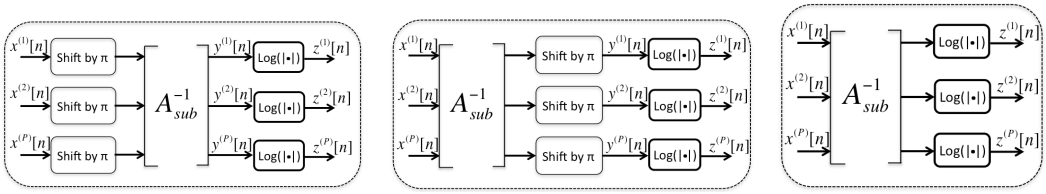
(a) Separate filtering and downsampling

(b) Filtering and downsampling combined

Figure 4-9: Filtering and down-sampling the coset samples: the filtering and down-sampling steps are combined to reduce computation complexity.

in the digital filtering procedure, we then represent the convolution operation between a segment of speech samples and the filter taps with matrix multiplication. The input speech signal has length N. It is multiplied with a filter matrix, where each column of the matrix contains the taps of the filter positioned to operate on a windowed segment of the speech. All the entries outside the window are zeros. In other words, filtering a speech segment with length N can be viewed as multiplication with an $N \times N$ filtering matrix where the number of non-zero entries in each column is equal to the length of the filter taps (i.e., N_{taps}). Hence, the total number of non-zero multiplications is equal to $N \times N_{\text{taps}}$. Notice that low-pass filtering is immediately followed with the down-sampling step. Hence, we can simplify the overall complexity of the system in Figure 4-9(a) by avoid computing the 'omitted' values in the first place. This is equivalent to removing the corresponding columns in the filtering matrix. Therefore, as shown in Figure 4-9(b), the simplified filtering and sampling procedure is equivalent to matrix multiplication with a $N \times L$ matrix. The complexity of this component is then reduced to $L \times N_{\text{taps}}$ multiplications for each coset vector.

As shown in Figure 4-10(a), baseband digital filtering and down-sampling are followed by the coset inversion procedure to recover the narrowband features. Figure 4-10 shows three equivalent systems that illustrate the steps taken to simplify the inversion process. As shown in Figure 4-10(a), the process begins with shifting the signal by π in the frequency domain, which is equivalent to multiplying the signal vector with a diagonal filter matrix (i.e., $\text{diag}(1, -1, 1, -1, \ldots)$). Since diagonal matrices commute with



(a) Coset sample inversion (b) Equivalent inversion system (c) Simplified system

Figure 4-10: This figure illustrated the procedure for extracting the narrowband contents from the coset samples. (a) shows the original process. By exchanging the matrix inversion step and the spectrum shifting step, which can be considered as multiplication of a diagonal matrix, we get an equivalent system, shown in (b). Shifting in the frequency domain does not affect the amplitude of the signal. Hence, the system in (b) can be simplified to the system shown in (c).

square matrices of the same size, the frequency shift step and the inversion step can be exchanged to arrive at the equivalent system shown in Figure 4-10(b). Then, the subsequent module computes the logarithm of the input signal amplitude. Since shifting the spectral signal by π is equivalent to element-wise multiplying the time-domain signal by the vector $[1, -1, 1, -1, \ldots]$, which does not affect the absolute signal amplitude, the phase shifting step can be omitted without affecting the final output of the system. Figure 4-10(c) shows the simplified coset inversion process.

The computation complexity of the inversion process can be estimated with the the complexity of multiplication with $\mathbf{A}_{\text{sub}}$. The matrix $\mathbf{A}_{\text{sub}}$ is a $P \times P$ sub-matrix of the $M \times M$ DFT matrix. Hence, the inversion matrix $\mathbf{A}_{\text{sub}}^{-1}$ also has dimension P and the inversion procedure involves $L \times P^2$ multiplications. It is important to note that, due to spectral symmetry of the real speech signal, we only need to recover either the positive or negative portion of the signal spectrum in practical implementations. In other words, the system will only reconstruct $P/2$ sub-bands and the computation complexity is thus $L \times P \times (P/2)$.

Combining the sub-systems in Figures 4-9(b) and 4-10(c), the feature extraction front-end in Figure 4-8 is simplified to the system shown in Figure 4-11. The number of ADCs is equal to the number of active narrowbands, P (including both positive and negative frequencies). Let K denote the number of bandpass filters required in the front-end filterbank. Then, $K = P/2$.

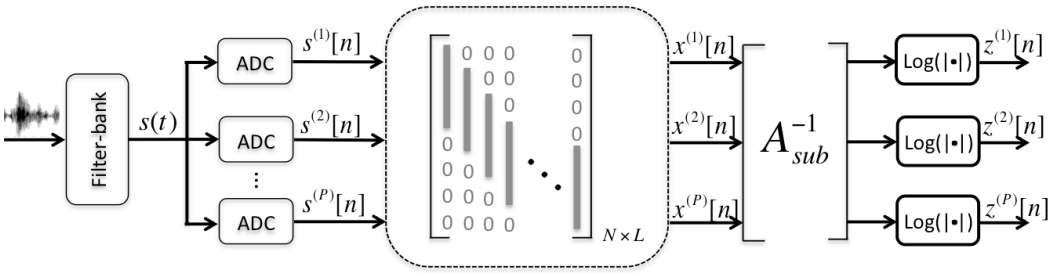


Figure 4-11: Block diagram of the simplified multi-coset feature extraction system.

4.3.2 Analog and digital filter design

Both analog and digital filter design choices affect the front-end performance. The fall-off rate of the bandpass filter bank directly affects the quality of the multi-coset reconstruction of the narrowband signals. The sharper the filter fall-off, the less signal distortion there is on the narrowband content and the less aliasing there will be from the multi-coset reconstruction process. Section A.1 shows the multi-coset reconstruction of the narrowband signals when bandpass filters with different fall-off rates are used. In these examples, the spectrum of the original speech signal, which has a bandwidth of 8kHz is divided into narrowbands and only 3 narrowbands are retained (6 bands if both the positive and negative spectra are included) by passing through the bandpass filterbanks. We can see that, the sharper the bandpass filters, the less distortion there is between the original narrowband signals and the reconstructed signals.

In a similar manner, the digital baseband low-pass filter following the ADCs (as shown in Figure 4-8) also affects the the quality of the multi-coset reconstruction. The longer the number of taps of the low-pass filter, the shaper the fall-off rate and less aliasing there will be after the signal is reconstituted. The examples in Section A.2 illustrate the effects of using digital low-pass filters with different number of taps.

In our simulations, we have implemented filterbanks with bandwidths equal to 200Hz and 400Hz. In these implementations, the filters have a 3dB cut-off at 50% of their bandwidth. The digital baseband low-pass filter is a 100 order finite-impulse-response (FIR) filter with Kaiser window. The frequency domain magnitude responses of the bandpass filters and the low-pass filter are given in Appendix B.

4.3.3 Computation complexity and power estimation

As shown in Figure 4-11, the computation complexity of the multi-coset reconstruction procedure is proportional to the number of sub-bands K. Recall L denotes the bandwidth of the narrow sub-bands, then the rate of incoming signal from each sub-band is L samples per second. Let N_{taps} denote the number of taps of the baseband low-pass filter. Then, the number of operations of multi-coset signal reconstruction is given in Table 4.1.

Table 4.1: Computation complexity of multi-coset feature extraction. K is the number of active narrowbands. L is the bandwidth of the narrowbands. N_{taps} is the order of the baseband lowpass filter.

	Filter/ down-sample		**Inversion**
multiplications (real $\times$ complex)	$2K \times N_{\text{taps}} \times L$	**multiplications (complex $\times$ complex)**	$2K \times K \times L$
complex additions	$2K \times N_{\text{taps}} \times L$	**complex additions**	$2K \times K \times L$
total operations	$8 \times K \times N_{\text{taps}} \times L$	**total operations**	$16 \times K^2 \times L$

4.4 Summary

In this chapter, we proposed a method to sample and extract the narrowband features using the technique of multi-coset sampling. We presented a detailed implementation of the feature extraction front-end that includes an analog filterbank, a set of low-rate samplers and an additional digital processing module. Unlike the conventional multi-band feature extraction methods, the proposed implementation does not require additional analog components after band-pass filtering. The low-rate coset samplers sample directly on the post-filtering analog signal at its minimal sampling rate. Narrowband features are reconstructed from the low-rate samples through digital processing.

Chapter 5

Text-dependent speaker verification

As shown in Figure 2-3, a voice-command recognition system can be decomposed into a pipeline of two components: a feature extraction front-end and a recognition back-end. In Chapters 3 and 4, we proposed an adaptive feature extraction front-end that aims to optimize system efficiency by varying computation complexity based on the input condition. Over the next two chapters, we develop speech recognition algorithms that support adaptive band selection and achieve low-power consumption for the applications of speaker-verification (SV) and user-independent command recognition.

First, we focus on the text-dependent SV problem. Existing SV methods have shortcomings relating to power consumption and noise susceptibility. An ideal SV system for such applications requires a combination of security, low power usage, noise resiliency, and customized passphrases. In consideration of these constraints, we develop a novel text-dependent SV system in which the user defines his or her own short passphrase ($<$ 1s in duration) by enrolling a small number of samples. The recognition algorithm aims at identifying the uttered command and the speaker in one shot (i.e., decision is positive only when the authentication command is uttered by the designated speaker). The main challenge lies within the limited amount of training samples. Unlike a user-independent command recognition system, whose parameters can be trained off-line with thousands of training samples, the user-dependent command recognition system learns to recognize a command with only 3 to 5 enrollment samples. In this chapter, we propose a

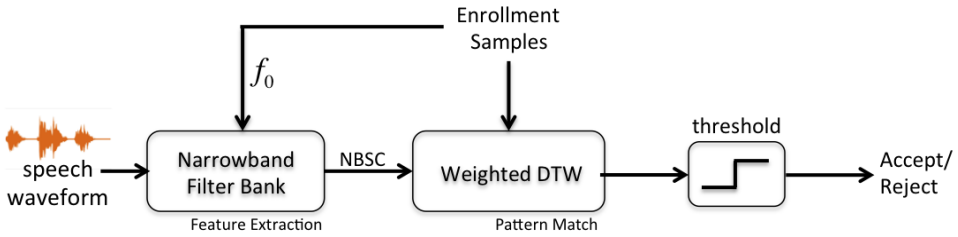


Figure 5-1: Block diagram of our proposed system including the feature extraction front-end, which consists of K ($\sim$ 10) narrowband filters with fixed bandwidth ($\sim$ 300Hz) centered around multiples of f_0 (estimated from enrollments). All or a subset of the K features are used for decision making depending on the background noise spectrum. The back-end is a weighted-DTW algorithm, in which the adaptive warping constraint is inversely proportional to the temporal signal energy.

low-power, text-dependent SV system comprising a NBSC feature extraction front-end and a back-end running an improved dynamic time warping (DTW) algorithm.

We review the background of the SV problem in Section 5.1. Next, we introduce the user-dependent narrowband feature extraction scheme in Section 5.2 and the weighted-DTW algorithm in Section 5.3. In Section 5.4, we compare our system performance with the conventional constrained DTW with MFCC features approach and with the widely used fixed-text SV method based on Gaussian mixture universal background models (GMM-UBM).

5.1 Background on speaker-verification systems

Existing techniques for SV can be 'text-independent' [53,54] or 'text-dependent' [55]. Text-independent SV has the flexibility to recognize a speaker's identity without constraints on the speech (i.e., any word can be uttered during enrollment and testing). However, it usually requires a large amount of speaker-specific enrollment data (typically more than 30s) to extract sufficient useful features to discriminate between speakers. A performance penalty is paid for the high degree of variability in speech contents. On the other hand, text-dependent SV assumes the utterances being tested are the same as, or a subset of, the enrollment lexicon. Therefore, a more specialized model can be built, achieving better accuracy using shorter enrollment (usually less than 8s). Our applications falls into the category of text-dependent SV.

A successful technique in SV is to leverage speech across a cohort of speakers to train a background model as a prior, which is then used to make speaker-specific refinements, see e.g. methods based on GMM-UBM [56, 57], i-vectors [58], DNNs [59, 60] and HMMs [61,62]. Since these methods require background model training on *a priori* known passphrases, it is not suitable for our application due to lack of training data besides the few samples of user enrollment.

Our system solves SV as a pattern matching problem based on similarity measures between the input signal and the enrollment samples directly. As shown in Figure 5-1, the process includes two stages: feature extraction and pattern matching on features. We develop novel designs in both stages and describe them in Sections 5.2 and 5.3, respectively.

5.2 Narrowband features for text-dependent speaker-verification

As discussed in Chapter 3, in speech recognition applications, the MFCCs [9, 63] are widely used and have yielded good performance. Nevertheless, the extraction process usually involves fast sampling, a large number of filters (26 to 40) and high-rate processing, that are associated with high computation and power costs. We proposed a low-complexity, power-efficient feature extraction front-end that completes feature extraction in two simple steps: (1) filtering the analog speech signal using a handful of ($\sim$ 10) fixed-width narrowband filters, whose center frequencies are chosen according to the fundamental frequency f_0 estimated from enrollments; and (2) taking the logarithm of the filterbank power. This approach offers the benefits of low-power implementation, high verification accuracy and noise robustness by automatically discarding features with high noise occupancy. The low-dimensionality of the features also reduces the back-end computation since the complexity of the back-end SV algorithm is proportional to the feature dimension. The detailed feature extraction procedure is given in Appendix D.

5.3 Dynamic time-warping algorithms

Speech pattern matching is often performed with DTW [64, 65]. Variations of the DTW algorithm are developed to constrain the warping path [65, 66], add weightings to the feature vector based on the intraspeaker variability for each feature element [64] or add weighting based on temporal characteristics of the warping path [65]. One common issue associated with applying these methods to our application is they either apply too much warping that distorts the signal characteristics or insufficient warping to compensate for the long pauses between words. We propose a modified version of the DTW algorithm that adaptively adjusts warping constraints based on the signal's total energy envelope, thus restricting excessive distortion on the main signal envelope while still allowing sufficient time warping to take care of long pauses between words and speaking rate variations.

5.3.1 A review: classical DTW

We start by describing the classical DTW algorithm. The back-end for non-parametric SV operates by comparing features of an input utterance with features from each of the enrollment samples according to a similarity measure. Under the simplest decision rule, the minimum of all distances under the measure is used to make the final verification decision.

Let the enrollment signal, R, and the input signal, T, each represent a sequence of feature vectors,

$$R = [R(1), R(2), \ldots, R(i), \ldots, R(I)];$$
$$T = [T(1), T(2), \ldots, T(j), \ldots, T(J)];$$

where $R(i)$ and $T(j)$ are feature vectors with dimension K, and I and J are the number of temporal frames in R and T, respectively. The enrollment sample R is generated by the target speaker. We would like to measure the similarity between R and T to determine whether T is generated by the same target speaker. Due to temporal variations such as speaking speed differences and pauses in the speech utterance (e.g., pauses between

words), the similarity between the input features and the enrollment features cannot be directly compared frame-by-frame. Standard algorithms such as the DTW algorithm [8, 64,65] are designed to mitigate the problem of signal misalignment by applying a warping function coupling two sequences so that they can be directly compared. The warping function, W, can be represented as a sequence of index pairs that provide a mapping between the frames of R and T. More specifically,

$$W = [W(1), W(2), \ldots, W(m), \ldots, W(M)],$$

where $W(m) = (i(m), j(m))$, and i and j are warping indexes corresponding to R and T, respectively. Given a warping path W, M corresponds to the length of the path.

The warping function W is chosen to minimize the accumulated distance along the path,

$$D_{\text{total}} = \min_{W} \sum_{m=1}^{M} \text{dist}(R(i(m)), T(j(m))), \tag{5.1}$$

where W needs to satisfy the following conditions:

1. Monotonicity:
   $i(m-1) \leq i(m)$ and $j(m-1) \leq j(m)$;
2. Continuity:
   $i(m) - i(m-1) \leq 1$ and $j(m) - j(m-1) \leq 1$;
3. Boundary conditions:
   $i(1) = j(1) = 1$, $i(M) = I$ and $j(M) = J$.

The following warping window length, m_0, can be additionally applied to reduce computation and constrain excessive warping [65]:

4. Warping window (Sakoe-Chuba) constraint : $|i(m) - j(m)| \leq m_0$.

As shown in Figure 5-2, if we lay the reference signal, R, along the horizontal axis (i-axis) and the input signal, T, along the vertical axis (j-axis), the warping function forms a path on the i-j plane.

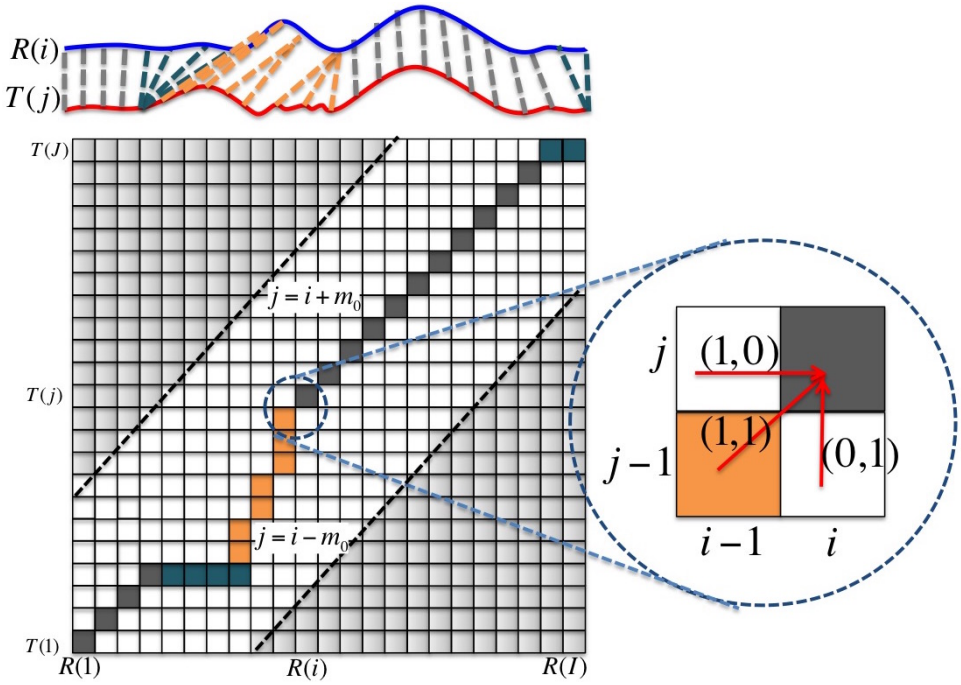


Figure 5-2: Illustration of the DTW algorithm. The warping path is represented by the highlighted line. The warping window (window length m_0) is represented by the unshaded area. At each point, there are three candidate movements: $(1,0)$, $(1,1)$ and $(0,1)$.

Due to the monotonicity and continuity conditions, two consecutive points on the warping path can only be connected by three candidate movements:

$$W(m) = \begin{cases} W(m-1) + (0,1), & \text{move up} \\ W(m-1) + (1,1), & \text{diagonal} \\ W(m-1) + (1,0). & \text{move right} \end{cases} \tag{5.2}$$

The optimal warping path minimizing (5.1), can be obtained using dynamic programming. First, an accumulative distance matrix $D_{I\times J}$ is created. Each entry $D(i,j)$ represents the accumulative distance between the partial sequences $R(1),...,R(i)$ and $T(1),...,T(j)$ along the warping path up to point $(R(i),T(j))$. Due to the boundary condition, the path starts at $(1,1)$ and ends at (I,J), and the warping path is found by filling up the accumulative distance matrix D column by column and traversing back the matrix along the

entries that yielded the minimum overall distance. More specifically:

$$\begin{aligned} D(i,1) &= \text{dist}(R(i),T(1)), \\ D(1,j) &= \text{dist}(R(1),T(j)), \\ D(i,j) &= \text{dist}(R(i),T(j)) + \min\{D(i-1,j), \\ &\qquad D(i-1,j-1), D(i,j-1)\}. \end{aligned} \tag{5.3}$$

Implementation details of this classical DTW algorithm can be found in [65]. Subsequent variations of the DTW algorithm have also been developed to add constraints to the warping function [65, 66], add weighting to the temporal envelope [65], or to add weighting to the feature vectors [64]. To improve the accuracy of word segmentation, a silence model can be incorporated into the DTW algorithm to detect pauses between words [67]. Even though these efforts have demonstrated improved performance compared to the classical DTW algorithm, they do not address specific issues pertaining to the problem of SV. Next, we propose a weighted-DTW algorithm, which is a modified version of the classical DTW algorithm and is designed specifically for our application of SV.

5.3.2 Weighted-DTW

For our SV application, the passphrase is defined by the user and could contain long gaps between words. The major challenge associated with using the classical DTW algorithm for our SV application is how to apply sufficient warping to realign the words while still preserving the temporal characteristics of the signal. The classical DTW algorithm and its variations do not address this issue properly. If too much warping is allowed (e.g., m_0 is large), the warping process often results in excessive signal mutation such that details of the signal characteristics are lost, which results in a large number of false-positive decisions. On the other hand, if the warping constraints are too strict (e.g., m_0 is small), it results in insufficient warping to take care of the long pauses between words and thus results in mis-detections.

In order to overcome this issue, our modified version of the DTW algorithm penalizes excessive warping according to the following factors:

- the penalty scales linearly with the number of consecutive warps of the same type (i.e., 'move up', 'diagonal' or 'move right');
- the penalty scales linearly with the amplitude of the total power envelope
  - more penalty when the signal amplitude is high in order to retain the shape of the temporal envelope;
  - less penalty when the signal amplitude is low, which is an indication of possible pauses.

More specifically, the warping function is found as follows: we define a movement matrix M ($M \in \{(1,0),(0,0),(0,1)\}^{I \times J}$) that records the type of movement taken to arrive at each point (i,j). We then define a step counter matrix C ($C \in \mathbb{N}^{I \times J}$) that records the number of accumulative same-type movement to arrive at each point. For example, if a path takes three consecutive horizontal steps (i.e., $(1,0)$) to arrive at (i,j), then $C(i,j) = 3$. The counter restarts whenever the previous step and the current step are not the same type. In order to limit mutation to the signal envelope, at each step, we use the total energy of the two signals (E_R and E_T) as a weighting function to determine the penalty of taking a certain step. So

$$D(i,j) = \mathrm{dist}(R(i), T(j)) + \min_S \{D((i,j) - S) + P((i,j), S)\}, \tag{5.4}$$

where

$$S \in \{(1,0),(1,1),(0,1)\},$$

and

$$P((i,j),S) = \mathbb{1}\{M(i-1,j) = S\}C(i-1,j)|\mathrm{E}_T(j)| + \mathbb{1}\{M(i,j-1) = S\}C(i,j-1)|\mathrm{E}_R(i)|.$$

Eq. (5.4) replaces (5.3) of the conventional DTW algorithm. To save computation, we use the L_1 norm as our distance measure and normalize it over the feature dimension K:

$$\text{dist}(R(i), T(j)) = \frac{1}{K} \sum_{k=1}^{K} |R(i)[k] - T(j)[k]|. \tag{5.5}$$

The matrices M and C are initialized with

$$M(1,1) = (0,0) \quad \text{and} \quad C(1,1) = 0;$$

and are updated with the S^* that yields the minimum $D(i,j)$ (Eq. (5.4)) at each step:

$$\begin{aligned} M(i,j) &= S^*, \\ C(i,j) &= (C((i,j) - S^*) + 1)\mathbb{1}\{C((i,j) - S^*) = S^*\}. \end{aligned}$$

Without the penalty term in (5.4), the weighted-DTW algorithm would yield the same path as the classical DTW algorithm.

For the classical DTW algorithm, the distance between R and T is equal to $D(I,J)$. That is not the case for the weighted DTW algorithm due to the additional penalty term. The final similarity measure between R and T is re-computed after obtaining the warping path. We also normalize the total distance such that the average distance is not biased by the warping path length. So,

$$D_{\text{norm}} = \frac{1}{M} \sum_{m=1}^{M} \text{dist}(R(i(m)), T(j(m))). \tag{5.6}$$

The Pseudo-code of the weighted-DTW algorithm is given in Appendix E.

5.3.3 Simulation: comparison between the weighted-DTW and the classical constrained DTW algorithm

Recall the task of SV is to identify whether an input signal corresponds to the passphrase uttered by the designated speaker. Instead of comparing the input signal and the reference signal directly, we first apply warping to align the bulk of signal envelopes to

overcome the problems of signal envelope misalignment and speech rate variation. The challenge is to align the signals without mutating the shapes of the signal envelopes.

Figures 5-3 and 5-5 show the simulation results of the weighted-DTW algorithm compared with results from the classical DTW and constrained DTW algorithms. The simulations demonstrate that the weighted-DTW algorithm is capable of applying sufficiently large amounts of warping in the case of misalignments, while refraining from excessively distorting the signal envelope. On the other hand, existing methods fail to align the signal envelopes when the warping constraint is tight and results in signal envelope distortion when the warping constraint is loose.

In each plot, the red curve corresponds to the reference signal, R, that we are comparing against (i.e., the targeting command) and the blue curve corresponds to the input signal, T.

In Figure 5-3, the input signal is the power envelope of a speech command that is the same as the reference signal command, whereas, in Figure 5-4, the input signal is a different command from the reference speech command. As we can see from the first plot of Figure 5-3, the overall envelope of the reference and the input signals are quite similar. However, the starting points of the reference signal and the input signal are misaligned, resulting in a large distance of 0.2 between the two signals. After warping, the classical DTW algorithm, the constrained DTW with 200ms warping window and the weighted-DTW algorithm are all able to align the two signals properly, hence achieving a small distance of less than 0.05. However, with stricter constraints on the warping window width, the constrained DTW with a 100ms warping widow scheme failed to properly align the two signals.

While the narrow 100ms warping window fails to align the signals in Figure 5-3, it provides sufficient distortion to the input signal in Figure 5-4 such that the distance between the warped input signal and the reference signal is reduced to less than 0.052. In fact, all the warping algorithms except the weighted-DTW algorithm excessively distorts the input signal and results in a close distance between the two different speech commands. In contrast, applying weighted-DTW retains the overall shape of the input signal and the

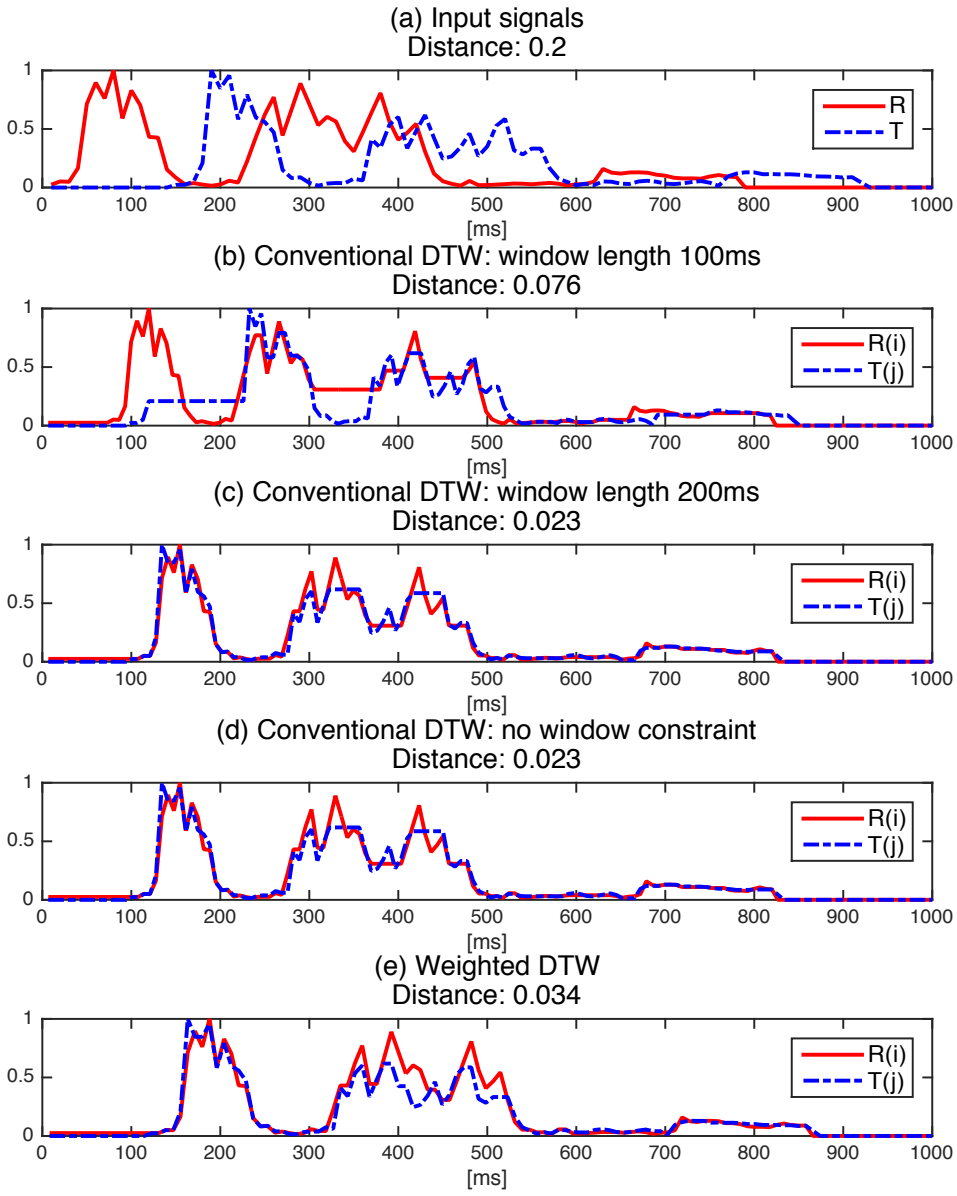


Figure 5-3: Weighted-DTW simulation with starting point mis-alignments: signal warping is performed with the unconstrained DTW, constrained DTW and weighted-DTW. Input signal corresponds to the same speech command as the reference signal. Unlike the conventional DTW algorithm that either fails to align the two signals (e.g., (b)) or causes signal envelope mutation (e.g., (c), (d)), the weighted-DTW properly aligns the two signals without causing envelope mutation (e.g., (e)).

distance between the two signals remains large. Note that, in both Figures 5-3 and 5-4, the shapes of the input signal envelopes are best preserved by the weighted-DTW algorithm.

Figure 5-5 provides an example of excessive signal mutation of the conventional DTW algorithm. In this case, the input signal, T, is a completely random sequence. As the

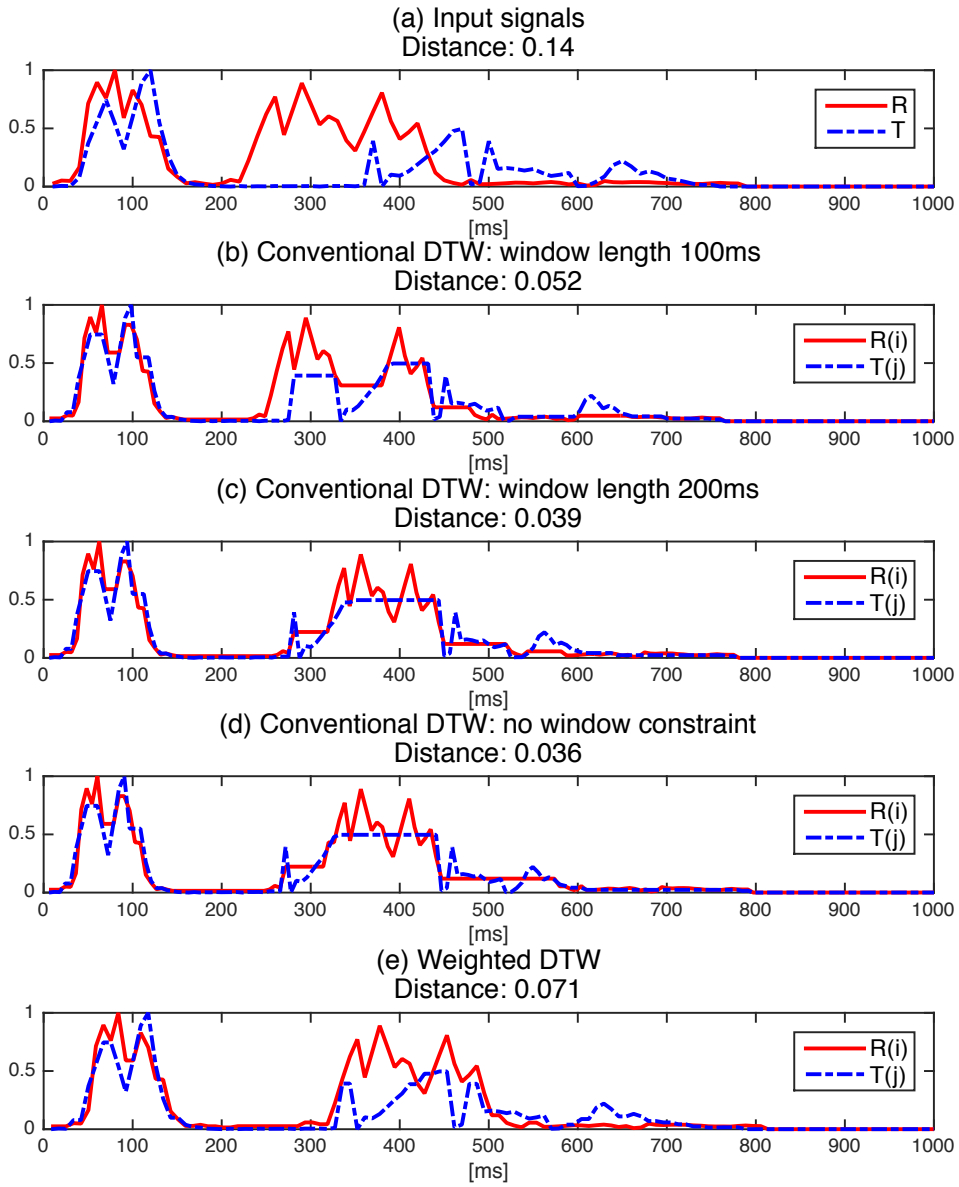


Figure 5-4: Warping of two signals R and T. The input, T, is a different utterance than the enrollment, R. (a) Two signals R and T before warping. There is a long pause in the signal T. (b) With window length 100ms, the classical DTW fails to realign the envelopes of the two signals. (c) With window length 200ms, even though the main bulk of the two signals are aligned, the temporal envelope of T is heavily mutated. (d) The weighted-DTW algorithm properly aligns the main bulk of the signals without excessive mutation on the shape of the signal envelopes.

window width becomes larger, the random sequence starts to lose its original shape and becomes similar to the reference signal. In contrast, the shape of the random sequence is retained using the weighted-DTW algorithm.

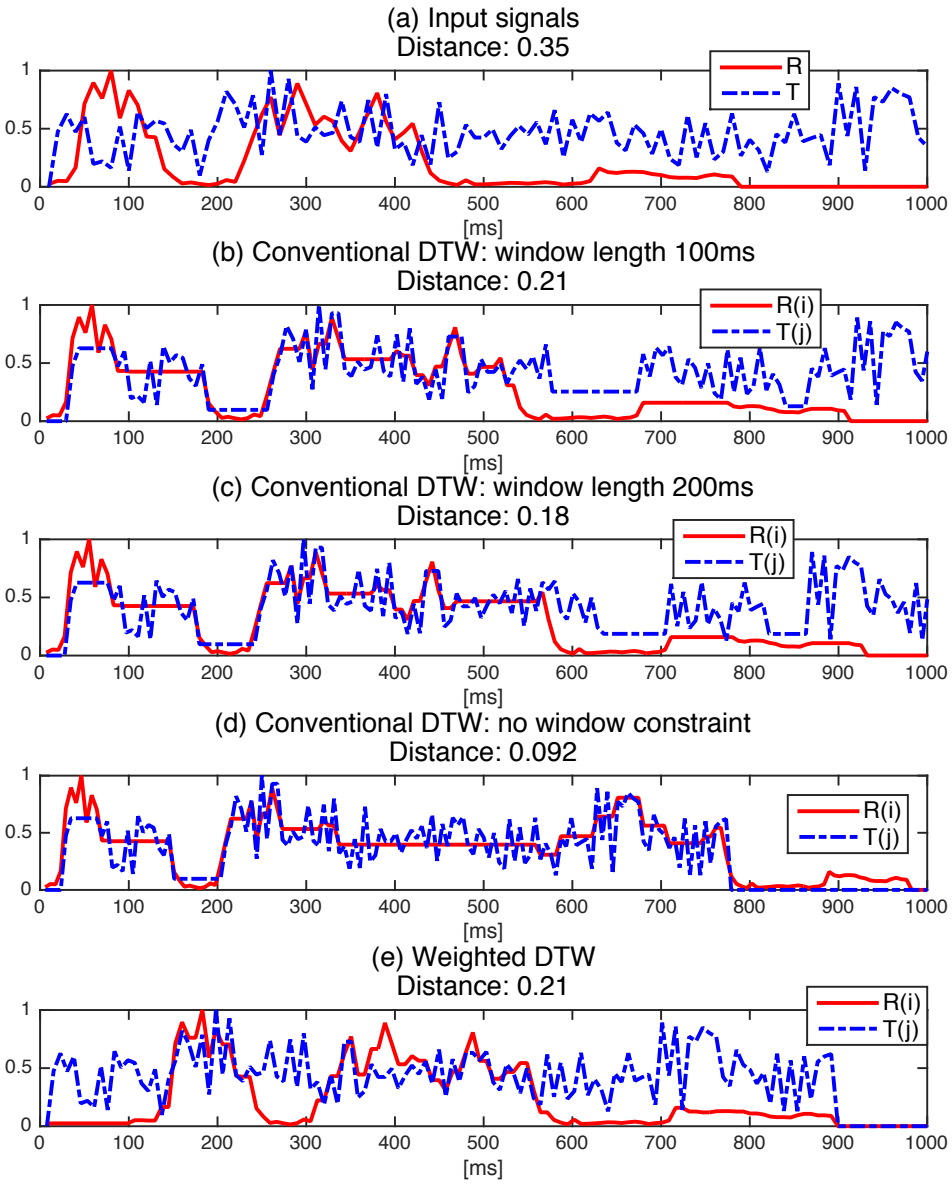


Figure 5-5: Weighted-DTW simulation with random input: the input signal is a random sequence. The conventional DTW algorithms mutates the signal envelopes such that the distance between the reference signal and the random input becomes very small (e.g., (d)). The weighted-DTW retains the shape of the input signal and the distance between the two signals remains large after warping.

5.3.4 Blockwise weighted-DTW

Recall the optimal warping path is obtained by first computing the accumulative distance matrix and then traversing back the shortest distance path. The algorithm memory requirement is thus proportional to the size of the D matrix, which is $\mathcal{O}(IJ)$. When we

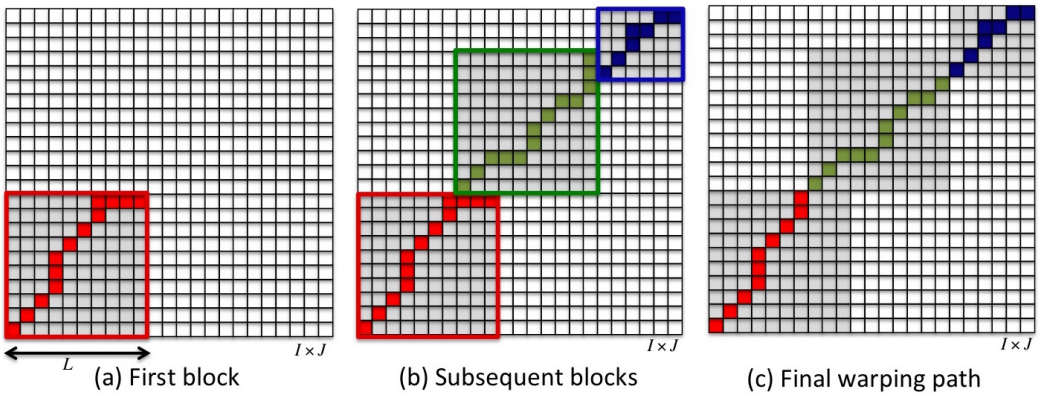


Figure 5-6: Blockwise warping with length L for each segment.

have limited memory resources, the algorithm memory requirement can be reduced to the order of $\mathcal{O}(2 \times m_0 \times \max\{I, J\})$ by applying a warping window constraint m_0. Nevertheless, as discussed in Section 5.3.3, a narrow warping window will result in insufficient warping to realign long pauses. Therefore, the classical constrained DTW approach fails to meet the memory requirement and apply sufficient warping at the same time. To fulfill the need of small memory computation, we introduce a blockwise variant of the weighted-DTW scheme, where a small segment of the signal is parsed at a time and the memory requirement is significantly reduced. In fact, bounded.

Let L denote the maximum width of a block. As shown in the first diagram of Figure 5-6, we start from the upper left corner of the D matrix and warp within the first sub-block using the weighted-DTW algorithm. The optimal warping path of this sub-block is shown in dark grey. In this example, the last few steps of the warping path travel horizontally along the block border, indicating the last element of the input signal are stretched out to be mapped to a segment of the reference signal, cause signal shape mutation and end-point mis-alignments. This is because the short segments of the input signal and the reference signal may not correspond to the same block of the speech. In other words, the mutated part of the reference signal is likely to be related to the next sub-block of the input signal instead of the current sub-block. In order to avoid throwing away useful information and to make an attempt to align the next sub-block properly, we roll back the reference signal by a few points to form the starting point of the next block (as shown in the green block of Figure 5-6-(b). Similarly, the last few steps of the warping path for the

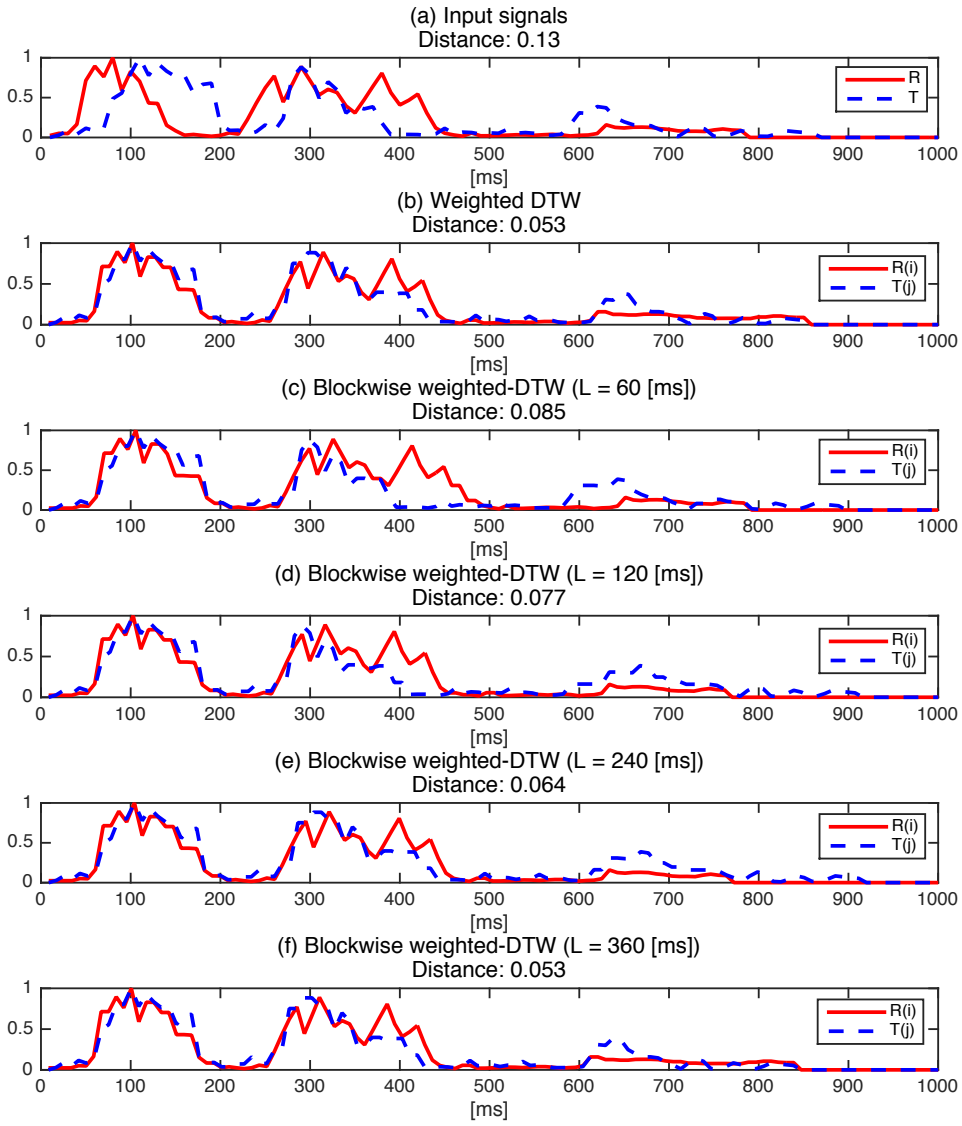


Figure 5-7: Simulation with blockwise weighted-DTW. The input signal corresponds to the same command as the reference signal except with different speaking speed and pauses. Signal warping is performed with weighted-DTW and blockwise weighted-DTW with different segment lengths.

green block travel vertically along the sub-block border. So we roll back the input signal to form the starting point of the next (blue) block. In the end, as shown in diagram 5-6-(c), the paths of the sub-blocks are combined to obtain the overall warping path.

Figure 5-7 shows the simulation results of the blockwise weighted-DTW algorithm compared with the (full) weighted-DTW algorithm. In Figure 5-7, the input signal has duration of 1s and it is a positive signal that corresponds to the same command as the

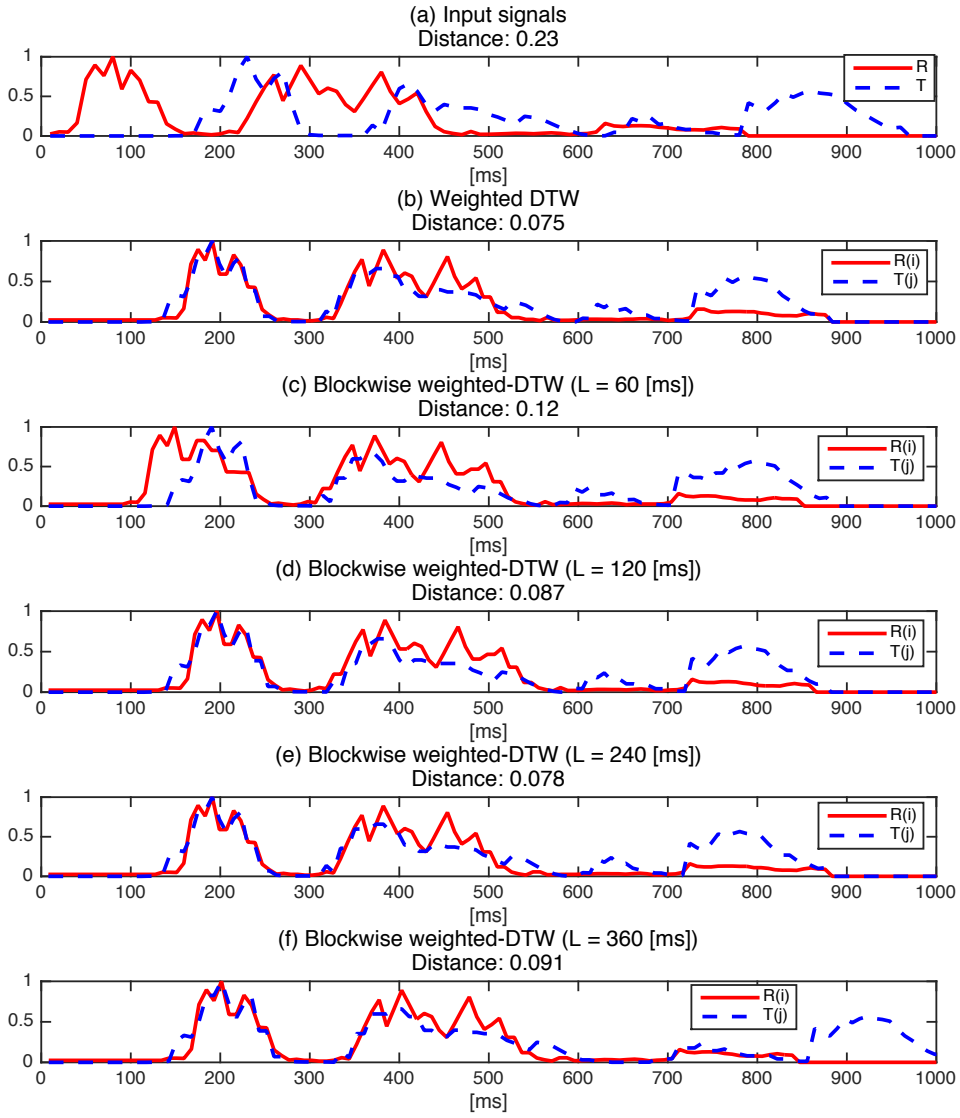


Figure 5-8: The input signal corresponds to a different command as the reference signal. The input signal is processed with the weighted-DTW and the blockwise weighted-DTW with different block lengths. This example shows that longer block length does not guarantee better warping as the block-width of 240ms scheme arrived at a similar result as the weighted-DTW scheme, whereas the block-width of 360ms scheme failed to align the signal envelopes properly.

reference signal. The overall envelope of the two signals are similar. They differ by some mis-alignments, pauses and distortion. With block length greater than or equal to 240ms (e.g., $L = 240$ms and $L = 360$ms), the blockwise weighted-DTW scheme was able to align the signals as effectively as the (full) weighted-DTW algorithm, which operated directly on signals at full length.

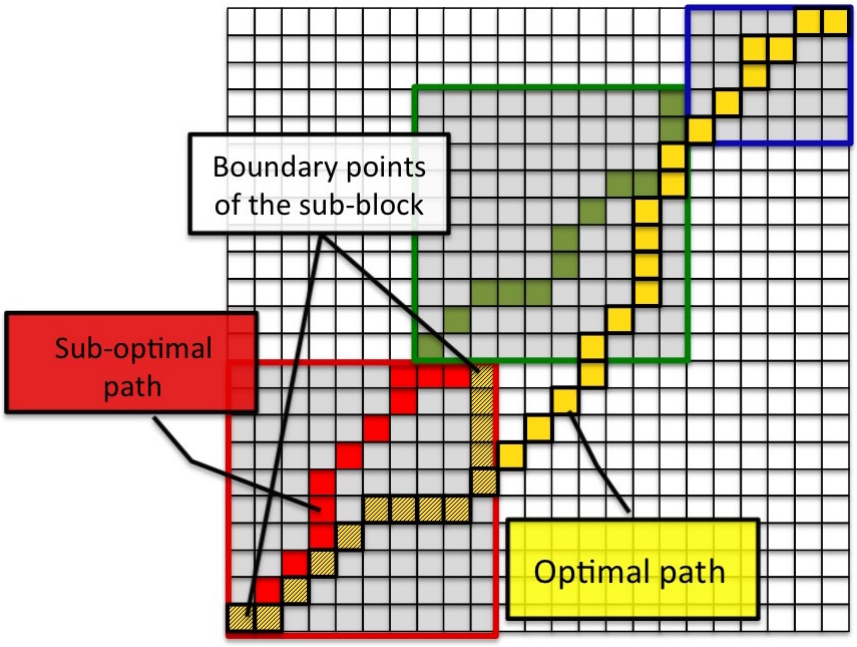


Figure 5-9: Illustration of sub-optimality of the blockwise weighted-DTW algorithm

Figure 5-8 illustrates a few issues associated with the blockwise weighted-DTW algorithm. When the block length is reduced to 60ms, the long delay between the input signal and the reference signal is not completely compensated. This is a result of truncating the signals into small blocks and performing warping, which enforce the start and end points of the signals to coincide. With block length 60ms, the first block of this specific input signal is almost silence (a trail of zeros), which should be ignored and compressed. Unfortunately, because the two ends of the sub-signals need to align, by boundary condition, ignoring the zeros is not an option. This results in the first set of zeros to be aligned with the beginning of the reference signal, causing the delay gap.

It is not hard to tell that the blockwise weighted-DTW algorithm is sub-optimal in comparison to the (full) weighted-DTW algorithm. In addition, one might be compelled to think that the longer the sub-block length, the better the results are. This is generally true, however it is not always the case. When there are no constraints on block length, any path connecting the upper left corner and the lower right corner of the D matrix is a candidate path and the optimal path is one of those candidate paths. When we restrict ourselves to processing a smaller block at a time, the candidate paths are now confined

to the vicinity of the small blocks (as shown in the gray zone in Figure 5-9). With bigger blocks, the coverage of the gray zone is larger compared to the shorter-block case; hence, it is more likely that the optimal path will fall within or close to one of our candidate paths. Therefore, having bigger sub-blocks will generally prevent choosing a warping path that is far away from the optimal.

Nevertheless, larger blocks do not guarantee better warping paths than smaller blocks. As shown in Figure 5-8, the blockwise weighted-DTW with block length 360ms scheme fails to align the envelope of the two signals, while the block length 240ms case performs as well as the weighted-DTW algorithm. As illustrated in Figure 5-9, this result is a by-product of using small blocks.

When we search for the optimal warping path within an individual block, boundary condition of the algorithm implies that the start points and the end points of the two sub-signals coincide. This assumption is reasonable when the signals are processed as a whole, because we know the corresponding contents lie somewhere within the segments. In this case, we can simply tie the two ends of the two signals and stretch the interior of the signals to make them align properly. This is no longer true when we subsequently divide the long signal into small segments. The sub-segments of the two signals may not correspond to the same portion of the speech sample. This issue is mitigated when we allow the signals to roll back if the previous block decided to wipe out the end of the last segment. However, this does not prevent the possibility that a sub-block might choose an alignment that is locally optimal but far away from the globally optimal path. This is why the last bulk of the input signal was not properly aligned in the example with sub-block length 360ms.

5.4 Experiments

In this chapter, we constructed a system with the NBSC feature extraction front-end and the weighted-DTW back-end for the application of text-dependent SV.

Using a data set collected at Texas Instruments Kilby Labs, we compare the speaker-verification accuracy of both the front- and back-end designs to baseline systems under different noise conditions. The experimental results in this chapter focus on the proposed

system's speaker-verification accuracy. Its power estimation is analyzed in Chapter 7. Collectively, the results demonstrate that equivalent or better accuracy can be obtained at much lower power with the proposed system compared to conventional systems.

5.4.1 Proposed and baseline systems

The proposed system comprises the NBSC feature extraction front-end, Section 5.2, and the weighted-DTW back-end, Section 5.3.2.

The baseline systems substitute either the front-end or the back-end, or both. For the front-end, the baseline substitutes are conventional features including MFCC and the more primitive MFSCs. For the back-end, the baseline substitutes include the classical DTW algorithm [65] and the model-based GMM-UBM based system [57], which requires model training with a large amount of training data.

All features are extracted with frame duration of 25 ms and frame rate of 10 ms. The detailed parameters are as follows. For the front-end:

- NBSC (Proposed features): We use the following parameter settings: B = 6 kHz (cutoff frequency of speech signal), W_0 = 200 Hz (bandwidth of narrowbands) and K= 6, 8, 10, 12. Fundamental frequency f_0 is estimated using the auto-correlation method [68].

- MFCC (Baseline feature): The MFCC features have 13 dimensions extracted from the 40-dim Mel frequency filterbank.

- MFSC (Baseline feature): We experiment on two sets of MFSC features: the 26 bands and the 13 bands Mel-frequency filterbank.

For the back-end:

- weighted-DTW (Proposed back-end): as described in Section 5.3.2 with a window length of 250 ms, or, if indicated, the blockwise weighted-DTW as described in Section 5.3.4.

- Classical DTW (Baseline back-end): the classical DTW [65] with the same window length of 250 ms.

- GMM-UBM (Baseline back-end): The GMM-UBM based SV system [57], which requires background model training and assumes prescribed passphrase. We vary the number of Gaussian mixtures and take the parameter that yields the best result.

5.4.2 Experimental set-up

Data set

The primary dataset consists of audio from three different passphrases (two in English and one in Chinese) spoken by 30 to 40 speakers with $20-40$ repetitions per speaker per passphrase (Table 5.1). The data set was collected in multiple sessions and about $2/3$ of all speakers are male and $1/3$ are female. Each passphrase is limited to a duration of 1s and was sampled at 16 kHz.

Table 5.1: Primary dataset

Passphrase	# of speakers	# of repetitions
Hi Galaxy	40	40
Ok Glass	40	20
Ok Hua Wei	30	20

A secondary dataset of out-of-vocabulary (OOV) utterances, consisting of 5000 samples (one second duration) of short commands, speech clips from conversations and audio books, is also used. Finally, noisy samples are generated by adding wind noise or car noise to each clean sample such that the total SNR within 1s is equal to 3dB. The noise samples are collected using the same equipments as the ones used to collect the command samples.

Evaluation and decision threshold

Given a passphrase, every speaker is chosen as the authenticating speaker once. We take 3 utterances from this speaker as enrollment samples and the rest are used as positive (authentic) test samples. The same passphrase from all other users are used as negative (impostor) samples for SV evaluation, while all samples of the OOV dataset are used as negative samples during false-trigger evaluation. For experiments involving noisy samples, the enrollment samples are clean.

The minimum of the distances between a test sample and the enrollment samples is compared with a threshold to make the final verification decision. The threshold is chosen *a posteriori* such that the false-positive and false-negative rates are equal (unless otherwise indicated), which gives the equal-error rate (EER).

For the GMM-UBM model training, the speakers are divided into two halves. The first half is used for background model training and the other half for evaluation. Each user from the evaluation set is chosen as the target speaker once and 4 utterances are used as enrollment samples for speaker specific model adaptation.

Feature adaptation under background noise

In experiments involving noisy samples, we assume a coarse estimate of the noise spectrum is known. The energy of the wind and car noises is concentrated in the low-frequency domain under 2kHz. Therefore, in those experiments, we simply discard spectral features below 2kHz and use the remaining features for NBSC and MFSC front-ends; for MFCC this is not feasible, so there is no feature adaptation (i.e., all of the 13-dim MFCC features are used for both quiet and noisy experiments).

5.4.3 Experiment results

Front-end and back-end combinations

The first set of experiments compare NBSC (proposed) and MFCC front-ends combined with weighted-DTW (proposed), DTW, or GMM-UBM back-ends, for both noiseless and noisy conditions.

Table 5.2: EER [%] for combinations of features and algorithms.

	Clean		Noisy (3dB)	
Algorithm \ Features	MFCC	NBSC	MFCC	NBSC
weighted-DTW	0.9	1.1	10.5	5.7
DTW	1.4	1.5	13	6.7
GMM-UBM	2.6	N/A	6.8	N/A

Table 5.2 shows the EER for systems with different feature and verification algorithms. Without background noise, all of the 12 bands are used as features. With background

noise, only the bands above 2kHz are active. The weighted-DTW algorithm yields better accuracy than the standard DTW algorithm for both the MFCC and the NBSC features. Without background noise, the 12-band NBSC yields slightly worse accuracy than the MFCC features. Under 3dB SNR, the NBSC yields much better performance than the MFCC features as a result of spectral domain feature selection. Without the need for background model training, the proposed system outperforms the GMM-UBM based system in both clean and noisy conditions.

Table 5.3: False-positive rates [%] with OOV dataset. Decision threshold is taken from the EER threshold obtained with the weighted-DTW algorithm in Table 5.2.

	Clean		Noisy (3dB)	
Features / Algorithm	MFCC	NBSC	MFCC	NBSC
weighted-DTW	0	0	1.4	0.6

In contrast to Table 5.2, which evaluates the systems' SV accuracies (same passphrase produced by different speakers), Table 5.3 shows the systems' false-positive rates against the OOV data set (to evaluate the false-triggering rate as a wake-up application). The back-end is fixed to the weighted-DTW algorithm and the decision threshold is the same as that yielded the EER in Table 5.2. Without background noise, the false-trigger rate is 0 for both the MFCC and NBSC features. Under 3dB SNR, the NBSC yields a false-trigger rate of 0.6%, much lower than the MFCC features with a false-trigger rate of 1.4%.

Spectral domain features: NBSC vs. MFSC

The second set of experiments fixes the weighted-DTW (proposed) back-end and compares accuracies of the NBSC (proposed) vs. the MFSC front-ends at various filter-band counts.

Table 5.4 shows that accuracy improves as the number of bands increases. With fewer bands than what is commonly used in MFSC-based front-ends, the NBSC performance is better than that of the MFSC. (Note that the power consumption increases proportionally with the number of bands.) When there is background noise, the accuracy improves significantly by using fewer features (i.e., adaptive band selection).

Table 5.4: EER [%] for NBSC and MFSC features with the weighted-DTW algorithm, under quiet condition and 3dB wind and car noise.

Features	NBSC				MFSC	
# of filters	6	8	10	12	13	26
Clean	1.99	1.9	1.54	1.1	1.95	1.83
Noisy (band selection)	6.8	6.6	6.3	5.7	16.4	17.2
Noisy (all bands)	15.5	15	15	14.5	33.4	33.9

Blockwise weighted dynamic warping

Table 5.5: EER [%] for NBSC (12 band with band selection) and MFCC features with the blockwise weighted-DTW algorithm for different block widths, under clean conditions and 3dB wind and car noise.

Block width (ms)		100	200	300	400	500	Full
NBSC	Clean	3.36	1.55	1.18	1.15	1.16	1.1
	Noisy (3 dB)	8.21	6.71	5.77	5.9	5.66	5.7
MFCC	Clean	2.7	1.2	1.15	0.95	0.98	0.9
	Noisy (3 dB)	12.8	13.04	12.17	12.07	10.9	10.5

Table 5.5 shows the performance of the blockwise weighted DTW algorithm. Generally, it is shown that the verification accuracy improves as the number of block width increases for both the NBSC and MFCC features under noiseless and noisy conditions. The performance improvements saturates at around 300ms block width.

5.5 Summary

In this chapter, we proposed a low-power, text-dependent SV system. The NBSC feature extraction front-end demonstrates improved noise robustness and accuracy for the SV application. The back-end implements a weighted-DTW algorithm that is designed to overcome signal misalignments and speaking rate variation. Compared to the classical

DTW algorithms, the weighted-DTW algorithm successfully aligns the envelopes of the signals without causing mutation on the shape of the signal envelope.

Using as few as 3 enrollments samples and a 12-band feature vector, the proposed system achieves an EER of 1.1%, compared to 1.4% with a conventional MFCC+DTW system and 2.6% with a GMM-UBM based system under noiseless conditions. At 3dB SNR, the proposed system achieves an EER of 5.7%, compared to 13% with a conventional MFCC+DTW system and 6.8% with a GMM-UBM based system. The significant gain in accuracy demonstrates the benefits of adaptive band selection.

Chapter 6

User-independent command recognition

In Chapter 5, we developed a system for the application of voice-authenticated wake-up, in which the device wakes up only when the prescribed command is uttered by the designated speaker. In this chapter, we propose methods for user-independent command recognition such that the system is triggered whenever the prescribed command is uttered by an arbitrary user.

While existing voice-command recognition algorithms based on HMM's or especially DNN's achieve excellent accuracy, they are computationally expensive for standalone devices. To overcome this, we propose a recognition algorithm based on a novel multi-band DNN back-end. In contrast to the generic, fully-connected DNN, the multi-band DNN model is a sparsely connected DNN matched to the spectral features. It simplifies the digital back-end by orders of magnitude vs. a conventional speech DNN for the same classification task. In addition, it supports adaptive feature selection rather than requiring a fixed-set of features like most conventional systems. The combined design of the NBSC feature extraction front-end and the multi-band DNN back-end enables high recognition accuracy, low power consumption, and noise robustness for the application of user-independent voice-command recognition.

6.1 Background on user-independent command recognition

Prior work relevant to user-independent command recognition is found in the KWS literature [26, 31, 32, 69]. The recognition task involves classical pattern recognition to dis-

tinguish candidate classes. For example, template-based algorithms match features from candidate class features directly to query features [26, 27], while model-based algorithms render the speech features as statistical emissions from a class [28–30]. Other methods simply apply large-vocabulary continuous speech recognition to KWS [32], in which speech features are rendered by an HMM driven by phonetic states. The phonetic posteriors generated by the HMM are then used for dictionary-based classification.

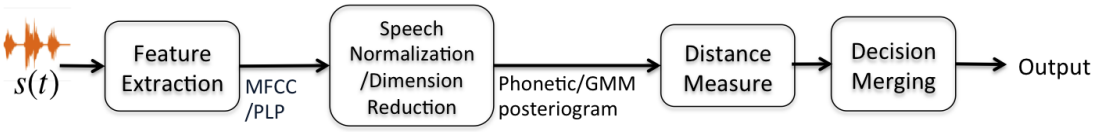


Figure 6-1: Conventional user-independent keyword spotting system with HMM model

Figure 6-1 shows the block diagram of a conventional KWS system based on HMM. The architecture of this system is similar to the general large vocabulary speech recognition system, which consists of an acoustic feature extractor, a phonetic model and an HMM that searches for the word(s) that best matches the input feature sequence. The difference lies within the specifics of the HMM. In the case of KWS, the HMM is trained to described the keyword model and the background (non-keyword) model. Within the keyword model, parameters of the HMM are trained to distinguish different keywords if there is more than one [28, 32, 69]. The HMM model is excellent for modeling the evolution of a time-series of frames. However, it imposes strict dependency constraints on adjacent frames and has widely known issues such as the label-bias problem [70].

Since the keyword spotting task focuses on words with short duration, it becomes possible to treat the entire utterance as a whole. Research has shown that KWS systems using DNNs, which perform classification directly on the acoustic feature vectors without phone level modeling, outperform the phonetic modeling with HMM approach [31].

Figure 6-2 shows the block diagram of Google's KWS system using DNNs [31]. First, acoustic features such as the MFCC features are extracted every 10ms with frame width of 25ms. Then, 40 adjacent frames are concatenated to form a group and each group of frames is interpreted with a DNN model. Figure 6-2-(b) shows the details of the DNN model used to recognize the keyword 'OK Google' in [31]. In this case, the DNN output

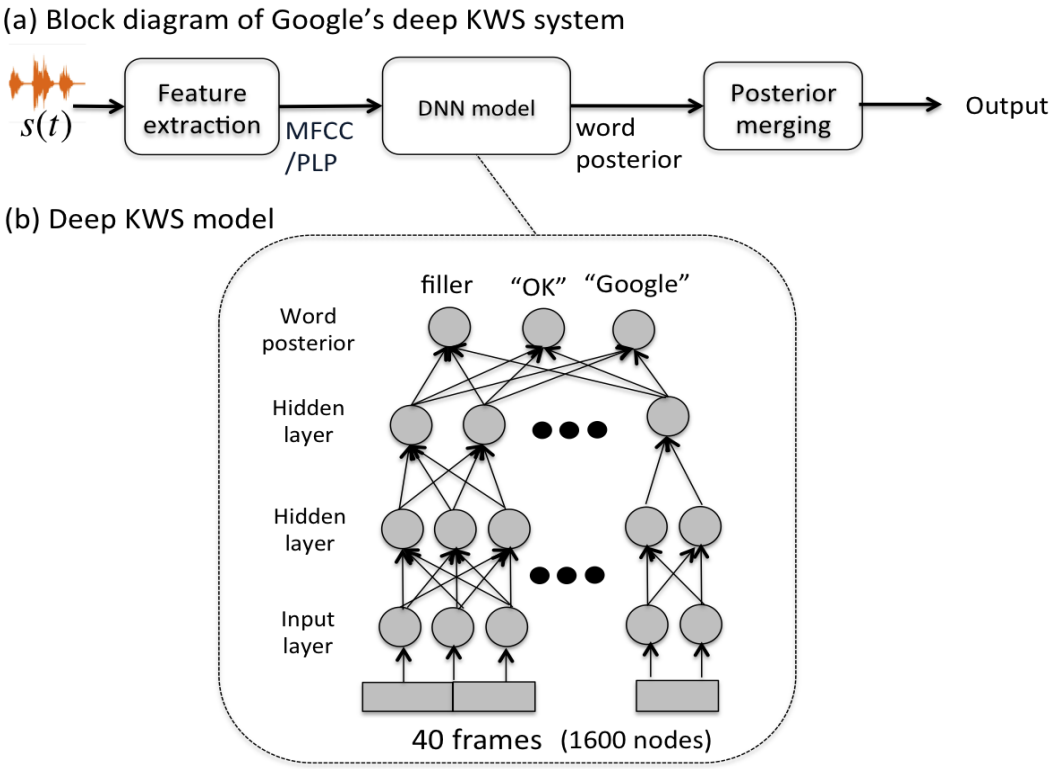


Figure 6-2: Google's KWS system with DNNs

belongs to one of the three classes: 'OK', 'Google' or filler (everything else). Since each frame is a 40-dimensional MFCC vector and there are 40 frames in a group, the input layer of the DNN consists of 1600 nodes. Two hidden layers were used to model non-linearity between the input and the output and a soft-output (i.e. posterior) is computed at every frame. In the end, as shown in Figure 6-2-(a), the output posteriors are sent to a posterior merging unit that makes the final classification decision. Even though this system yields excellent recognition accuracy, the fully-connected DNN model with the high-dimensional feature input requires heavy computation, thus making it impractical for our low-power speech-recognition application.

6.2 Block diagram of the proposed system

Figure 6-3 shows the high-level architecture of the proposed system. The main unit consists of the feature extraction AFE and the multi-band DNN back-end. An external control unit wakes up the main unit whenever an incoming signal of sufficient total power is

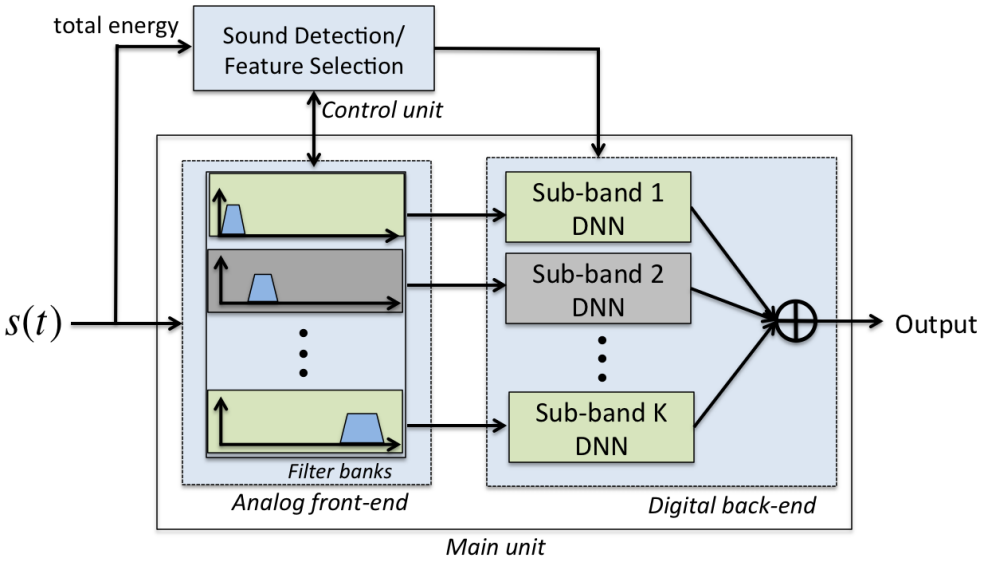


Figure 6-3: System block diagram: the filterbank outputs from each sub-band is fed to its corresponding sub-module of the classifier. The final recognition decision is a weighted sum of sub-band decisions. During training all the sub-bands are turned on. During prediction, the control unit pre-selects a subset of all available sub-bands and only the selected sub-band features (e.g., the green blocks) are extracted and processed.

detected, and more importantly, it adaptively selects a small subset (e.g., ~ 5) of spectral features based on criteria such as the in-band signal-to-noise ratio (SNR) of the sub-bands.

Both the AFE and the back-end are designed to efficiently support adaptive band-selection so that only the selected features are extracted and processed to make the recognition decision. As introduced in Section 4.3, our AFE implementation contains of a band-pass filter bank, which extracts the contents within the selected frequency sub-bands of the original signal. The AFE outputs are the accumulated power within each time frame for the selected sub-bands. The backend DNN-based classifier employs a multi-band model. In contrast to the conventional approach of interpreting a time-sequence of spectral feature vectors as a single super-vector, the multi-band model performs classification disjointly for each sub-band using their corresponding time-sequence of single-band features (as shown in Figure 6-3). The final output is based on a weighted sum of the individual sub-band decisions. The sparsely-connected structure of the multi-band model not only enables adaptive feature selection, but also requires less computation than the fully-connected DNN (e.g., [31]) for any fixed feature dimension.

6.3 Adaptive multi-band DNN back-end

Deep learning with neural networks has demonstrated state-of-the-art performance in a range of speech recognition tasks [25, 31]. In contrast to the conventional approach of modeling the time-frequency features as a whole using one fully-connected DNN (e.g., concatenating 40 frames of 40-dim features into a super-vector for word level classification in [31]), we use a multi-band DNN in the back-end.

6.3.1 The multi-band DNN structure

As illustrated in Figure 6-4, the time-frequency features are divided into separate sub-bands $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$. Each of $\mathbf{x}_i$ represents a time-sequence of filterbank features within a single sub-band over the duration of a keyword (e.g., $\sim$ 1s). Each sub-band is then modeled with a fully-connected DNN whose top layer has two nodes at the output, representing the 'keyword' and 'out-of-vocabulary' (OOV) classes. The top layers of all sub-band DNNs are then connected to the final decision output layer, which also has two nodes representing the 'keyword' class and the 'OOV' class, respectively.

The multi-band DNN model offers the following key benefits:

- *Adaptive band-selection:* When the sub-band parameters are trained disjointly, the multi-band model can be used to support real-time adaptive band-selection such that only the selected sub-bands are used to make each decision.

- *Model size:* Let N denote the total number of frequency bands. Given a fixed number of hidden layers and a fixed number of nodes per layer, the number of edges in the multi-band DNN model increases linearly with N, whereas it increases with N^2 in the fully-connected DNNs because in the latter, nodes corresponding to different bands are cross-connected in each layer. As a result, the multi-band DNN model requires a factor of N fewer multiplications for the recognition task given fixed feature and hidden-layer dimensions; and its sparser structure requires less data for model training. Combining adaptive feature selection with the multi-band model, computation complexity and power consumption of the back-end processing is reduced.

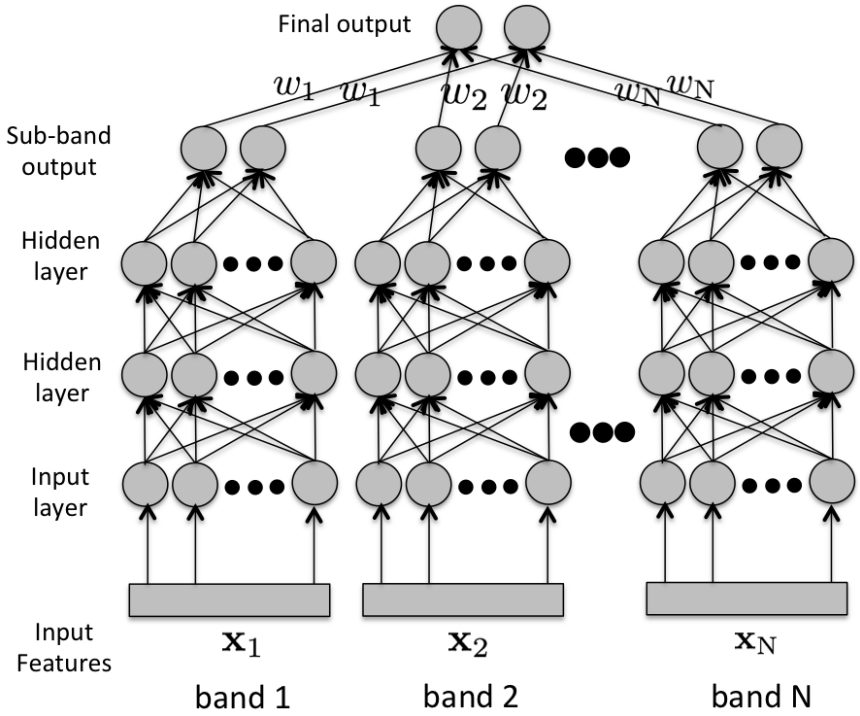


Figure 6-4: The multi-band DNN model: features of each sub-band are input to a separate fully-connected DNN and the individual sub-band decisions at the sub-band output layer are merged into the final output.

6.3.2 Training and classification for adaptive multi-band DNN

Training

The parameters for each sub-band DNN can be trained in a substantially disjoint fashion. We take two approaches for training. In the first approach, each sub-band DNN is treated as an independent classifier trained with the back-propagation algorithm, followed by the weighted-majority algorithm [71] to obtain the weights of output layer with all sub-bands simultaneously presented. Higher weights are assigned to sub-bands with better accuracy.

In the second approach, the sub-band DNNs are first trained independently. Then, the parameters of the individual sub-bands are fine-tuned in sequence using the back-propagation algorithm along with AdaBoost [72–74], which combines a set of weak classifiers to construct a stronger classifier. At each iteration, the weights of the training samples are updated based on the errors made by the current sub-band classifier, and these

weights are used to adjust the back-propagating error for each sample when training the next classifier. At the end, sub-band weights at the top layer are obtained as a result from AdaBoost.

Classification

As illustrated in Figure 6-4, the sub-band decisions are combined as a weighted sum at the final output layer. Let $S \subseteq \{1, ..., N\}$ denote the set of active bands, $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ denote inputs to the N sub-bands, $\{w_1, \ldots, w_N\}$ denote the weights at the sub-band outputs, and $Y = \{y_1, y_2\}$ denote labels for the two output classes. Let $h_n(\mathbf{x}_n, y_i)$ represent the soft-decision output at sub-band n. Then, the final output is a weighted sum of the individual decisions at the active sub-bands:

$$\sum_{n \in S} w_n h_n(\mathbf{x}_n, y_i).$$

This final soft output is followed by a standard decision slicer.

6.4 Band selection

The proposed user-independent command recognition system consists of the universal NBSC front-end and the multi-band DNN back-end.

As introduced in Sections 3.4 and 3.5, the universal NBSC is a set of narrowband spectral features designed to retain the most essential speech information using only a small number of narrowbands.

In the case where sub-bands are selected based on SNR, the active band set S for classification is chosen as follows. We first estimate the sub-band in-band SNRs using the spectrum power distribution obtained from speech training samples and real-time noise power measurements in each band. Let θ_{SNR} denote the minimum in-band SNR threshold and let $K_{\max}$ denote the maximum number of active bands for decision making. The active set S includes the bands with the best in-band SNRs such that a maximum of $K_{\max}$ bands are chosen and all the bands in S must have SNR higher than θ_{SNR}. If S is empty, then use the single band that has the highest in-band SNR.

6.5 Experiments

Through experiments, we evaluate the effectiveness of the universal NBSCs by comparing its recognition accuracy with the conventional MFSC and MFCC features, in combination with the multi-band DNN back-end.

Aside from the choice of acoustic features, there is also the question of whether the sparser connection of the multi-band structure results in a loss of accuracy. We answer this question by comparing the recognition accuracy of the multi-band DNN model with that of a fully-connected DNN model for the same feature type. Finally, we demonstrate the recognition accuracy can be improved while performing less computation by using adaptive band-selection schemes in the presence of noise.

6.5.1 Experiment setup

We analyze the multi-band DNNs model in two sets of experiments. First, we investigate how well the multi-band DNN structure can model speech commands compared to with the conventional fully-connected DNN, in which all the sub-bands are cross-connected. In this case, we fix the band selections and analyze the performance when the same sub-bands are used for both training and classification. The band selection is chosen in a way that yields the best accuracy among all choices of the same subset cardinality. We analyze the performance as the number of sub-bands increases. Second, we study the system performance of adaptive feature selection (Section 6.4), with SNR threshold set to $\theta_{\text{SNR}} = 5$ dB and maximum band usage $K_{\max} = 5$.

Data sets and features

The clean data set includes 3000 positive examples of the keyword 'Hi Galaxy' recorded by 100 different speakers, and 32K negative examples (12K examples of other commands and 20K short phrases taken from audio books and audio shows). The bandwidth of the speech samples is 8kHz. The noisy condition data sets are generated by adding to each sample of the clean data set either a recording of real noise data or a pseudo-noise sample of defined spectral statistics.

We experiment with two types of spectral features: the 13-band MFSCs and the universal NBSCs (Section 3.5). Unless otherwise specified, the universal NBSCs are extracted from 400Hz narrowbands, that are evenly spaced out across the speech spectrum.

DNN model size and parameters

Recognition is performed at the command level with 1.2s of audio content. The features are extracted at a frame rate of 10ms. As a result, the input feature dimension is 120 for each of the K sub-bands of the multi-band model and it is $K \times 120$ for the fully-connected model (K is the number of active bands). Both the multi-band DNN and the fully-connected DNN have 3 hidden layers, whose dimension (i.e., the number of nodes in each layer) reduces by $1/2$ at each layer.

The back-propagation algorithm is implemented with the mean-square-error cost function and random parameter initialization. The learning rate is 0.01 and the training procedure terminates when the gradient is less than 10^{-7} or when it exceeds 1000 iterations. For the fully-connected model, when it is trained with dropout [75], the probability of retention is 0.9 for the input layer and 0.5 for the hidden layers.

Performance measurement

In each simulation configuration, a random 90% of the samples are used for training the DNN and the remaining 10% are withheld for classification. This is repeated 10 times and the results are averaged.

6.5.2 The fixed band DNN experiments

In these experiments, the band selection is presumed fixed for both training and classification. Noisy samples are generated by adding samples of real noises (e.g., babble noise, car noise, wind noise, radio and audio book noise) to clean samples such that the total SNR is 0dB.

Figure 6-5 compares the accuracy (1 - EER) of the fixed multi-band model and the fully-connected model as a function of the number of frequency bands under quiet and noisy conditions, using the conventional MFSC features. There are three main points to note from Figure 6-5. First, even though the multi-band model presumes a disjoint

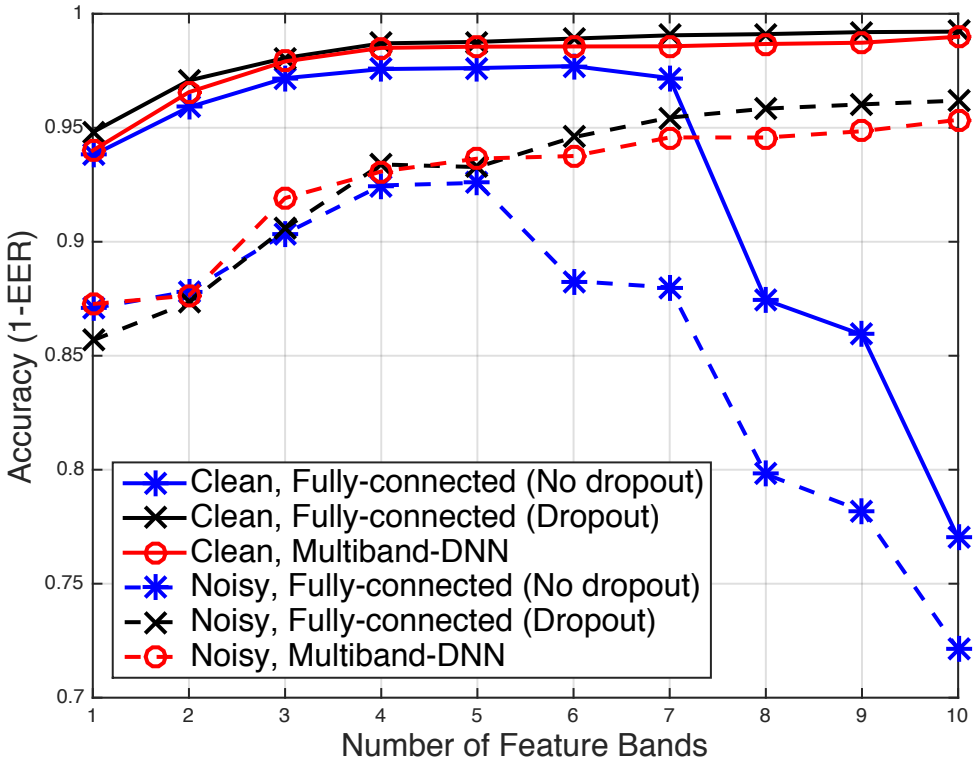
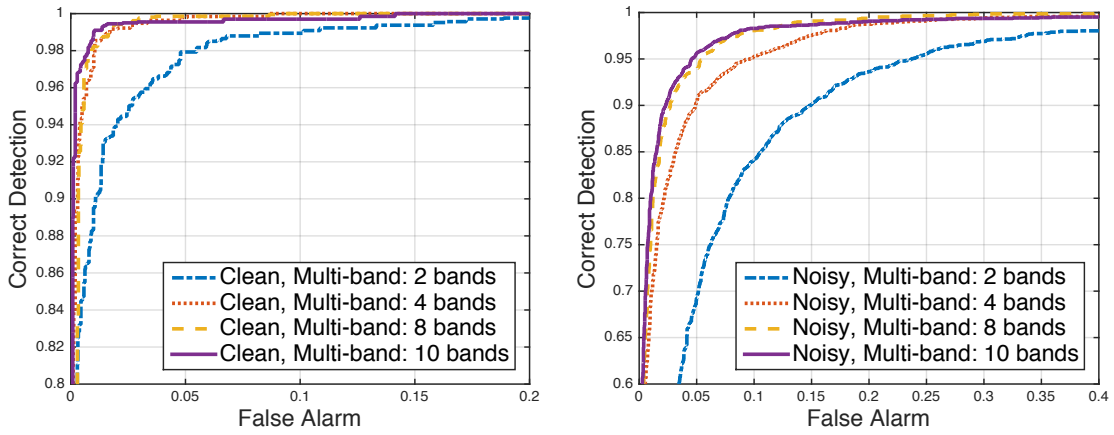


Figure 6-5: EER performance with MFSC: the multi-band DNN offer accuracies comparable to the fully-connected DNN. Unlike the fully-connected model, the multi-band model does not suffer from over-fitting when the number of bands increases.



(a) Noiseless background (b) Noisy background at 0dB SNR

Figure 6-6: ROC performance with MFSC: ROC corresponding to operating points from Figure 6-5. Recognition accuracy improves as the number of bands increases and stops at around 4 bands when there is no background noise and at 8 bands under noisy conditions.

structure among different bands, the multi-band model yields similar recognition accuracy as the fully-connected model. Secondly, the performance of the multi-band model

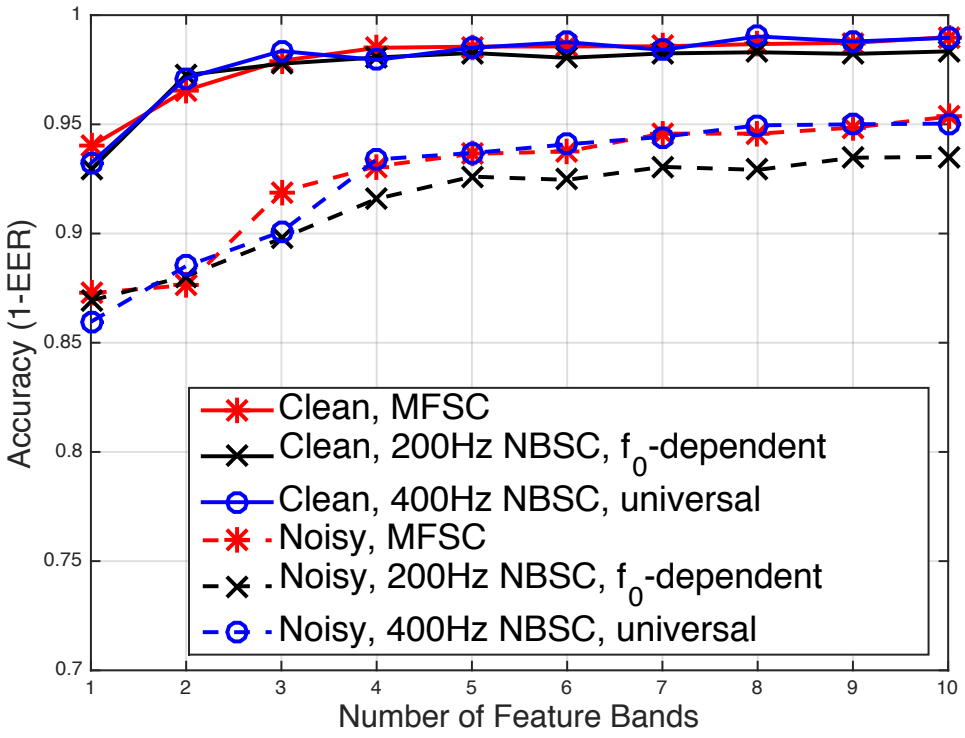
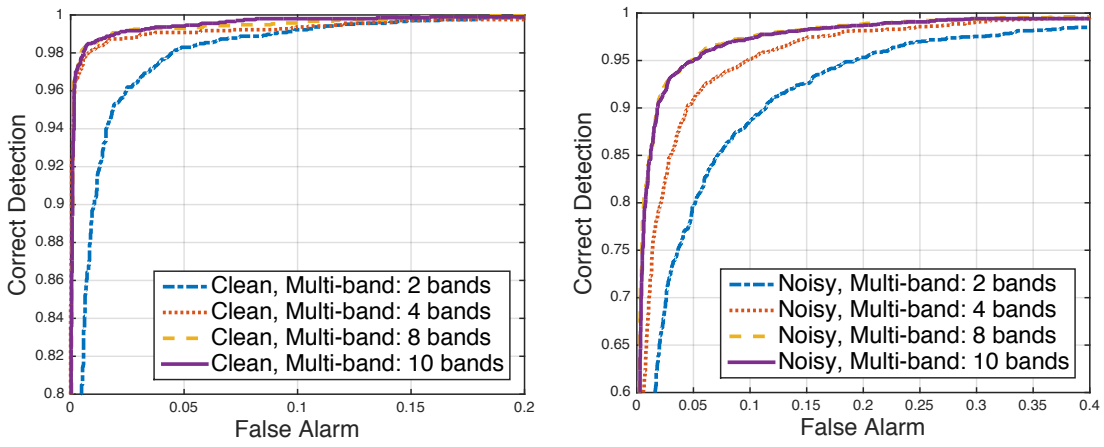


Figure 6-7: EER performance with NBSC and MFSC using the multi-band DNN model. NBSC offers similar performance as the MFSC. Two types of NBSCs were used: the 400Hz fixed band NBSC that are chosen to be evenly spaced out across the spectrum and the 200Hz adaptive band NBSC that are chosen to be around the harmonics of the speech and spread out cross the spectrum. The harmonic based NBSC shows inferior performance when there is noise due to inaccuracies in pitch estimation.

increases steadily with the number of bands and saturates at 4 bands and 8 bands under noiseless and noisy conditions, respectively. Similar behavior can be seen from the receiver operating characteristic (ROC) curves shown in Figure 6-6. This implies that, computation resources can potentially be optimized by using fewer bands under quiet conditions and by including more bands for recognition when noise is present. Lastly, the multi-band structure allows the training data size per band to be independent of the number of bands, whereas the fully-connected model requires an increasing number of training samples with increasing number of bands N. This is illustrated in Figure 6-6(a), where it shows that, when the fully-connected DNN model is trained without dropout, over-fitting occurs when the number of bands exceed a certain threshold.

Figure 6-7 compares the recognition accuracy between MFSCs and NBSCs using the same multi-band DNN back-end. Two types of NBSCs are used: the 400Hz universal NB-



(a) Noiseless background (b) Noisy background at 0dB SNR

Figure 6-8: ROC corresponding to operating points from Fig. 6-7. Recognition accuracy improves as the number of bands increases and stops at around 4 bands when there is no background noise and at 8 bands under noisy conditions.

SCs (labeled as '400 Hz NBSC, universal') are chosen to be evenly spaced out across the spectrum and the 200Hz f_0-dependent NBSCs (labeled as '200Hz NBSC, f_0-dependent') [1]) are chosen to be around the harmonics of the speech and spread out cross the spectrum. Generally, the NBSCs offer similar performance as the MFSCs except the f_0-dependent NBSCs show inferior performance when there is noise due to inaccuracies in pitch estimation. Similar to the MFSCs, when the NBSCs are used as features, recognition accuracy also saturates after the number of bands reaches a certain threshold. For example, recognition accuracy saturates at 3 bands and 8 bands under clean and noisy conditions, respectively. This is also illustrated in the ROC curves shown in Figure 6-8.

6.5.3 The adaptive multi-band DNN experiment

In these experiments, the effect of band selection according to SNR is investigated. Here, two types of noisy samples are used. First, pseudo-noise are added to clean samples. The spectrum of the noise samples are shaped to be band-wise white in 500Hz bands in the range of 0-8kHz with in-band SNR randomly chosen between -10dB and 15dB. A noisy

[1] Pitch estimation is performed using the whole 1.2 second speech segment and with the auto-correlation pitch estimation method [68]. This experiment is conducted for the purpose of feature analysis. It helps us to understand how much loss there is by choosing a universal set of features instead of f_0-dependent features. In practice, real-time pitch estimation may be computationally expensive when speech is processed continuously in real time.

band is discarded if its in-band SNR is less than 5dB (i.e., θ_{SNR}) and a maximum of 5 bands are used. In the second set, real noises (wind and car noises) are added to clean samples such that the total SNR is randomly chosen between -5dB and 10dB. Similar to the feature selection approach used for speaker-verification in Section 5.4.2, features under 2kHz are discarded under noisy conditions (i.e., total SNR falls below the 5dB SNR threshold).

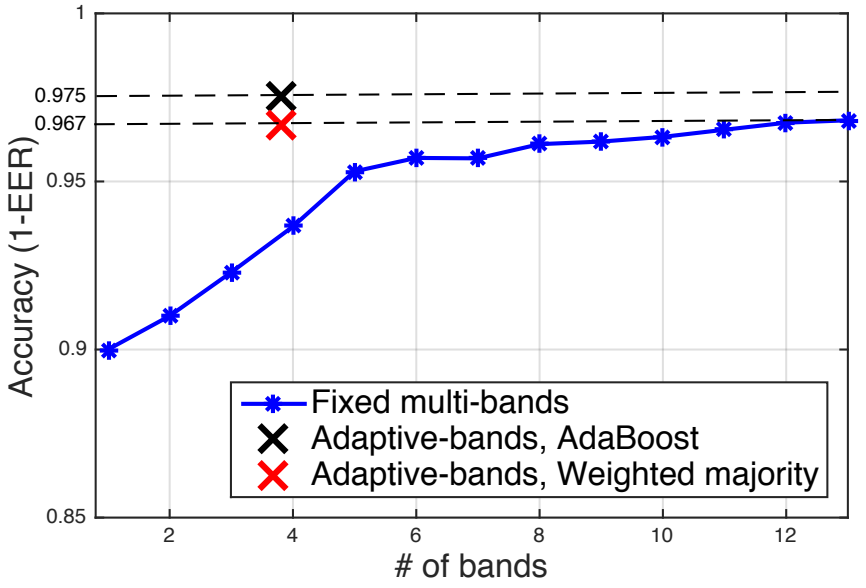
Pseudo-noise

Figure 6-9(a) shows the performance of two adaptive multi-band schemes relative to the fixed multi-band using MFSCs under pseudo-noise. On average, fewer than 4 frequency bands are chosen to be active by adaptivity. The AdaBoost method and the weighted-majority method yield an accuracy of 97.5% and 96.7%, respectively. The best performance for the fixed multi-band method is achieved with all 13 bands, and yields an accuracy of 96.8%, which is slightly less than the AdaBoost method, demonstrating the substantial benefits of rejecting noisy bands adaptively. Similar observations are shown in the ROC plot in Figure 6-9(b).

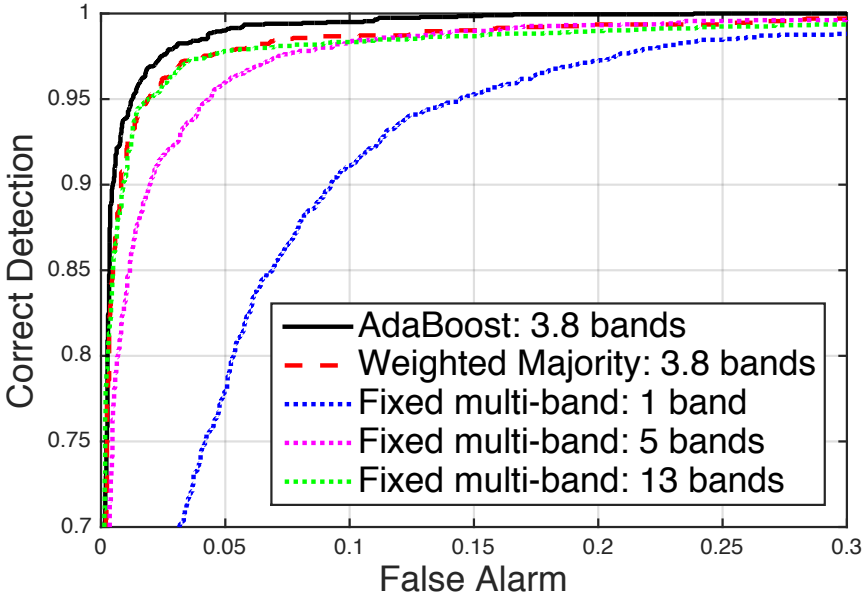
Similar to Figure 6-9(a)-(a), Figure 6-10-(a) plots the same figure when the universal NBSCs are used instead of the MFSCs. With an average of 4.2 active frequency bands, the adaptive multi-band method achieves an accuracy of 98.1%, which slightly outperforms the fixed multi-band approach with all 10 bands, as well as the adaptive multi-band scheme with MFSCs. In addition, we have implemented the conventional system with the 13-dim MFCC features (generated from the 40-band Mel-frequency filterbanks) with the fully-connected DNN back-end. The fully-connected DNN model is trained with the back-propagation algorithm with dropout. The MFCC scheme achieves an EER of 98.7%. Figure 6-10(b) plots the ROC of the universal NBSC schemes and the MFCC scheme. As shown in the figure, the adaptive NBSC multi-band scheme achieves comparable performance as the MFCC scheme while using fewer features and less processing.

Real noise

Figures 6-11 and 6-12 plot the analogous results as Figures 6-9 and 6-10, for samplers with wind and car noises, which are common for our application and have the special charac-



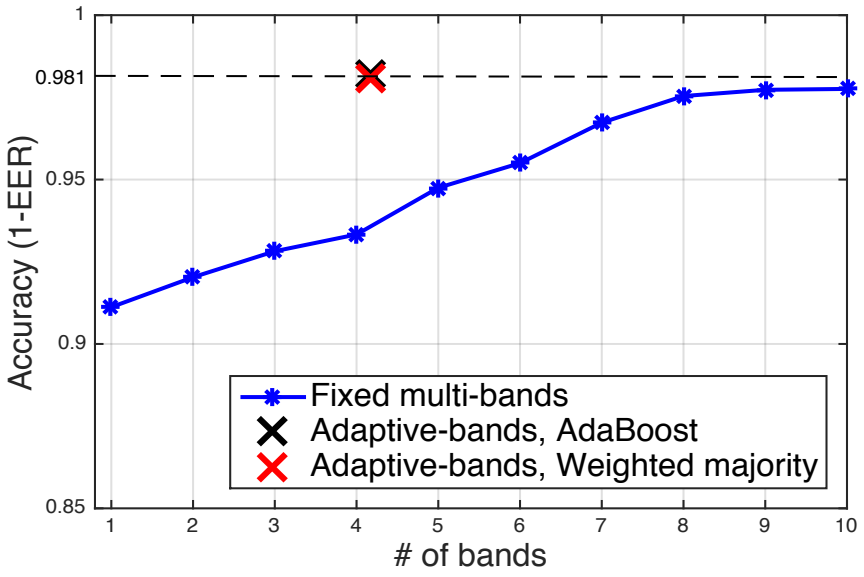
(a) Pseudo noise: EER



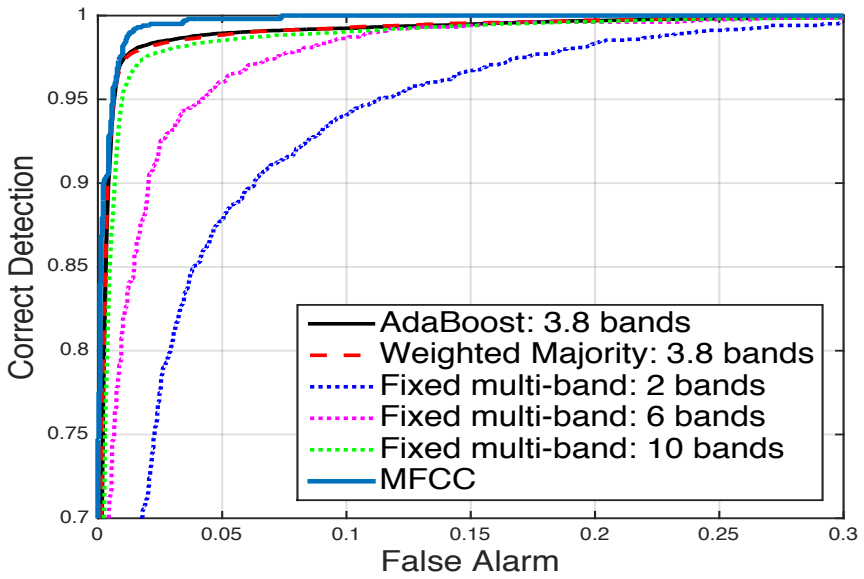
(b) Pseudo noise: ROC

Figure 6-9: MFSC with adaptive multi-band DNN under pseudo-noise: with an average of less than 4 frequency bands, the adaptive multi-band method achieves an accuracy of 97.5%, which outperforms the fixed multi-band approach with 13 bands.

teristics of concentrating narrowly in low frequencies. Similar to the case with pseudo-noise, by adaptively selecting a subset of 5 features using the procedure described in Sections 6.4 and 6.5.1, the adaptive system achieves comparable performance as the fixed (non-adaptive) approach, which uses more than twice the number of features. Similar to



(a) Pseudo noise: EER

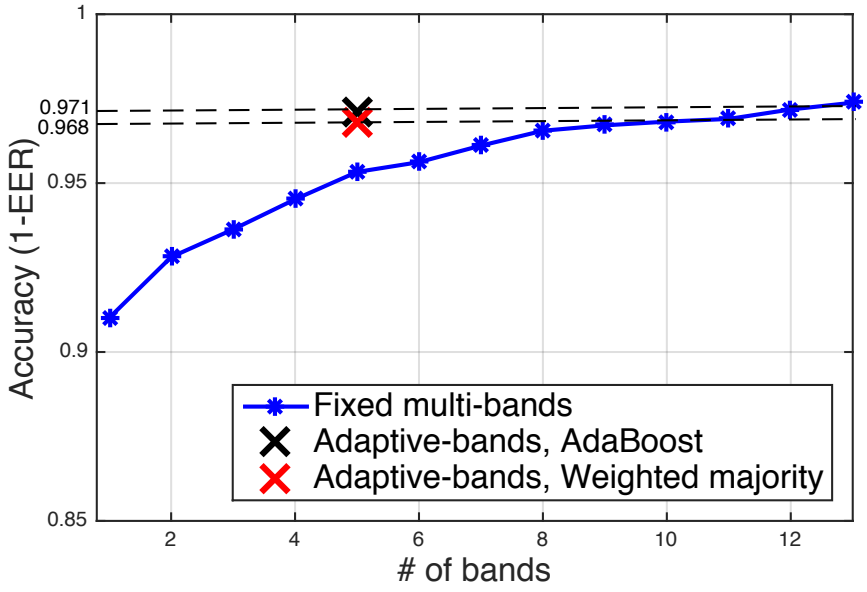


(b) Pseudo noise: ROC

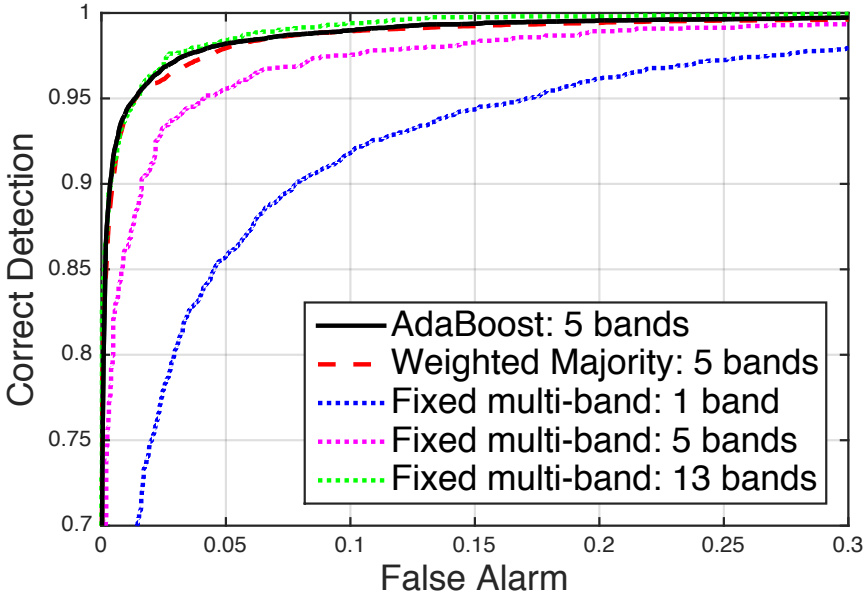
Figure 6-10: NBSC with adaptive multi-band DNN under pseudo-noise: with an average of 4.2 frequency bands, the adaptive multi-band method achieves an accuracy of 98.1%, which outperforms the fixed multi-band approach with 10 bands. The MFCC in (b) is a 13-dim MFCC generated with a 40-band Mel-Frequency filterbank. The recognition algorithm with the MFCC features is the fully-connected DNN with trained with back-propagation and dropout. The MFCC scheme achieves an EER of 98.7%

.

the experiment with pseudo noise, the NBSC schemes yield improved performance than the MFSC schemes.



(a) Real noise: EER

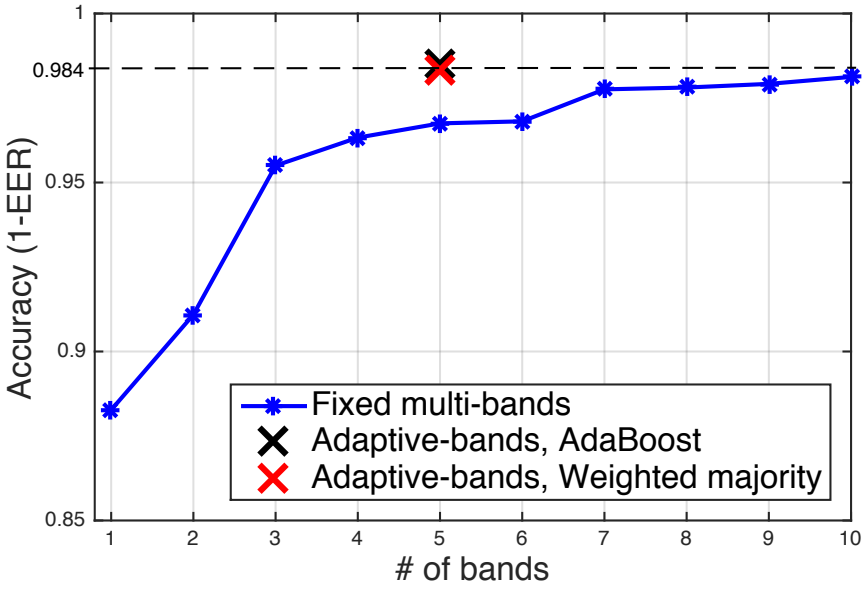


(b) Real noise: ROC

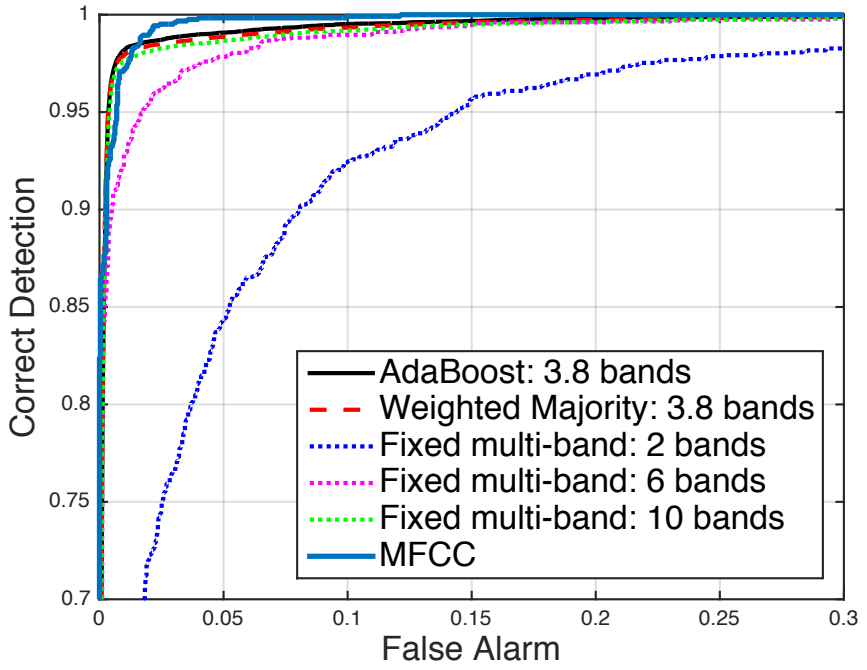
Figure 6-11: MFSC with adaptive multi-band DNN under real noise: with 5 frequency bands, the adaptive multi-band approach achieves comparable performance as the non-adaptive approach that uses 13 bands.

6.6 Summary

In this chapter, we presented a low-power user-independent voice-command recognition system, which consists of the universal NBSCs front-end and a word-level command rec-



(a) Real noise: EER



(b) Real noise: ROC

Figure 6-12: Universal NBSC with adaptive multi-band DNN under real noise: with 5 frequency bands, the adaptive multi-band approach achieves comparable performance as the non-adaptive approach that uses 10 bands. The MFCC scheme achieves an EER of 98.3%.

ognizer. The back-end command recognizer, supporting adaptive processing, is enabled by a novel multi-band DNN model that processes only the selected features at each de-

cision. Without degrading the recognition accuracy, the multi-band structure requires fewer training samples and less computation for classification compared to the conventional fully-connected DNN model. In addition, the adaptive band selection approach that activates a subset of all available frequency bands leads to simpler processing, improved noise robustness and low power consumption. In particular, our adaptive scheme, using an average of 5 spectral band features, achieves comparable accuracy and improved efficiency over a generic fully-connected DNN model using the full speech spectrum.

Chapter 7

Hardware implementation and power estimation

In this chapter we outline a hardware design for our proposed system (based on the Cortex-M0 micro-controller for its digital portions), and make power estimations against it. To demonstrate this design achieves low-power consumption, we calculate the end-to-end system power through a combination of physical measurements and collection of reported figures on existing technologies, including those of alternate subsystems of similar design and functionality. In the following, we first introduce the digital subsystem. Then, the system's front- and back-end portions are discussed separately.

7.1 Digital processing hardware

The hardware we use for digital processing is a Cortex-M0 micro-controller designed by Texas Instruments (TI). The chip has a memory space of 40kB. Figure 7-1 shows the test board of the chip. As shown in the figure, the chip is connected in series with a 10Ω resistor. We measure the voltage across the resistor, which is then used to compute the current going through the resistor. This current is approximately the same as the current going through the chip, which we simply denote as I. The voltage supply for the chip is $1.8V$. Therefore, the energy drawn by the chip for a certain task can be estimated as $1.8 \times I \times t_{\text{duration}}$ J, where t_{duration} corresponds to the duration of processing time for the task.

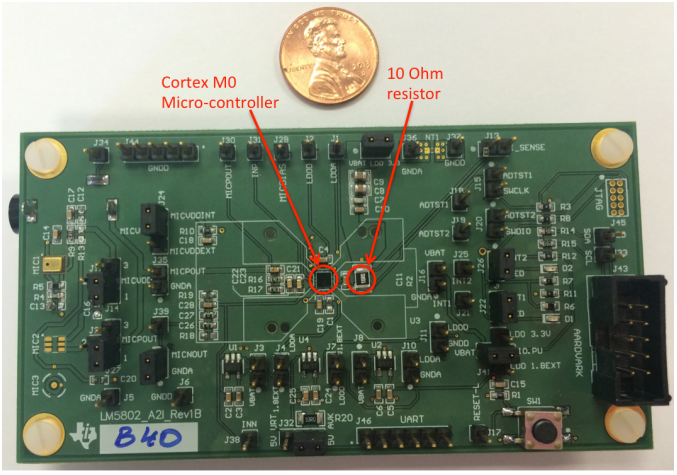


Figure 7-1: Test board of TI's low-power speech recognition chip design, which includes a Cortex-M0 micro-controller for digital computations. The voltage supplying the chip is 1.8V.

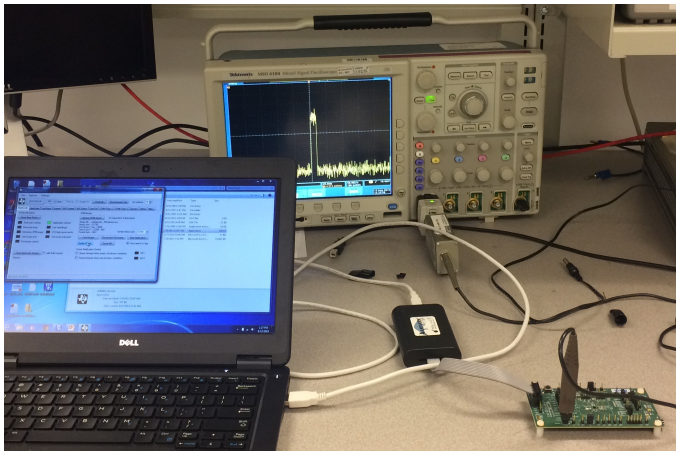


Figure 7-2: Simulated outputs from the front-end filter banks are loaded onto the chip. Then, the subsequent multi-coset reconstruction procedures are coded and ran on the micro-controller. The voltage across the 10Ω resistor and t_{duration} are measured with an oscilloscope.

7.2 Front-end subsystem

First, we estimate the power consumption of the NBSC feature extraction front-end proposed in Chapter 4, broken down functionally by its analog filterbank and digital post-processing constituents. Then, we briefly introduce TI's existing implementation of an

MFSC feature extraction front-end module as an alternate hardware reference and report its power consumption for comparison. Recall that, similar to the NBSCs, the MFSCs is also a set of features representing the power envelopes of individual frequency bands (the difference is that the bandwidths of the MFSCs are chosen according to the Mel-frequency scale where the NBSCs have narrowbands of equal bandwidths). And as introduced in the Section 5.4 and Section 6.5, the MFSC features are fully compatible with our back-end algorithms. Therefore, the power consumption of the MFSC front-end is indicative of the power consumption of an NBSC front-end if fully integrated into hardware, due to similarities in design. It also serves as an alternate choice for the front-end subsystem that exists today.

7.2.1 NBSC Feature extraction

The NBSC feature extraction front-end consists of a microphone, a set of bandpass filterbank, analog-to-digital converters (ADCs) and subsequent digital processing in section 4.3). In general, the system requires a microphone for its usage and there are low-power active dynamic microphones such as the ICS-40310 MEMS microphone, which consumes less than 20μW of power [1]. Nevertheless, for the task of system wake-up, our system uses a coil moving loud speaker which is a passive component consuming zero power and is already built-in in most mobile devices. Therefore, the total power consumption of the feature extraction front-end is equal to the sum of the power consumptions of the filterbanks, ADCs and addition digital processing. We estimate the power consumptions of the analog filterbank and ADCs from existing research work. We implement the post-processing procedures on the Cortex-M0 micro-controller and measure the power consumption for subsequent processing.

Existing work on analog filterbank and low-rate ADC designs

There is a large number of analog front-end filter designs that achieve low-power consumption for speech frequency signals [45, 76, 77]. Most of these existing works use the subthreshold Gm-C filter designs due to its wide tuning range and low-power consump-

[1] The ICS-40310 ultra-low current microphone from InvenSense transforms an acoustic signal to an analog electric signal. It runs from a 1V supply and consumes only 16μA of current while providing a 64dB SNR.

tion property [78, 79]. The circuit topology uses a cascade of first-order highpass and first-order lowpass filters based on RC primitives. The lower and higher cutoff frequencies can be tuned by changing the bias currents and the fall-off rates of the filters are related to the design parameter: the Q-factor, which depends on the structure of the RC cascade filterbanks.

For example, a 16 channel programmable analog filter bank is presented in [45]. The filters are designed to operate under the audio frequency range: 20Hz-20kHz and the center frequencies and bandwidths of the filters can be reprogrammed by varying the floating-gate current sources. The filter bank chip includes 16 parallel channels of band-pass filters, magnitude detectors [44] and current biasing floating-gate transistors [80]. The total power consumption of the chip is 63.6μW. A similar low-voltage programmable filter designs is introduced in [76], which achieves a power consumptions of $\leq$ 6.36μW per band.

At low sampling rates, the ADC power consumption is limited by its leakage power and ADCs should be designed to have very low-leakage current in order to achieve nWs power-consumption levels. We estimate the ADC power consumption from the low-leakage successive approximation (SAR) ADC design presented in [81]. In this design, the leakage power is kept under 0.5nW. The operating power consumption can be estimated using the Walden figure of merit (FoM) [35] given in [81]. FoM is defined as:

$$\mathrm{FoM} = \frac{\mathrm{P}}{f_\mathrm{S} \times 2^{\mathrm{ENOB}}} \tag{7.1}$$

where ENOB is the effective number of bits, f_S is the sampling rate and P is the ADC's power consumption. With FoM = 17fJ per conversion step [81] and $f_\mathrm{S} = 400$Hz, a 10 bit ADC would have a power consumption of:

$$\begin{aligned} \mathrm{P} &= \mathrm{FoM} \times f_\mathrm{S} \times 2^{\mathrm{ENOB}} \\ &= 17 \times 10^{-15} \times 400 \times 2^{10} \\ &\approx 6.96 \text{ nW}. \end{aligned} \tag{7.2}$$

In summary, [45] and [76] offer audio-frequency analog filter designs that achieve an average power consumption of less than 4μW and 7μW per band, respectively. The power consumption of a 400Hz ADC can be kept at 7nW with low-leakage ADC designs [81]. Therefore, with existing technologies, the total power consumption of the analog filter-banks and the ADCs can be estimated to be under 10μW per band.

Multi-coset sampling post-processing

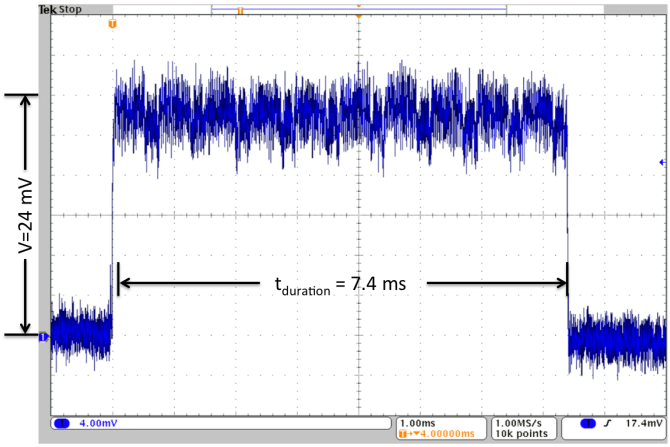


Figure 7-3: Power measurement for multi-coset processing of 1s of data with 10 active bands. The total measured energy is approximately $32\mu J$.

In this implementation, the simulated analog filterbank outputs are loaded onto the chip. The chip performs matrix inversion of $\mathbf{A}_{\text{sub}}$ and multi-coset feature extraction as shown in Figure 4-11. The filter bandwidth is set to 400Hz, which is the widest bandwidth we select for our experiments. The baseband digital filter has 100 taps (i.e., $N_{\text{taps}} = 100$ in Section 4.3.3). Figure 7-3 shows the energy consumption for extracting 10 narrowband features over one second of speech data.

The power consumption of the chip is equal to $V_{\text{chip}} \times I \times t_{\text{duration}}$, where $V_{\text{chip}} = 1.8V$ and I and t_{duration} are shown in these screen-shots (the horizontal axis corresponds to time and the vertical axis corresponds to voltage). The current, I, going through the chip is the same as the current through the resistor, which is equal to the shown voltage divided by the resistor resistance (i.e. 10Ω). The computation duration, t_{duration}, is given by the time-axis readings and it indicates the duration of time the micro-controller takes to process

one second of speech data. In summary, the computation power for $K = 10$ is given by:

$$\begin{aligned} \mathrm{P} &= \text{Energy per second} \\ &= V_{\text{chip}} \times I \times t_{\text{duration}}/1\text{s} \\ &= 1.8\text{V} \times \frac{24\text{mV}}{10\Omega} \times 7.4\text{ms}/1\text{s} \\ &= 32\mu\text{W}. \end{aligned} \tag{7.3}$$

Hence, the processing power is approximately 32μW for $K = 10$ bands and it is approximately 3.2μW per band.

Summing the power consumptions of the analog filterbank and subsequent processing, the total power consumption for the NBSC feature extraction front-end is estimated to be under 14μW per band and less than 140μW for a 10-band front-end.

7.2.2 TI's MFSC AFE

A proprietary MFSC analog feature extraction front-end is under design by TI. The front-end consists of a voice-activity-detector (VAD) and a 12 band MFSC feature extraction unit. The filter fall-off characteristics are similar to our NBSC front-end design. The cutoff frequency of the speech signal is at 6KHz. The sub-band features are extracted at a frame rate of 200 frames per second with frame duration of 5ms. The front-end has a fixed power consumption of 150μW and an additional power cost of 10 μW per band.

7.3 Back-end subsystem

The text-dependent speaker-verification and user-independent command recognition back-ends are also implemented in hardware. Their power consumption is measured by computing the energy consumption for each decision and then multiplied with the number of decisions per second. In the end, the power consumption of the end-to-end system are measured as the sum of the front-end and back-end power consumptions.

7.3.1 Text-dependent speaker-verification

For the application of text-dependent speaker verification, the verification decision is made with 1.2s of buffered data at every 60ms. The features are extracted at a rate of 100 frames per second. In other words, 1.2 seconds of speech content consists of 120 frames. Each frame is represented by a feature vector of dimension K, where K is equal to the number of active bands.

The speaker-verification algorithm is the blockwise weighted-DTW algorithm introduced in section 5.3.2. In order to reduce memory requirements, the 120 dimensional input features are divided into smaller blocks of size 40 for processing. Three enrollment samples are used for decision making. The decision threshold is chosen a priori, based on offline training results. At each decision, the distance between the input signal and each of the enrollment samples are computed in series using the blockwise weighted-DTW algorithm. The minimum of the distances are compared with the verification threshold to make the final decision.

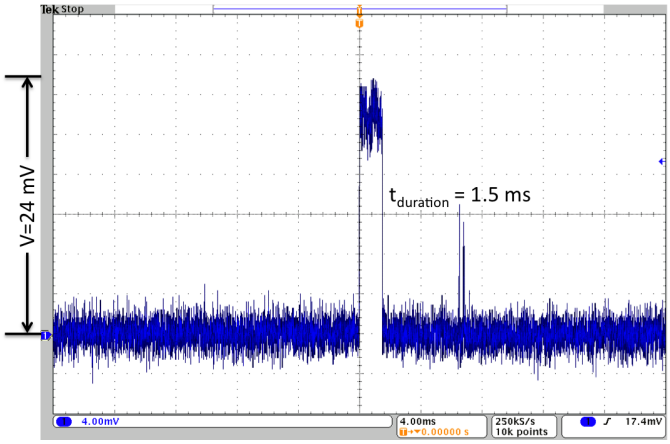


Figure 7-4: Energy consumption for text-dependent speaker verification per decision with 12 active bands. The total energy consumption is approximately 6.48μJ.

The back-end algorithm is implemented on the Cortex-M0 processor described in section 7.1. With number of bands $K = 12$, the firmware implementation for the algorithm and data occupies less than 40kB memory. Figure 7-4 refers to the power measurement for making a single decision when $K = 12$. As shown in the figure, t_{duration} corresponds to the duration of time that the micro-controller is turned on for making a single decision.

Therefore, the average power consumption for 12 active bands at a decision rate of every 60ms is given by:

$$\begin{aligned} \mathrm{P} &= \text{Energy per decision} \times \text{number of decisions per second} \\ &= V_{\text{chip}} \times I \times t_{\text{duration}} \times \frac{1}{60\text{ms}} \\ &= 1.8\text{V} \times \frac{24\text{mV}}{10\Omega} \times 1.5\text{ms} \times \frac{1}{0.06\text{s}} \\ &= 107.57\mu\text{W}. \end{aligned} \tag{7.4}$$

Because the complexity of the algorithm scales linearly with K (see section 5.3.2) and the power consumption is directly proportional to the number of operations, the power consumption per band is estimated to be less than 9μW per band.

7.3.2 User-independent command recognition

The user-independent command recognition power consumption is measured in a similar manner. A 10-band multi-band DNN model was implemented on the Cortex-M0 processor. The input feature in each band is down-sampled to a rate of 50 samples per second and 1.2s of buffered speech is used to make each decision. A recognition decision is made at every 40ms (i.e., 25 times per second).

The input feature dimension for each sub-band is 60. Each sub-band has three hidden layers and the sizes of the hidden layers are 60 nodes, 30 nodes and 15 nodes, respectively. As described in section 6.3, the final hidden layer is connected to two output nodes, representing the command class and the 'OOV' class. The parameters of the multi-band DNN model occupies around 40kB of memory. The sigmoid function is implemented as a look-up table.

Power consumption is estimated in a similar manner as the text-dependent speaker-verification back-end. The energy consumption per decision is approximately $3.88\mu J$ when $K = 10$. Then, the energy value per decision is multiplied by 25 decisions per second to obtain the power consumption measurement. With $K = 10$, the total power consumption under 100% duty cycling is approximately $97\mu W$. Since the power con-

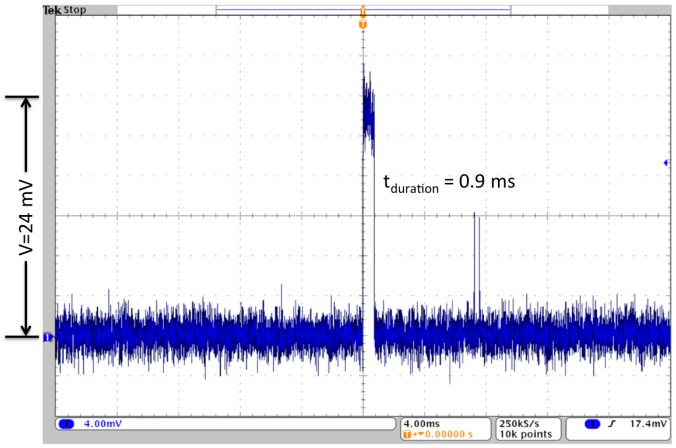


Figure 7-5: Energy consumption for under-independent command recognition per decision with 10 active bands. The total energy consumption is 3.88μJ.

Table 7.1: Summary of system power consumption for different components

System Component		Power per band (uW)	Total Power (10 bands) (uW)
NBSC	Analog filterbank	<10	<140
	Multi-coset processing	<4	
TI's MFSC front-end		N/A	250
Text-dependent speaker verification		<9	<90
User-independent command recognition		<10	<100

sumption increases linearly with the number of active sub-bands, the power consumption per band is approximately 10μW per band.

7.4 Summary

The total system power consumption can be estimated as the sum of the front-end and back-end power consumptions. A summary of the power consumptions of different front-ends and back-ends is given in Table 7.1.

In this chapter, we provided power estimations for two front-end designs: the NBSC and the MFSC feature extraction front-ends, whose power consumptions for 10 band feature extractions are estimated to be under 140μW and 250μW, respectively. For such analog front-end designs, the actual power consumption will vary depending on design techniques and design choices. Nevertheless, these power estimates suggest that, with exist-

ing technologies, the power consumption for such spectral feature extraction front-ends can be kept within a few hundred μW.

The blockwise weighted-DTW algorithm and the adaptive multi-band DNN model are implemented on the Cortex-M0 micro-controller. For both applications, the algorithm occupies less than 40kB of memory and consumes less than 100μW power when 10 bands are active. Generally, the total system power consumption is proportional to the total number of active bands. Average system power consumption can be further reduced by performing adaptive band selection and duty-cycling with the assistance of a VAD.

Chapter 8

Conclusion

In this thesis, we have proposed a new architecture for an instance of resource limited signal processing (Chapter 2). It achieves low-power consumption through data pruning in an early stage to reduce computation complexity in all downstream processing and adaptive processing such that more information is acquired and processed only when it is needed. More specifically, we applied this architecture in the design of low-power voice-command recognition systems.

8.1 Review

8.1.1 Early stage dimension reduction using analog components

Generally, a practical signal processing pipeline can be broken down into two steps: convert the analog input signal into digital forms and perform signal processing. Most conventional systems have adopted the convenient model of first retrieving all available information and then extracting the core information through multi-stage digital processing. The advantage of this approach is generality. Since there is no information loss at the signal retrieval step, the blind data acquisition front-end can be used for a variety of back-end applications and to accommodate different input conditions. The shortcomings of the general purpose design approach include fast sampling and fast rate processing of high-dimensional data.

When there is a limit on power-consumption, computation complexity, response-time or communication bandwidth, an application-specific approach may be considered. The idea is that, for a specific application, the most essential information may be condensed into a low-dimensional representation in a transfer domain. If the relationship between the input signal and the transfer domain signal can be realized using analog components such as filters, integrators, amplifiers, etc, then the overall system complexity can potentially be reduced due to low digitization rate and lower-dimensional signal processing.

8.1.2 Acoustic feature extraction using analog filterbanks and multi-coset sampling

We have demonstrated the benefits of early stage signal dimension reduction for the application of voice-command recognition. In Chapter 3, we have shown through cepstral analysis that the most essential information for speech recognition is captured by the low quefrency cepstral coefficients. We then established a direct relationship between the narrowband speech signal and the low-quefrency coefficients. This relationship allows us to transform the analog speech signal directly to its feature domain using a set of analog filterbank. After filtering, the remaining signal is a multi-band signal whose total occupied bandwidth is significantly smaller than the occupied bandwidth of the original speech signal.

In Chapter 4, we proposed extracting features from the multi-band signal using a non-uniform sampling method such that the samples are obtained at the minimum rate (i.e., twice the total occupied bandwidth). We developed simplified procedures for recovering the narrowband features from the low-rate samples. The output of the feature extraction front-end are sequences of power magnitudes corresponding to the non-zero narrowbands. The feature extraction process adjusts according to different band occupation and computation complexity scales linearly with the number of occupied bands.

8.1.3 Adaptive feature pre-selection

The conventional method of speech recognition takes a non-adaptive approach. It samples speech signals at 16 to 24kHz and extracts a pre-determined set of features such

as the MFCC features in a fixed manner. This approach is neither power efficient nor achieving of the best accuracy. As shown in the text-dependent SV application in Chapter 5, with noisy band subtraction, the adaptive NBSCs yield significantly better recognition accuracy than the fixed MFSC and MFCC features.

In addition, the user-independent command recognition experiments in Chapter 6 indicate that, when there is no background noise, the additional gain in performance by including more features saturates at around 5 bands. When there is background noise, having more features helps improve the performance. Since processing complexity is directly proportional to the number of active bands, the cardinality of the active feature set can be adjusted based on the background noise level to minimize computation. In experiments, the adaptive feature pre-selection scheme, using an average of fewer than 5 high quality bands, achieves comparable performance as the 13-dimensional MFCC feature, which is extracted with a 40 band filterbank.

In short, by adaptively adjusting the number of features and adaptively selecting the high quality features, recognition accuracy can be improved while reducing computation complexity.

8.1.4 Feature adaptive algorithm design

Another key feature of our low-power system is feature adaptive recognition algorithm design. For the applications of text-dependent SV (Chapter 5) and user-independent command recognition (Chapter 6), we have developed the weighted-DTW algorithm and the multi-band DNN model, respectively. These back-end algorithms are designed to achieve high accuracy, low computation complexity, as well as to support adaptive feature inputs. With adaptive feature processing, the overall system complexity scales proportionally with the number of active bands. By optimizing processing complexity based on factors such as the background noise level, the average system power consumption is estimated to be under a few hundred μWs when left perpetually on (Chapter 7), enabling the practical low-power voice-command recognition application, which is the primary concern of this thesis.

8.2 Future work

The proposed system architecture has demonstrated significant improvement in power efficiency for the application of voice-command recognition. Depending on the specific application requirement and operating condition, certain aspects of the system may be further explored to improve recognition accuracy. In addition, the overall architectural design may be applied to a wider range of applications. We will discuss the details below.

8.2.1 Noise spectrum estimation and feature selection

In our experiments, adaptive feature selection based on the background noise spectrum has delivered improved noise robustness. One limitation of our experiments is its small variety of noises such as wind and car noises, and pseudo narrowband noises. The reason we chose these noises is because they are common for our application and they have the characteristics of concentrating within narrowbands, hence enabling simple band-selection algorithms.

Nevertheless, a much larger variety of noises are encountered in daily life and the noise spectrum can be fast varying. Therefore, it would be worthwhile to develop more sophisticated algorithms for real-time noise spectrum estimation and feature selection schemes based on the estimated noise spectrum. Existing work on noise estimation can be found in [44].

Another important parameter for the feature extraction front-end is the number of active bands required to achieve a desirable recognition accuracy. In our implementation, the approach was to estimate the number of required features based on simulation results and hard-code it into our algorithm. It would be helpful to have a framework that quantifies how many feature vectors are needed to achieve a certain recognition accuracy given information such as estimations of the feature in-band SNRs and the quantization noise level.

8.2.2 Coset selection and filter-band support recovery

As discussed in Chapter 4, when the system is prone to quantization and sampling noise, the coset sampler selection procedure becomes critical, because an ill-conditioned $\mathbf{A}_{\text{sub}}$

matrix may result in a large amount of noise amplification. In practical systems with low-resolution ADCs and coarse front-end filters, the quantization and sampling noises are not negligible. In this case, it is important to integrate a well-conditioned sampler selection scheme into system design and compare the effects of different sampler selection schemes on the quality of multi-coset reconstruction.

Even though we assumed the frequency band support (i.e., the filter selection set Y) is known by the multi-coset reconstruction unit, this information is not required for signal reconstruction. The multi-coset sampling technique offers the feature of blind sampling, in which the frequency band support can be recovered if we sample at slightly higher rates [48]. This feature can be useful if your application involves detection of a narrowband sound or signal (e.g., a whistle or a single tone) whose frequency support is unknown.

8.2.3 MFCC feature analysis

As shown in Chapter 3, the NBSCs are designed based on cepstral analysis. The effects of varying feature selection parameters such as the spacing between the narrowbands and the bandwidth of the narrowbands become observable in the cepstral domain. The conventional MFCC features are constructed based on the human auditory system. It would be useful to analyze and interpret the MFCC features in a similar manner and observe the effect of Mel-frequency scaling v.s. equal bandwidth scaling in the cepstral domain. This understanding may help us to design a potentially even better set of acoustic features.

8.2.4 Model adaptation with decision feedback

Our voice-command recognition systems are mainly designed for mobile devices, which are likely to be used by a single user instead of being shared by many people. Therefore, it is useful to adapt the algorithm parameters during its usage in order to provide better recognition accuracy for the designated user.

Depending on whether follow-up actions were taken on the host unit, decision feedbacks can be given to the voice command recognition unit. Parameters can be tuned based on the difference between the correct decision and the decision output. When no decision

feedback is available, parameter adaptation can still be conducted based on a confidence measure of each decision.

8.2.5 Multiple commands recognition

In this thesis, the algorithms were experimented with only a single command. Nevertheless, these algorithms can be easily modified to recognize more than one command. For example, with the speaker-verification algorithm, more templates can be stored when there is more than one command. During prediction, each input signal will be compared with all command templates and the one that gives the minimal distance would be the output decision. As for the user-independent command recognition system, the multi-band model can simply be trained with training samples of multiple commands. The output node will have more than two classes, where each command corresponds to a different class and one more class is assigned to OOV.

As a next step, it would be useful to evaluate the recognition accuracies of both the feature extraction front-end and the recognition back-end for multiple command recognition and characterize the limitations of the system in this regard.

8.2.6 Application to other problems

The proposed architecture design may be applied to other applications with limited resource. An example may be application in image recognition for autonomous vehicles, which demands fast response time, limited signal transmission bandwidth and efficient processing of high-dimension data; or imaging for portable medical devices, which requires low-power consumption and low-computation complexity. In more and more settings, too, low-power sensors constrained by power and communication bandwidth are becoming prevalent. In all of these cases, efficient analog front-ends may be designed to directly extract the essential information for the end application and the feature adaptive processing approach may be used to accommodate different operating conditions.

Appendix A

Multi-coset reconstruction with different filter characteristics

The top plot in each figure shows the frequency spectrum of the multi-band signal after filtering with the simulated analog filterbank of indicated fall-off rates and bandwidths. Three narrowbands are retained after filtering. The bottom three plots in each figure show the power envelope of each narrowband in the time domain. The sampling frequency of the signal is 16kHz. In these experiments, the narrowband bandwidths is set to 400Hz. The 400Hz narrowband signals (obtained either from Nyquist sampling or multi-coset sampling) are then smoothed out by convolving with a coarse filter whose coefficients is $[1, 1, 1, 1]$. The smoothed envelopes of the Nyquist signals and the multi-coset signals are compared. The solid blue curve corresponds to the narrowband signal extracted directly from the original Nyquist rate signal (without band-pass filtering). The dashed red curve corresponds to the multi-coset reconstruction of the Narrowband envelopes.

From the following plots, we can see that better reconstruction accuracy can be obtained with faster fall-off rates of the band-pass filters and longer low-pass filters.

A.1 Multi-coset reconstruction with different band-pass filter fall-off rates

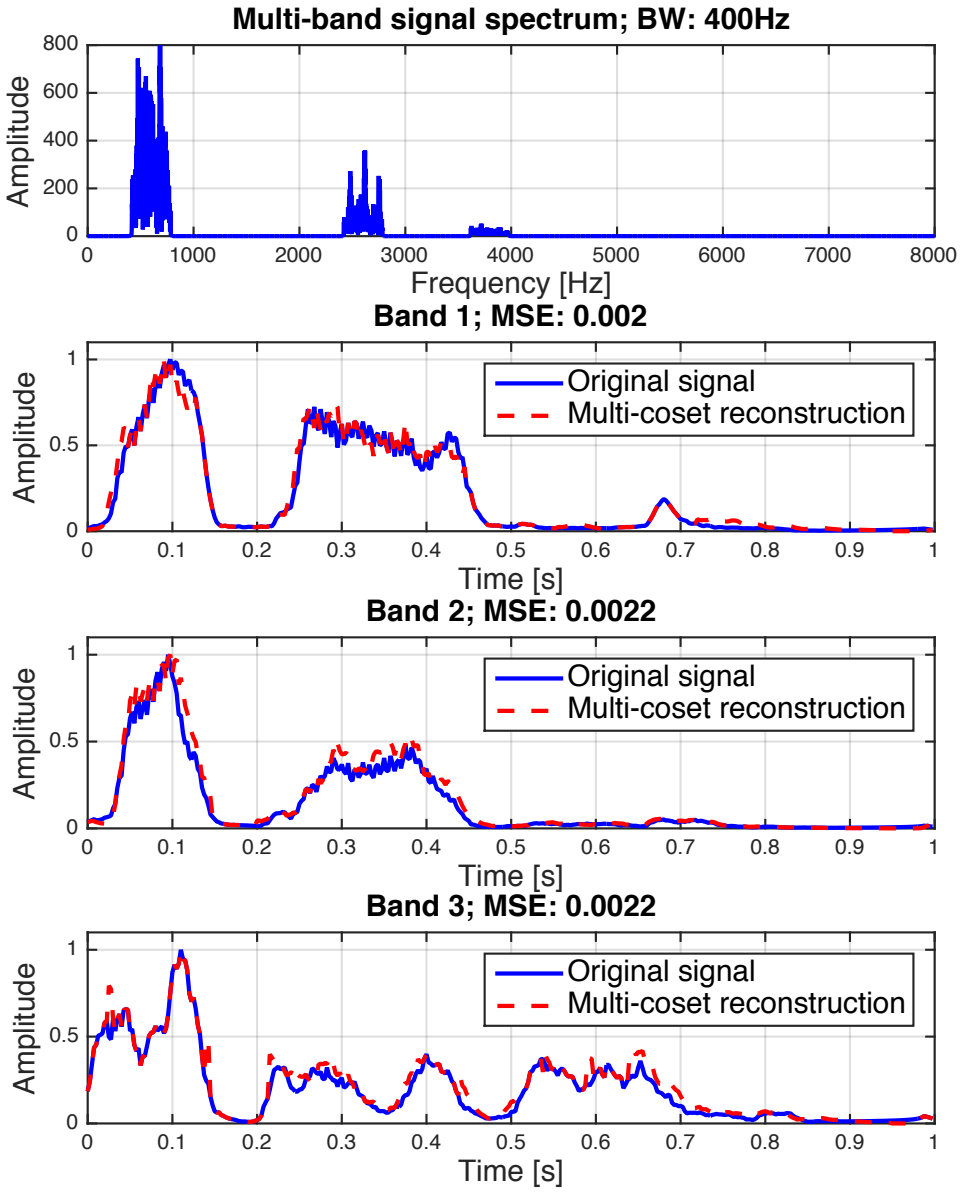


Figure A-1: Multi-coset reconstruction with bandpass filterbank having 3dB cut-off at 20% filter bandwidth. Filterbank band-widths = 400 Hz. High precision low-pass filter for multi-coset reconstruction with $N_{\text{taps}} = 300$.

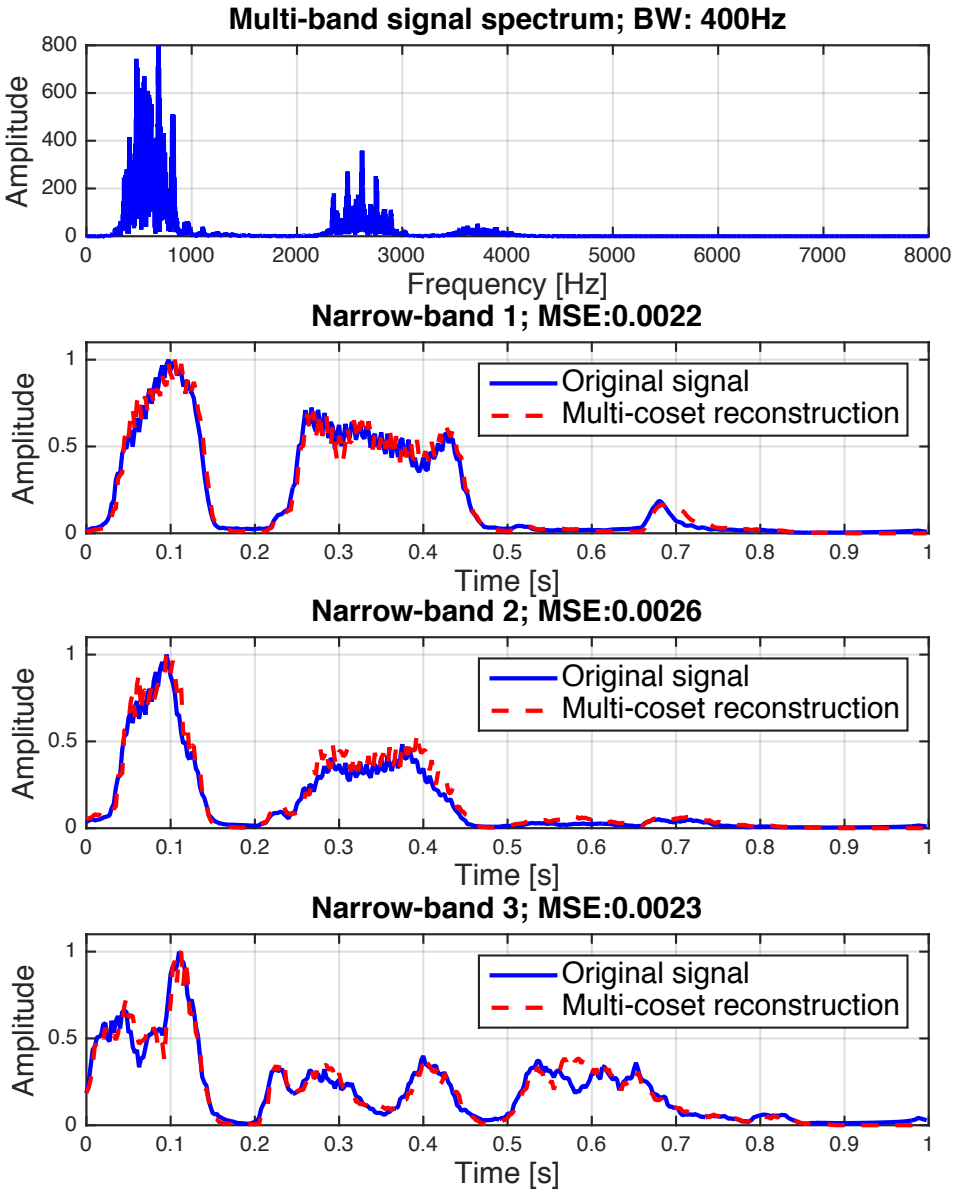


Figure A-2: Multi-coset reconstruction with bandpass filterbank having 3dB cut-off at 50% filter bandwidth. Filterbank band-widths = 400 Hz. High precision low-pass filter for multi-coset reconstruction with $N_{\text{taps}} = 300$.

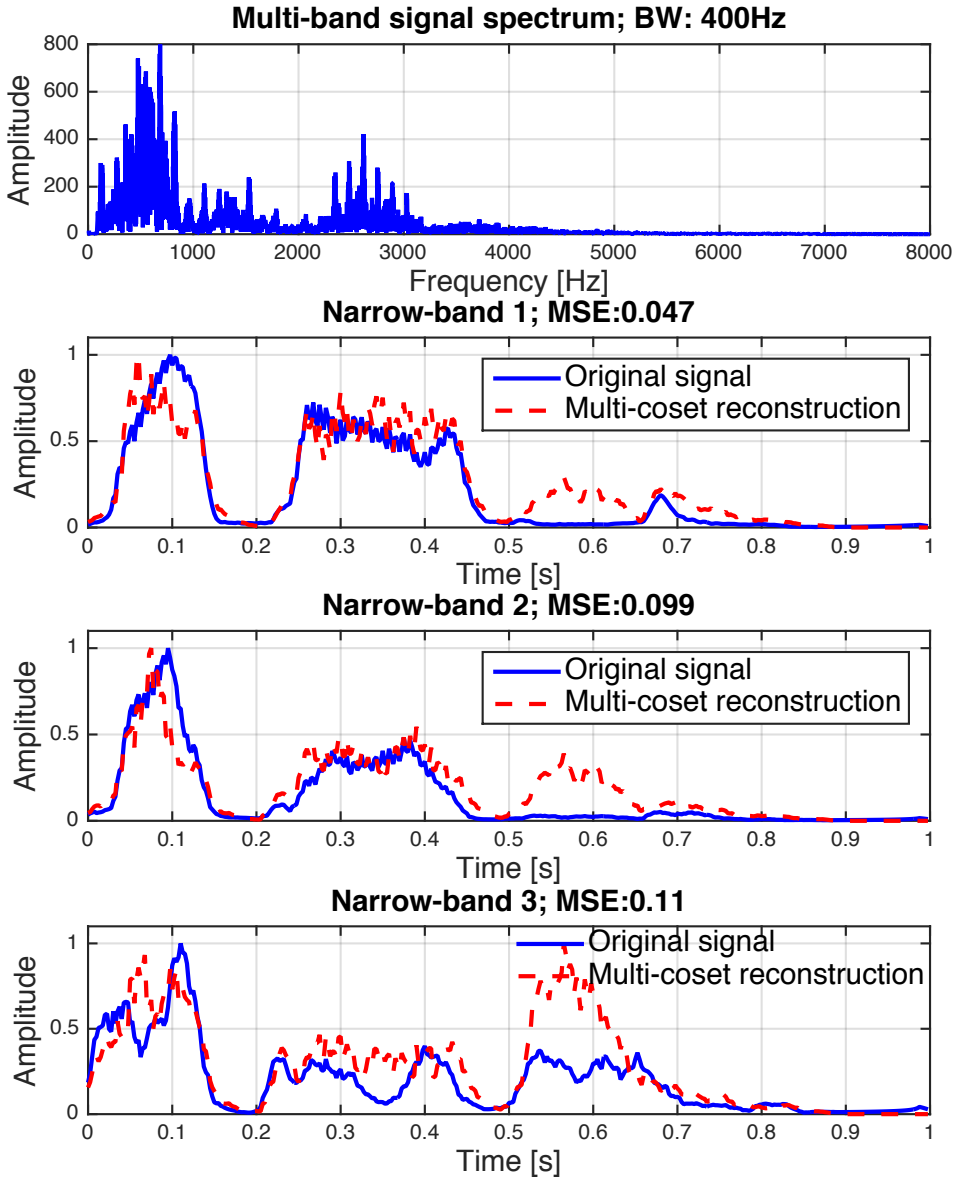


Figure A-3: Multi-coset reconstruction with bandpass filterbank having 3dB cut-off equal to the filter bandwidth. Filterbank band-widths = 400 Hz. High precision low-pass filter for multi-coset reconstruction with $N_{\text{taps}} = 300$.

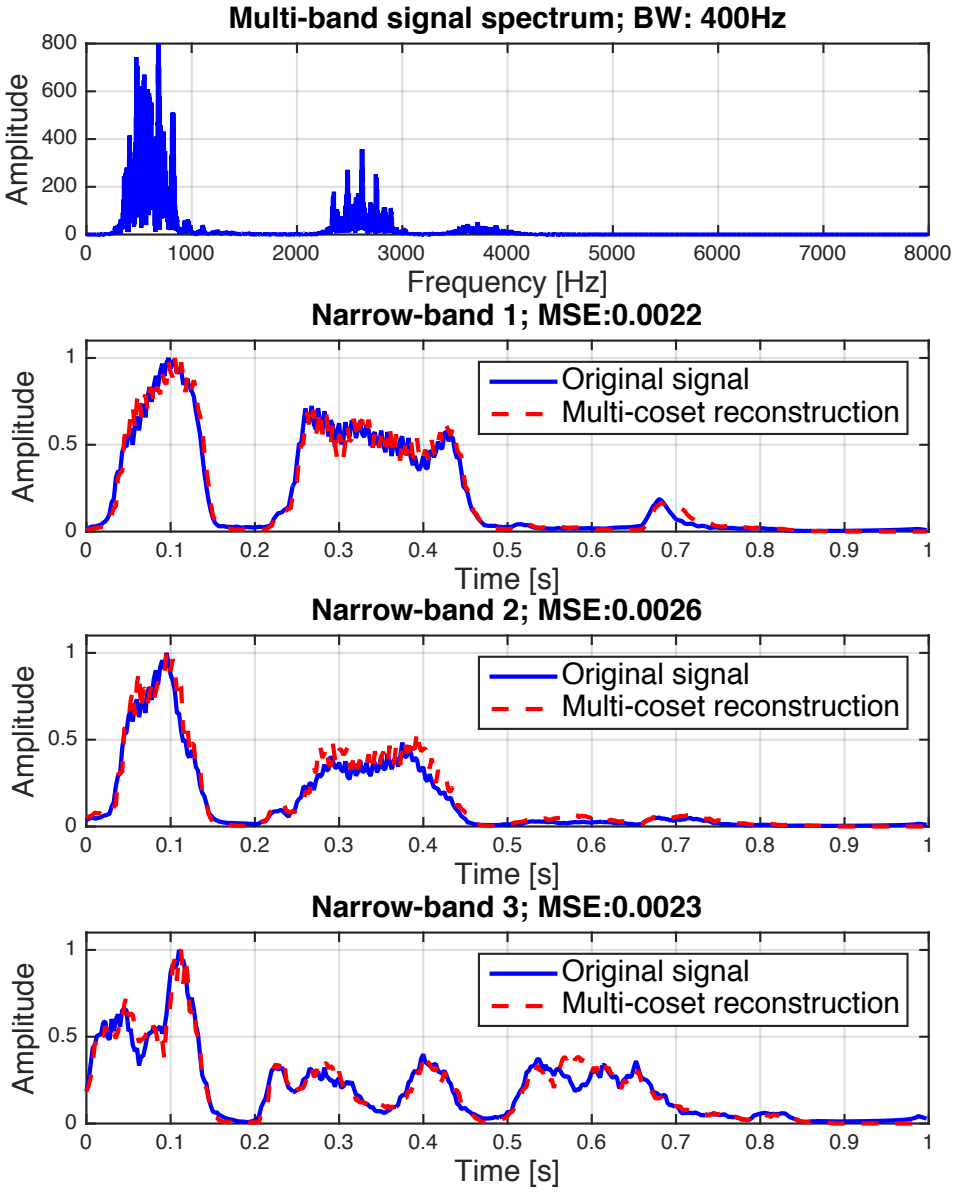


Figure A-4: Multi-coset reconstruction with bandpass filterbank having 3dB cut-off at 50% filter bandwidth. Filterbank band-widths = 400 Hz. High precision low-pass filter for multi-coset reconstruction with $N_{\text{taps}} = 300$.

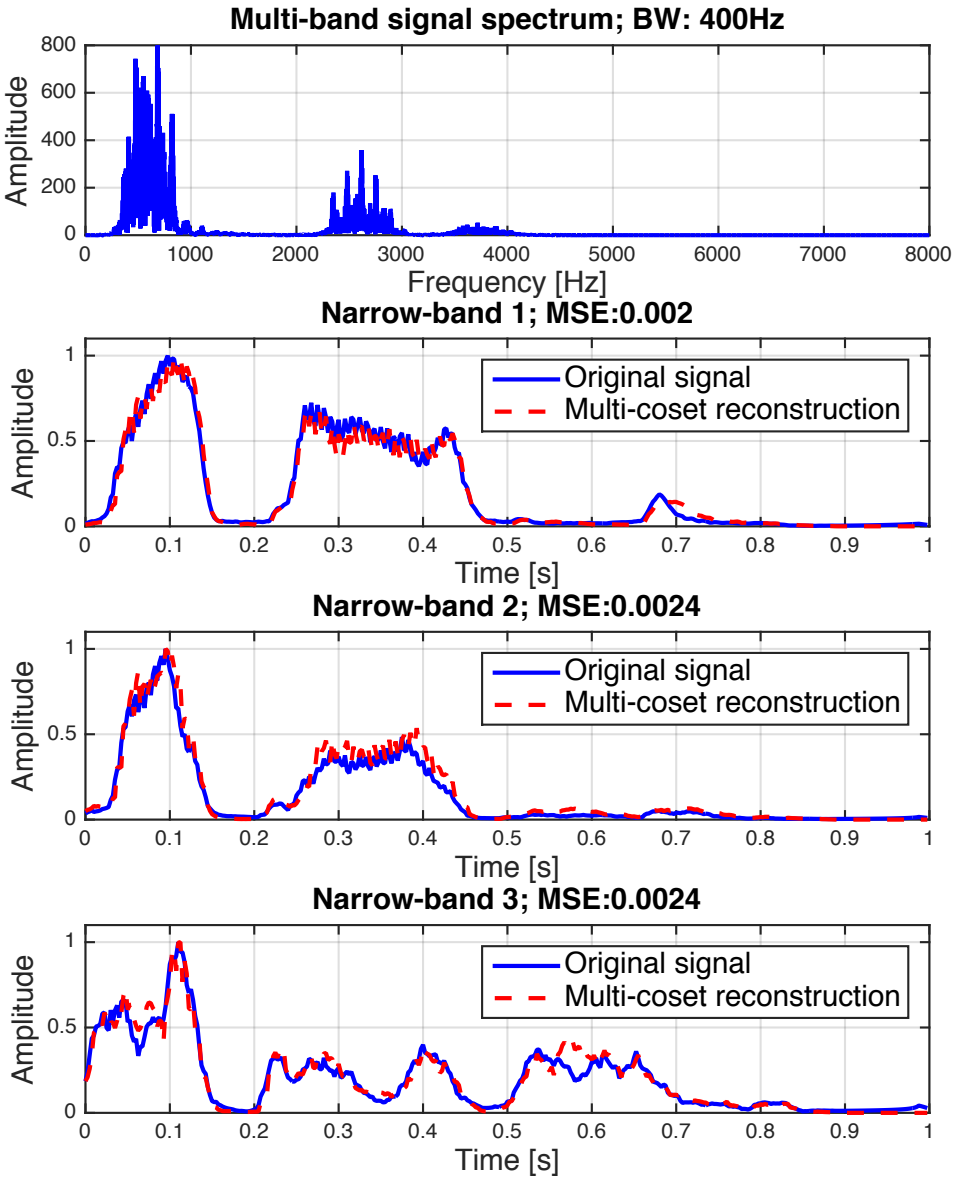


Figure A-5: Multi-coset reconstruction with bandpass filterbank having 3dB cut-off at 50% filter bandwidth. Filterbank band-widths = 400 Hz. Low precision low-pass filter for multi-coset reconstruction with $N_{\text{taps}} = 100$.

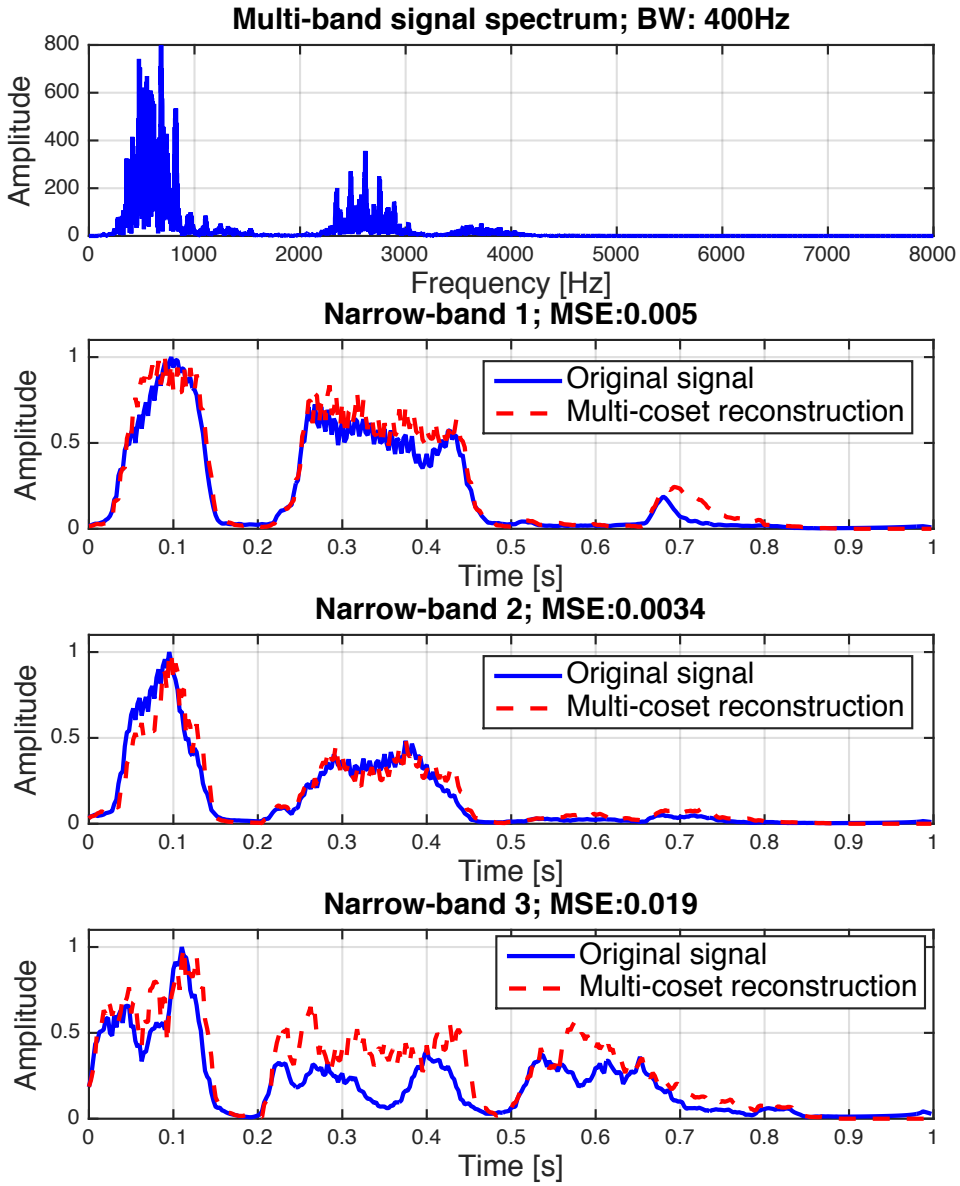


Figure A-6: Multi-coset reconstruction with bandpass filterbank having 3dB cut-off at 50% filter bandwidth. Filterbank band-widths = 400 Hz. Low precision low-pass filter for multi-coset reconstruction with $N_{\text{taps}} = 50$.

Appendix B

Digital filter frequency response

The following plots show the magnitude response of the filters used in simulations in this thesis. The analog filter bank in the system is simulated with a digital filter bank with the indicated filter specs. The magnitude response of the digital low-pass filter used in multi-coset reconstruction is also given here.

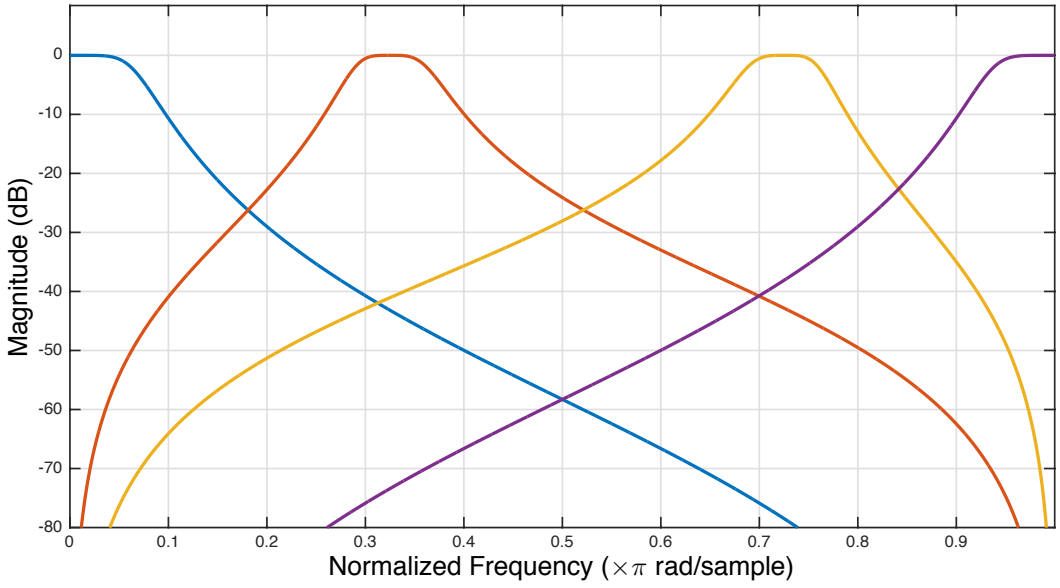


Figure B-1: Magnitude response of experiment bandpass filter having 3dB cut-off at 50% filter bandwidth. Filterbank bandwidth = 400Hz.

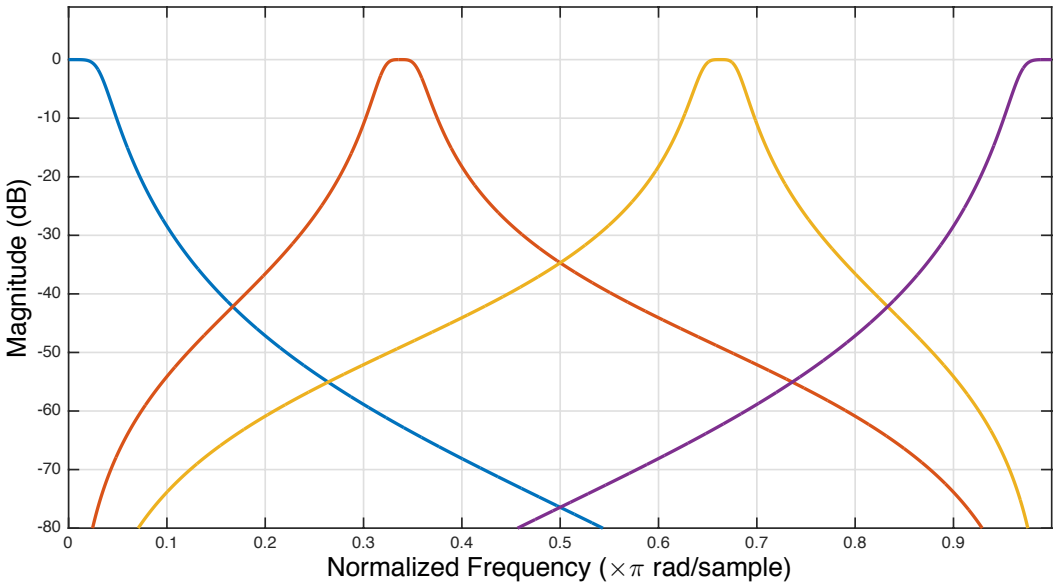


Figure B-2: Magnitude response of experiment bandpass filter having 3dB cut-off at 50% filter bandwidth. Filterbank bandwidths = 200Hz.

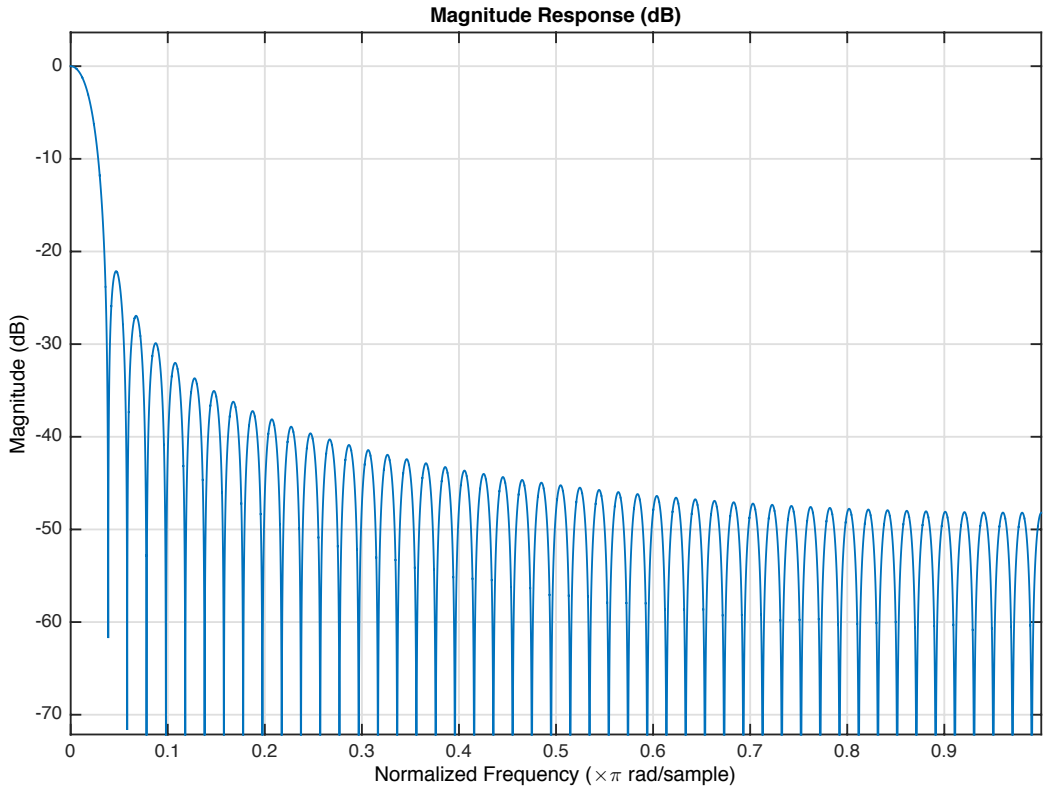


Figure B-3: Magnitude response of experiment lowpass filter having 3dB cut-off at 50% filter bandwidth. Filterbank bandwidth = 400 Hz.

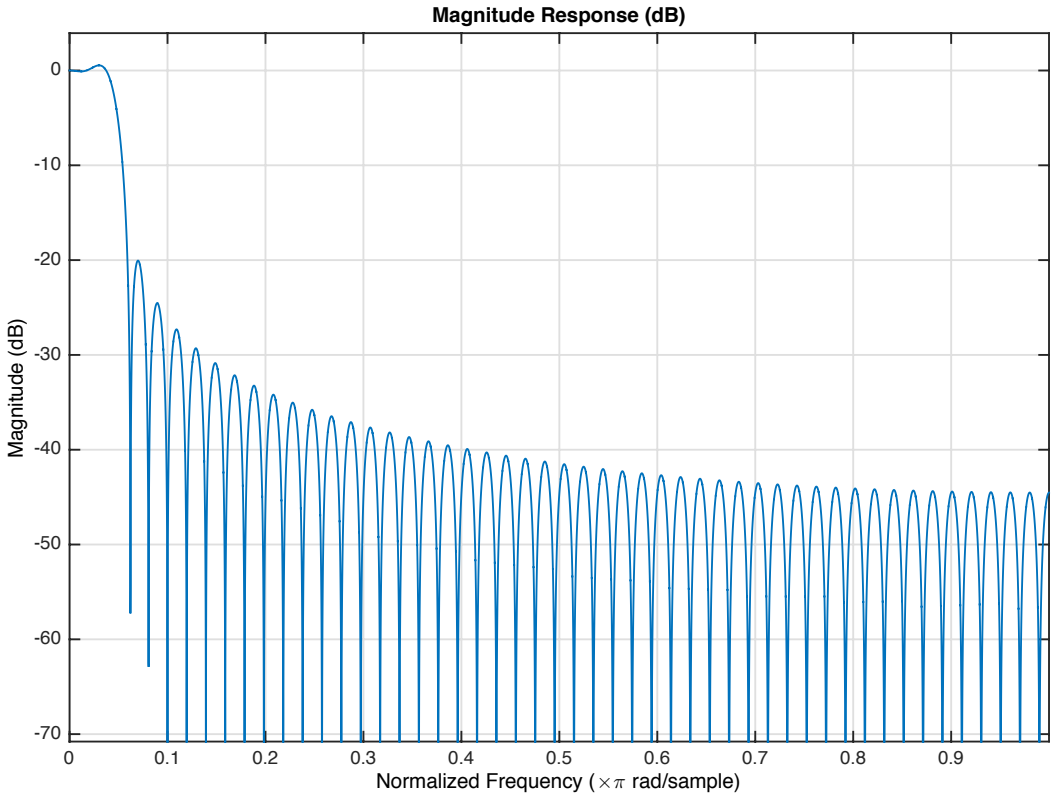


Figure B-4: Magnitude response of experiment lowpass filter having 3dB cut-off at 50% filter bandwidth. Filterbank band-widths = 200 Hz.

Appendix C

Coset sampler selection

Three different coset sampler selection schemes are implemented and their corresponding condition numbers are compared. (1): the optimal scheme. Given M, the dimension of the DFT matrix; P, the dimension of the sub-matrix of the DFT matrix; and the P selected columns (i.e., the active filter set Y), we use a brute force search algorithm to find the optimal row selections (i.e., sampler set $\mathcal{X}$) that yields the minimum condition number. (2): the bunched scheme. For all Y, the bunched scheme always uses the first P samplers (i.e., $\mathcal{X} = \{1, 2, \ldots, P\}$). (3) The co-array method [51]. Given M and P, the co-array method recommends a row selection $\mathcal{X}_{\text{co-array}}$ with cardinality P, for use with any Y of cardinality P. The co-array sampler, $\mathcal{X}_{\text{co-array}}$, yields close to the minimum worst-case condition number among samplers universal to Y of cardinality P.

Table C.1 compares the worst case condition numbers, denoted by κ, for the three sampler selection schemes. Given M, P and Y, $\mathcal{X}$ is chosen using each of the three schemes. The condition number of each resulting sub-matrix is evaluated. This is repeated for all

Table C.1: Condition number comparison of optimal sampler selection, bunched sampler selection and co-array sampler selection.

	Optimal (worst κ)	Bunched (worst κ)	Co-array (worst κ)
M=9; P = 5	3.45	24.27	12
M = 11; P = 5	2.98	69	9.28
M = 11; P = 6	3.14	69	9.5
M = 13; P = 5	2.9	157.8	25.4

column selections Y of dimension P and the worst case condition number for each scheme is given in the table. As shown in the table, the co-array method yields much better worst case condition number than the bunched selection scheme.

The following figures (Figures C-1 through C-4) show a few examples of the three sampler selection schemes for different M and P combinations. The chosen rows and columns are highlighted. The worst case condition number for the bunched selection scheme occurs when Y is also bunched together and the bunched scheme yields better condition number when Y is more spread out. The co-array method yields much better worst case condition number than the bunched selection.

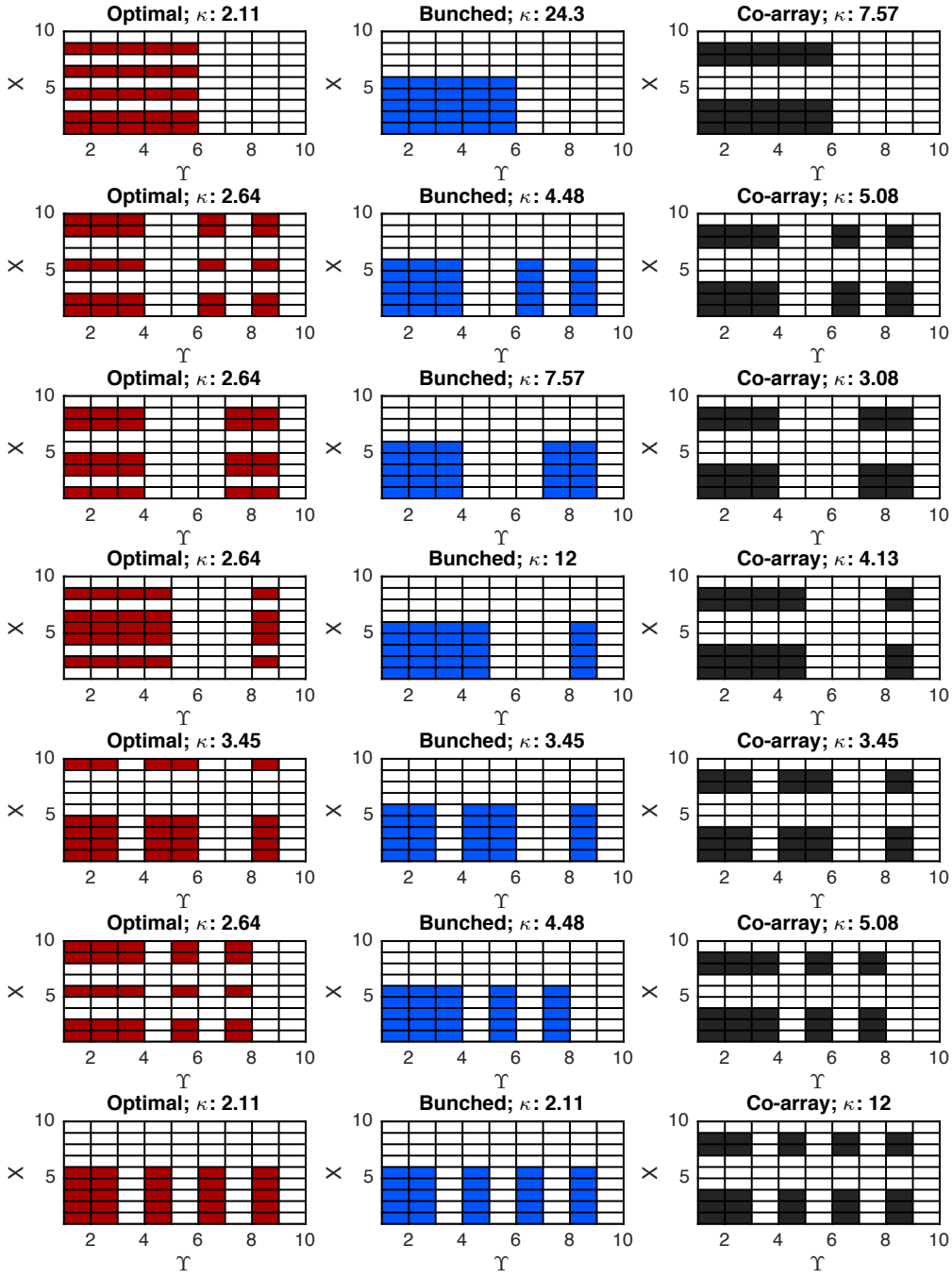


Figure C-1: M = 9, P = 5

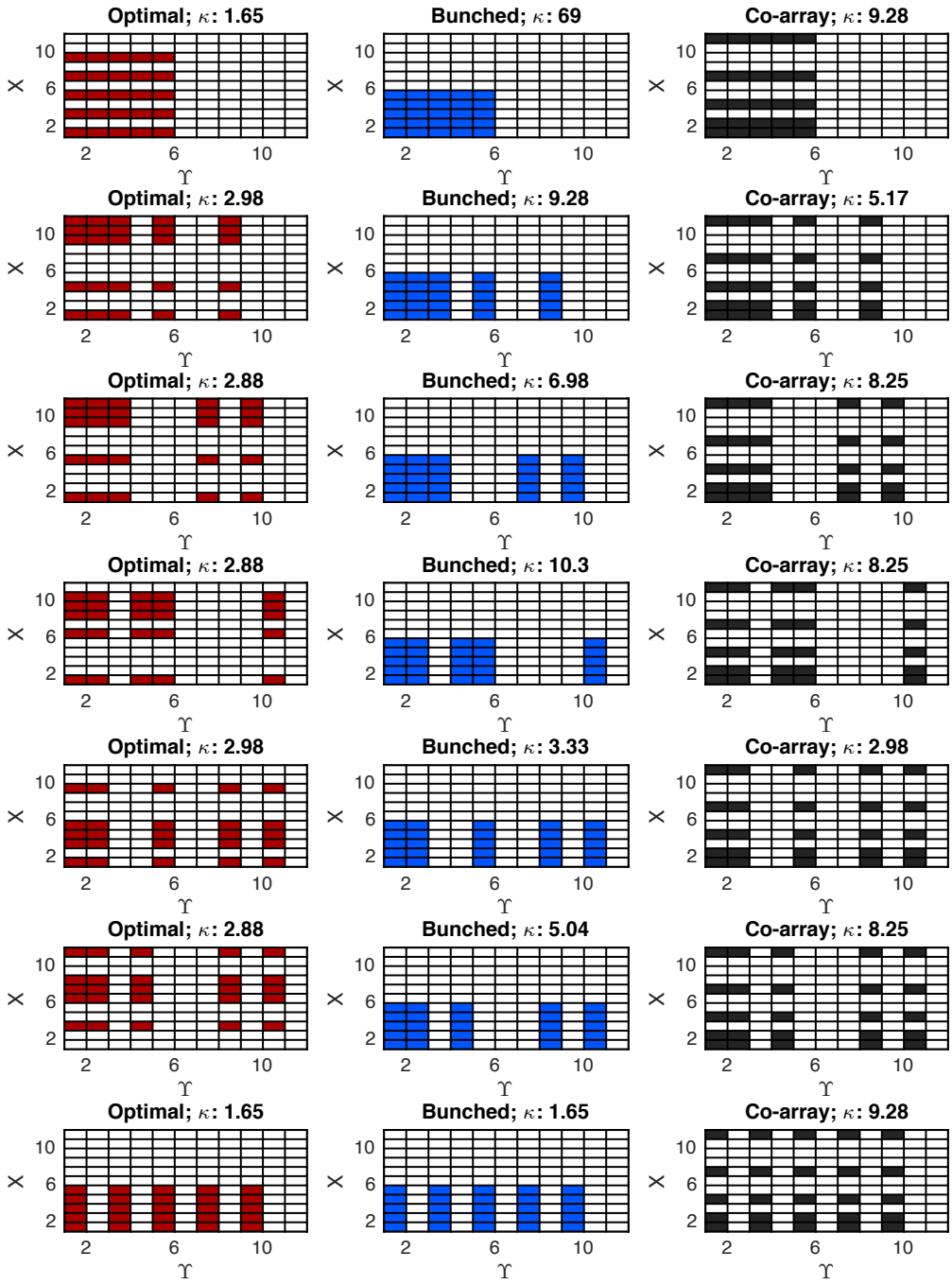


Figure C-2: M = 11, P = 5

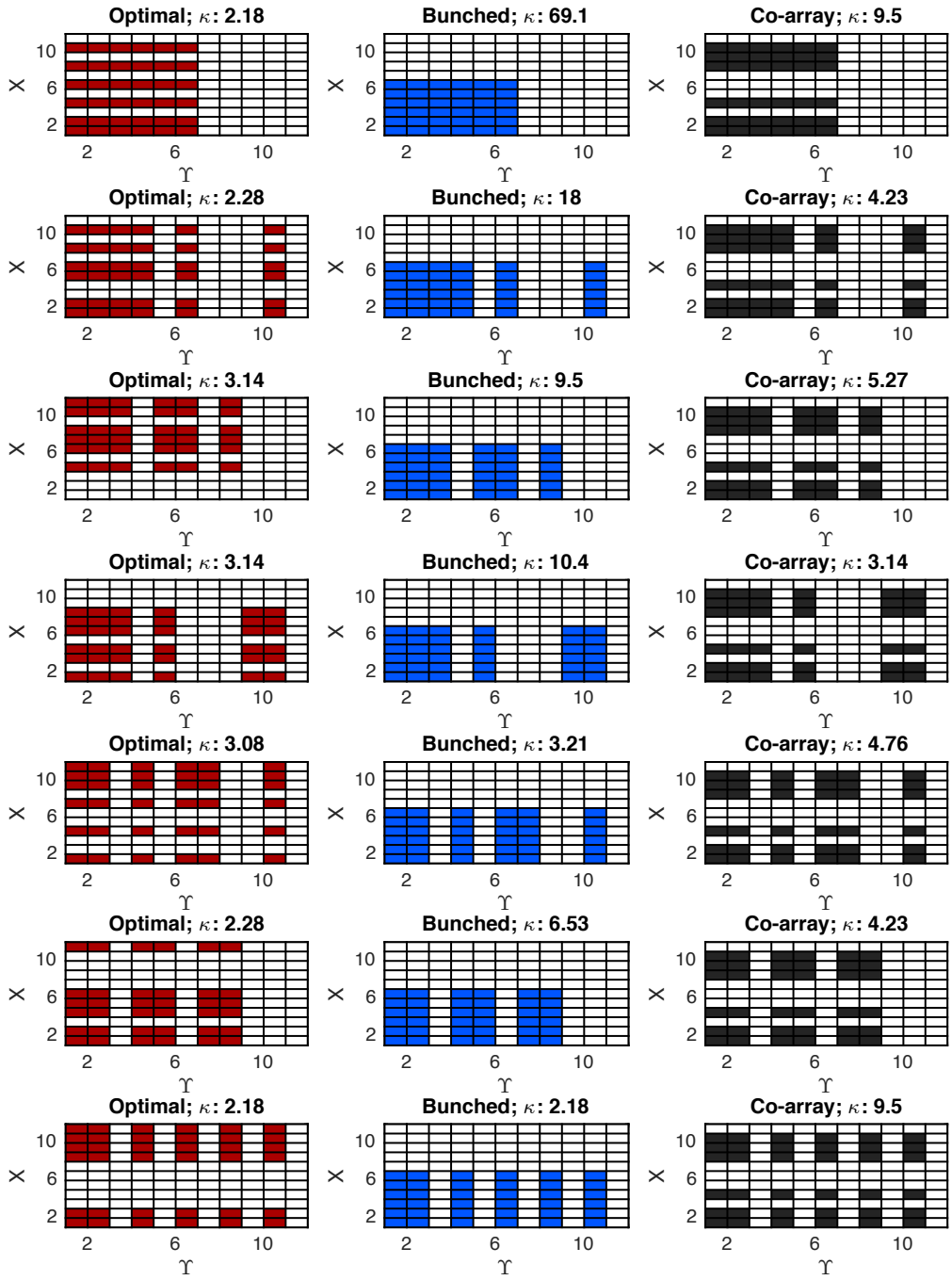


Figure C-3: M = 11, P = 6

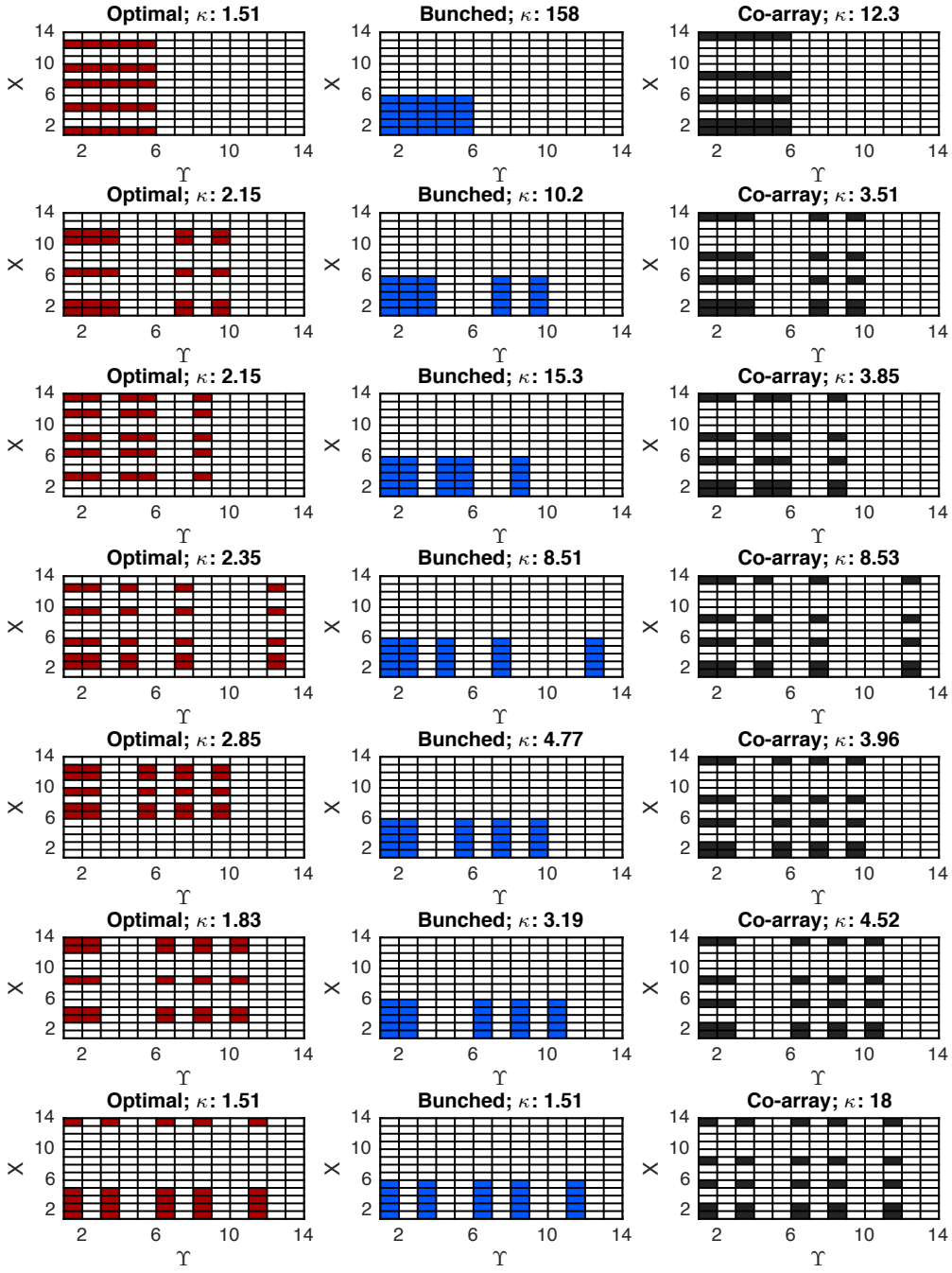


Figure C-4: M = 13, P = 5

Appendix D

Pitch dependent NBSC feature extraction

The following procedure describes the steps taken for pitch dependent NBSC feature extraction, which is used in text-dependent speaker-verification in Chapter 5.

1. Estimate the average fundamental frequency f_0 from each enrollment sample. The final f_0 is determined as the mean of the f_0 estimation from all enrollments. This step is completed off-line during the training procedure. The signal was sampled at 16kHz and its fundamental frequency is obtained using the auto-correlation method [68, 82].

2. Let B be the bandwidth of the speech signal and K, the number of filterbanks. The center frequencies of the filterbanks are at

$$k \times f_0 \times \left\lfloor \frac{B}{f_0 K} \right\rfloor, \quad 1 \leq k \leq K.$$

   These K filters are evenly spaced across the frequency spectrum and B/K is approximately the spacing between adjacent filters. K is selected such that B/K is smaller than a threshold parameter θ_h (i.e., $K \geq B/\theta_h$). The parameter θ_h is dependent on the property of speech (defined in Section 3.4). Usually, $\theta_h = 2$ cycle/kHz, is sufficient. For example, with B = 4 kHz and $\theta_h = 2$ cycle/kHz, $K \geq 8$. The bandwidth of the filters is narrow ($\sim$ 200 Hz) and Section 3.4 discusses the effects of the filter bandwidth.

3. The logarithms of the narrowband filter energies are aggregated and framed to form the narrowband spectral coefficients (NBSCs). The dimension of the feature vector is equal to the number of bands K.

4. Assuming knowledge of the noise spectrum, a subset of the NBSCs, where the SNR is the highest, is retained as features. The remaining bands are discarded.

Appendix E

Pseudo-code for the weighted-DTW algorithm

Algorithm 1 Pseudo-code for computing the accumulative distance matrix of the W-DTW algorithm

$D \leftarrow \mathbf{0}_{I \times J}$ ▷ Accumulative distance matrix
$C \leftarrow \mathbf{0}_{I \times J}$ ▷ Movement count distance matrix
$M \leftarrow \mathbf{0}_{I \times J \times 2}$ ▷ Movement type matrix
$D(1,1) \leftarrow \text{dist}(R(1), T(1))$
for i = 2 to I **do**
    $D(i,1) \leftarrow \text{dist}(R(i), T(1)) + D(i-1,1) + C(i-1,1)|E_R(i)|$
    $C(i,1) \leftarrow C(i-1,1) + 1$
    $M(i,1) \leftarrow (1,0)$
end for

for j = 2 to J **do**
    $D(1,j) \leftarrow \text{dist}(R(1), T(j)) + D(1,j-1) + C(1,j-1)|E_T(j)|$
    $C(1,j) \leftarrow C(1,j-1) + 1$
    $M(1,j) \leftarrow (0,1)$
end for

for i = 2 to I **do**
    for j = 2 to J **do**
        $D(i,j) \leftarrow \text{dist}(R(i), T(j))$ + MINCOST$((i,j))$
    end for
end for

Algorithm 2 Find the best step to be taken at given point (i, j)

function MINCOST((i, j))
    $S^* = \min_{S \in \{(0,0),(0,1),(1,0)\}} \{D((i,j) - S) + \text{PENALTY}((i,j), S)\}$
    $M(i, j) = S^*$ ▷ Update the movement matrix
    if $C((i, j) - S^*) = S^*$ **then** ▷ Update the counter matrix
        $C(i, j) = C((i, j) - S*) + 1$
    else
        $C(i, j) = 0$
    end if
    $c = \text{PENALTY}((i, j), S^*)$ ▷ Incremental distance
    return c
end function

Algorithm 3 Compute the penalty for taking a certain step.

function PENALTY((i,j), S)
    if $S = (1, 0)$ and $M(i - 1, j) = S$ **then**
        $p \leftarrow C(i - 1, j)|E_T(j)|$
    else if $S = (0, 1)$ and $M(i, j - 1) = S$ **then**
        $p \leftarrow C(i, j - 1)|E_R(i)|$
    else
        $p \leftarrow 0$
    end if
    return p
end function

Bibliography

[1] M. Livingstone and D. Hubel, "Segregation of form, color, movement, and depth: anatomy, physiology, and perception," *Science*, vol. 240, no. 4853, pp. 740–749, 1988.

[2] S. Thorpe, D. Fize, C. Marlot, *et al.*, "Speed of processing in the human visual system," *Nature*, vol. 381, no. 6582, pp. 520–522, 1996.

[3] J. B. Allen, "How do humans process and recognize speech?," *IEEE Trans. Speech and Audio Processing*, vol. 2, no. 4, pp. 567–577, 1994.

[4] H. Bourlard and S. Dupont, "A new ASR approach based on independent processing and recombination of partial frequency bands," in *Proc. Int. Conf. Spoken Language*, vol. 1, (Philadelphia, US), p. 426429, October 1996.

[5] H. Hennansky, S. Tibrewala, and M. Pave, "Towards ASR on partially corrupted speech," in *Proc. IEEE Int. Conf. Spoken Language*, vol. 1, (Philadelphia, US), pp. 462–465, October 1996.

[6] J. M. Baker, L. Deng, J. Glass, S. Khudanpur, C.-H. Lee, N. Morgan, and D. O. Shaughnessy, "Developments and directions in speech recognition and understanding, Part 1," *IEEE Signal Processing Magazine*, vol. 26, no. 3, pp. 75–80, 2009.

[7] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[8] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[9] B. Gold, N. Morgan, and D. Ellis, *Speech and audio signal processing: processing and perception of speech and music*. John Wiley & Sons, 2011.

[10] I. Szöke, P. Schwarz, P. Matejka, L. Burget, M. Karafiát, M. Fapso, and J. Cernockỳ, "Comparison of keyword spotting approaches for informal continuous speech.," in *Proc. Int. Cons. on Spoken Language Processing*, (Lisbon, Portugal), pp. 633–636, September 2005.

[11] A. Mandal, K. P. Kumar, and P. Mitra, "Recent developments in spoken term detection: a survey," *Int. Journal of Speech Technology*, vol. 17, no. 2, pp. 183–198, 2014.

[12] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. Prentice Hall, 2008.

[13] Y.-K. Choi, K. You, J. Choi, and W. Sung, "A real-time FPGA-based 20 000-word speech recognizer with optimized DRAM access," *IEEE Trans. Circuits and Systems I: Regular Papers*, vol. 57, no. 8, pp. 2119–2131, 2010.

[14] J. Choi, K. You, and W. Sun, "An FPGA implementation of speech recognition with weighted finite state transducers," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, (Dallas, US), March 2010.

[15] K. You, J. Choi, and W. Sung, "Flexible and expandable speech recognition hardware with weighted finite state transducers," *Journal of Signal Processing Systems*, vol. 66, no. 3, pp. 235–244, 2012.

[16] P. J. Bourke and R. A. Rutenbar, "A low-power hardware search architecture for speech recognition," in *Proc. Int. Conf. on Spoken Language Processing*, (Dallas, US), pp. 2102–2105, March 2008.

[17] J. R. Johnston and R. A. Rutenbar, "A high-rate, low-power, ASIC speech decoder using finite state transducers," in *Proc. Int. Conf. on Application-Specific Systems Architectures and Processors*, (Delft, NL), pp. 77–85, July 2012.

[18] G. He, T. Sugahara, Y. Miyamoto, T. Fujinaga, H. Noguchi, S. Izumi, H. Kawaguchi, and M. Yoshimoto, "A 40 nm 144 mW VLSI processor for real-time 60-k word contin-

uous speech recognition," *IEEE Trans. Circuits and Systems I: Regular Papers*, vol. 59, no. 8, pp. 1656–1666, 2012.

[19] O. A. Bapat, P. D. Franzon, and R. M. Fastow, "A generic and scalable architecture for a large acoustic model and large vocabulary speech recognition accelerator using logic on memory," *IEEE Trans. on Very Large Scale Integration Systems*, vol. 22, no. 12, pp. 2701–2712, 2014.

[20] M. Price, J. Glass, and A. Chandrakasan, "A 6 mW, 5,000-Word Real-Time Speech Recognizer Using WFST Models," *IEEE Journal of Solid-State Circuits*, vol. 50, pp. 102–112, January 2015.

[21] K. M. Badami, S. Lauwereins, W. Meert, and M. Verhelst, "A 90 nm CMOS, power-proportional acoustic sensing frontend for voice activity detection," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 1, pp. 291–302, 2016.

[22] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[23] R. Rojas, *Neural networks:A Systematic Introduction*. Springer, 1996.

[24] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[25] A.-r. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[26] Y. Zhang and J. R. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," in *Proc. IEEE Workshop on Automatic Speech Recognition & Understanding*, (Merano, IT), November 2009.

[27] A. S. Park and J. R. Glass, "Unsupervised pattern discovery in speech," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 186–197, 2008.

[28] R. C. Rose and D. B. Paul, "A hidden Markov model based keyword recognition system," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, (Albuquerque, US), April 1990.

[29] J. R. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden Markov modeling for speaker-independent word spotting," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, (Glasgow, UK), May 1989.

[30] M.-C. Silaghi, "Spotting subsequences matching an HMM using the average observation probability criteria with application to keyword spotting.," in *Proc. National Conf. Artificial Intelligence*, vol. 20, (Pittsburgh, US), p. 1118, July 2005.

[31] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, (Florence, IT), May 2014.

[32] I. Szöke, P. Schwarz, P. Matějka, L. Burget, M. Karafiát, and J. Černockỳ, "Phoneme based acoustics keyword spotting in informal continuous speech," in *Proc. Text, Speech and Dialogue*, (Karlovy Vary, CZ), September 2005.

[33] S. A. Dudani, "The distance-weighted k-nearest-neighbor rule," *IEEE Trans. Systems, Man and Cybernetics*, no. 4, pp. 325–327, 1976.

[34] A. Ganapathiraju, J. E. Hamaker, and J. Picone, "Applications of support vector machines to speech recognition," *IEEE Trans. Signal Processing*, vol. 52, no. 8, pp. 2348–2355, 2004.

[35] R. H. Walden, "Analog-to-digital converter survey and analysis," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 4, pp. 539–550, 1999.

[36] S. Okawa, E. Bocchieri, and A. Potamianos, "Multi-band speech recognition in noisy environments," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 2, (Seattle, US), May 1998.

[37] H. Bourlard and S. Dupont, "A new ASR approach based on independent processing and recombination of partial frequency bands," in *Proc. IEEE Int. Conf. Spoken Language Processing*, (Philadelphia, US), pp. 426–429, October 1996.

[38] L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams, *et al.*, "Recent advances in deep learning for speech research at Microsoft," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, (Vancouver, Canada), May 2013.

[39] R. W. Schafer, "Homomorphic systems and cepstrum analysis of speech," in *Springer Handbook of Speech Processing*, pp. 161–180, Springer, 2008.

[40] L. R. Rabiner and R. W. Schafer, *Theory and application of digital speech processing*. Prentice hall, 2009.

[41] A. V. Oppenheim and R. W. Schafer, "From frequency to quefrency: A history of the cepstrum," *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 95–106, 2004.

[42] T. M. Elliott and F. E. Theunissen, "The modulation transfer function for speech intelligibility," *PLOS Computational Biology*, vol. 5, no. 3, 2009.

[43] H. Landau, "Necessary density conditions for sampling and interpolation of certain entire functions," *Acta Mathematica*, vol. 117, no. 1, pp. 37–52, 1967.

[44] B. Rumberg, D. W. Graham, V. Kulathumani, and R. Fernandez, "Hibernets: Energy-efficient sensor networks using analog signal processing," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 3, pp. 321–334, 2011.

[45] B. Rumberg and D. W. Graham, "A low-power and high-precision programmable analog filter bank," *IEEE Trans. Circuits and Systems II: Express Briefs*, pp. 234–238, 2012.

[46] Y.-P. Lin and P. Vaidyanathan, "Periodically nonuniform sampling of bandpass signals," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol. 45, no. 3, pp. 340–351, 1998.

[47] M. Mishali and Y. C. Eldar, "Blind multiband signal reconstruction: Compressed sensing for analog signals," *IEEE Trans. Signal Processing*, vol. 57, no. 3, pp. 993–1009, 2009.

[48] P. Feng and Y. Bresler, "Spectrum-blind minimum-rate sampling and reconstruction of multiband signals," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 3, (Atlanta, US), pp. 1688–1691, May 1996.

[49] B. Foster and C. Herley, "Exact reconstruction from periodic nonuniform samples," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 2, (Detroit, US), pp. 1452–1455, IEEE, May 1995.

[50] B. Alexeev, J. Cahill, and D. G. Mixon, "Full spark frames," *Journal of Fourier Analysis and Applications*, vol. 18, no. 6, pp. 1167–1194, 2012.

[51] J. D. Krieger, Y. Kochman, and G. W. Wornell, "Design and analysis of multi-coset arrays," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, (Vancouver, CA), pp. 3781–3785, May 2013.

[52] A. Moffet, "Minimum-redundancy linear arrays," *IEEE Trans. Antennas and Propagation*, vol. 16, pp. 172–175, 1968.

[53] D. A. Reynolds and W. M. Campbell, "Text-independent speaker recognition," in *Springer Handbook of Speech Processing*, pp. 763–782, Springer, 2008.

[54] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Communication*, vol. 52, pp. 12–40, 2010.

[55] M. Hébert, "Text-dependent speaker recognition," in *Springer handbook of speech processing*, 2008.

[56] D. E. Sturim, D. A. Reynolds, R. B. Dunn, and T. F. Quatieri, "Speaker verification using text-constrained Gaussian mixture models," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2002.

[57] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1, pp. 19–41, 2000.

[58] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[59] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pp. 4052–4056, May 2014.

[60] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–13, 2015.

[61] T. Matsui and S. Furui, "Concatenated phoneme models for text-variable speaker recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, (Minneapolis, US), 1993.

[62] A. E. Rosenberg, C.-H. Lee, and F. K. Soong, "Sub-word unit talker verification using hidden Markov models," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, (Albuquerque, US), 1990.

[63] 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*, April 1980.

[64] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Trans. Acoustics, Speech and Signal Processing*, march 1981.

[65] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoustics, Speech and Signal Processing*, April 1978.

[66] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Trans. Acoustics, Speech and Signal Processing*, 1975.

[67] A. Lee and J. Glass, "A comparison-based approach to mispronunciation detection," in *Proc. IEEE Spoken Language Technology Workshop*, (Miami, US), pp. 382–387, December 2012.

[68] L. Rabiner, M. J. Cheng, A. E. Rosenberg, C. McGonegal, *et al.*, "A comparative performance study of several pitch detection algorithms," *IEEE Trans. Acoustics, Speech and Signal Processing*, pp. 399–418, 1976.

[69] J. G. Wilpon, L. Rabiner, C.-H. Lee, and E. Goldman, "Automatic recognition of keywords in unconstrained speech using hidden Markov models," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 38, no. 11, pp. 1870–1878, 1990.

[70] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. Int. Conf. Machine Learning*, (San Francisco, US), pp. 282–289, 2001.

[71] N. Littlestone and M. K. Warmuth, "The weighted majority algorithm," *Information and computation*, vol. 108, no. 2, pp. 212–261, 1994.

[72] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proc. Computational learning theory*, (Barcelona, Spain), March 1995.

[73] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," *Journal-Japanese Society For Artificial Intelligence*, vol. 14, no. 771-780, p. 1612, 1999.

[74] H. Schwenk and Y. Bengio, "Training methods for adaptive boosting of neural networks for character recognition," *Advances in Neural Information Processing Systems*, vol. 10, pp. 647–653, 1998.

[75] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[76] C. D. Salthouse and R. Sarpeshkar, "A practical micropower programmable bandpass filter for use in bionic ears," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 1, pp. 63–70, 2003.

[77] L. Pylarinos and K. Phang, "Low-voltage programmable g m-C filter for hearing aids using dynamic gate biasing," in *Proc. IEEE Int. Symposium on Circuits and Systems*, (Kobe, Japan), pp. 1984–1987, May 2005.

[78] R. J. W. Wang, R. Sarpeshkar, M. Jabri, and C. Mead, "A low power analog front-end module for cochlear implants," in *Proc. World Congress of Otorhinolaryngology Head and Neck Surgery*, (Sydney, Australia), March 1997.

[79] R. Sarpeshkar, R. F. Lyon, and C. Mead, "A low-power wide-dynamic-range analog VLSI cochlea," in *Neuromorphic Systems Engineering*, pp. 49–103, Springer, 1998.

[80] D. W. Graham, P. D. Smith, R. Ellis, R. Chawla, and P. E. Hasler, "A programmable bandpass array using floating-gate elements," in *Proc. IEEE Int. Symposium on Circuits and Systems*, vol. 1, (Vancouver, CA), pp. I–97, May 2004.

[81] F. M. Yaul and A. P. Chandrakasan, "A 10 bit SAR ADC with data-dependent energy reduction using lsb-first successive approximation," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 12, pp. 2825–2834, 2014.

[82] M. M. Sondhi, "New methods of pitch extraction," *IEEE Trans. Audio and Electroacoustics*, vol. 16, no. 2, pp. 262–266, 1968.