VOICE TRANSFORMATION AND DEVELOPMENT OF RELATED SPEECH ANALYSIS TOOLS FOR TURKISH

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÖZGÜL SALOR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JANUARY 2005

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. İsmet Erkmen
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. Mübeccel Demirekler
Supervisor

Examining Committee Members

Prof. Dr. Zafer Ünver	(METU, EE)	
Prof. Dr. Mübeccel Demirekler	(METU, EE)	
Assoc. Prof. Dr. Tolga Çiloğlu	(METU, EE)	
Assoc. Prof. Dr. Engin Tuncer	(METU, EE)	
Assist. Prof. Dr. Gökhan İlk	(Ankara University, EE)	

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Özgül Salor

Signature :

ABSTRACT

VOICE TRANSFORMATION AND DEVELOPMENT OF RELATED SPEECH ANALYSIS TOOLS FOR TURKISH

Salor, Özgül

Ph. D., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Mübeccel Demirekler

January 2005, 132 pages

In this dissertation, new approaches in the design of a voice transformation (VT) system for Turkish are proposed. Objectives in this thesis are two-fold. The first objective is to develop standard speech corpora and segmentation tools for Turkish speech research. The second objective is to consider new approaches for VT.

A triphone-balanced set of 2462 Turkish sentences is prepared for analysis. Audio corpus of 100 speakers, each uttering 40 sentences out of the 2462-sentence set, is used to train a speech recognition system designed for English. This system is ported to Turkish to obtain a phonetic aligner and a phoneme recognizer. The triphone-balanced sentence set and the phonetic aligner are used to develop a speech corpus for VT.

A new voice transformation approach based on Mixed Excitation Linear Prediction (MELP) speech coding framework is proposed. Multi-stage vector quantization of MELP is used to obtain speaker-specific line-spectral frequency (LSF) codebooks for source and target speakers. Histograms mapping the LSF

spaces of source and target speakers are used for transformation in the baseline system. The baseline system is improved by a dynamic programming approach to estimate the target LSFs. As a second approach to the VT problem, quantizing the LSFs using k-means clustering algorithm is applied with dimension reduction of LSFs using principle component analysis. This approach provides speaker-specific codebooks out of the speech corpus instead of using MELP's pre-trained LSF codebook. Evaluations show that both dimension reduction and dynamic programming improve the transformation performance.

Keywords: voice transformation, phonetic aligner, phoneme recognizer, phonetic alphabet, speech corpus

ÖZ

SES ÇEVİRME VE TÜRKÇE İÇİN İLGİLİ KONUŞMA ANALİZİ ARAÇLARI GELİŞTİRME

Salor, Özgül
Doktora, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Mübeccel Demirekler

Ocak 2005, 132 sayfa

Bu tezde, Türkçe ses dönüştürme (SD) sistemi tasarlamak için yeni yaklaşımlar önerilmiştir. Tezde iki amaç vardır. İlk amaç, SD sistemi geliştirmek için gerekli olan, konuşma veritabanını ve bölütleme araçlarını geliştirmek, ikinci amaç ise SD için yeni yaklaşımlar denemektir.

Analiz için, 2462 Türkçe tümce içeren üçlü-ses dengeli bir küme hazırlanmıştır. Her konuşmacının 2462 tümce kümesinden 40'ar tümce seslendirdiği, 100 konuşmacılık bir veritabanı, İngilizce için tasarlanmış bir konuşma tanıma sistemini eğitmek üzere kullanılmıştır. Bu eğitim sonucu, sistem Türkçe için çalışır duruma getirilmiş ve fonetik hizalayıcı ile fonem tanıyıcı araçlar elde edilmiştir. Üçlü-ses dengeli tümce kümesi ve fonetik hizalayıcı, SD sistemi için ses veritabanı geliştirilmesinde kullanılmıştır.

SD için yeni oluşturulan yöntemlerin ilki, MELP (Mixed Excitation Linear Prediction) konuşma kodlama algoritmasına dayanmaktadır. MELP'in çok katlı vektör nicemlemesi kaynak ve hedef konuşmacılar için konuşmacıya özgü çizgisel spektrum frekansı (ÇSF) kod çizelgelerinin oluşturulmasında kullanılmıştır. Taban sistemde dönüştürme için, kaynak ve hedef konusmacıların

ÇSF uzaylarını eşleyen histogramlar kullanılmıştır. Taban sistemin hedef ÇSF'leri kestirmesi dinamik programlama yaklaşımıyla geliştirilmiştir. İkinci bir yaklaşım olarak, ÇSF'leri nicemlemek için, ana bileşenler analizi ile ÇSF boyutu düşürülerek, k-ortalama topaklama algoritması kullanılmıştır. Bu yaklaşım, MELP'in önceden eğitilmiş kod çizelgesini kullanmak yerine, konuşmacıya özgü kod çizelgeleri oluşturulmasını sağlamıştır. Nesnel ve öznel değerlendirmeler, boyut düşürmenin ve dinamik programlamanın dönüştürme başarımını arttırdığını göstermiştir.

Anahtar Kelimeler: Ses çevirme, fonetik hizalayıcı, fonem tanıyıcı, fonetik alfabe, konuşma veritabanı

To my parents and Özge,

ACKNOWLEDGMENTS

I am greatly indebted to my supervisor Prof. Dr. Mübeccel Demirekler for her invaluable guidance and encouragement throughout my graduate studies. Her contributions in every stage of this research are gratefully acknowledged. Her guidance and support were critical to the formulations developed in this dissertation.

I would also like to thank Dr. Bryan Pellom for advising me during the 17-month period I spent at the Center for Spoken Language Research (CSLR) of University of Colorado at Boulder, USA. I am grateful to him not only for his technical guidance but also for his hospitality at CSLR. I would also like to thank Prof. Dr. John Hansen at CSLR for spending his time to give me technical suggestions and comments on my work.

I am grateful to my thesis committee member Assoc. Prof. Dr. Tolga Çiloğlu for keeping in touch when I was at CSLR, sending me the resources I required from the Middle East Technical University (METU), and keeping track of the speech collection at METU, which were all critical for my thesis. I also wish to acknowledge Çağla Önür, and Yücel Özbek for conducting the speech collection when I was at CSLR. I would also like to acknowledge Assoc. Prof. Dr. Tolga Çiloğlu again for providing the environment and spending his time for our voice conversion data collections at METU.

I would like to thank my thesis committee member Assist. Prof. Dr. Gökhan İlk for his valuable criticism and suggestions and lively discussions, which have contributed to this work.

I also wish to thank TÜBİTAK, the Scientific and Technical Research Council of Turkey, for supporting my work through a combined doctoral scholarship. I would like to thank Prof. Dr. Erol Kocaoğlan and Ayşe Ataş at

TÜBİTAK for their continued support.

I would like to thank my friends Umut Orguner and Emre Özkan for providing their voices for my voice transformation experiments and struggling with long voice recording sessions with patience. I would like to thank my friend Ece Güran for her understanding and support whenever I needed. I would also like to thank my office-mate Yücel Özbek for helping me gratuitously at the most difficult times.

Finally, I wish to express my deepest gratitude to my parents, Doğan and Tülay Salor, who have encouraged me the entire way and supported me all my life. I am also deeply grateful to my sister, Özge, who has shared all my concerns and has been with me through it all.

TABLE OF CONTENTS

ABSTRACT iv

ÖZ vi

DEDICATION viii

ACKNOWLEDGMENTS viii

TABLE OF CONTENTS xi

LIST OF TABLES xv

LIST OF FIGURES xvii

CHAPTER

1 INTRODUCTION 1

1.1 Motivation 2

1.2 Contributions 4

1.3 Outline of the Thesis 5

2 AN OVERVIEW OF VOICE TRANSFORMATION SYSTEMS 6

2.1 Basic Properties of the Speech Signal 6

2.1.1 Speech Production 6

2.1.2 Voice Individuality 8

2.1.2.1 Speaker recognition by humans . . 10

2.1.2.2 Relative effects of segmental and suprasegmental cues on speaker individuality and human perception . 11

2.2 An Overview of Voice Transformation Systems 12

2.2.1 Speech Corpus 13

2.2.2 Speech Modelling 13

2.2.3 Transformation Methods 15

2.3 Evaluation of VT Systems 18

2.3.1 Objective Evaluation 18
2.3.2 Subjective Evaluation 19
3 SPEECH CORPORA AND RECOGNITION TOOLS DEVELOPED FOR TURKISH SPEECH RESEARCH 21
3.1 Introduction 21
3.2 Phonetic Alphabet for Turkish 22
3.3 Design of a Phonetically Rich Audio Corpus 23
3.3.1 Construction of a Phonetically Rich Sentence Set 25
3.3.2 Collecting the Audio Corpus 28
3.4 Speech Recognition Tools Developed for Turkish 30
3.4.1 The Sonic Continuous Speech Recognizer . . . 30
3.4.2 Systems Developed for Turkish Speech Research 32
3.4.2.1 Phonetic Aligner 32
3.4.2.2 Phoneme Recognizer 34
3.4.2.3 Evaluations on the Aligner 34
3.4.2.4 Evaluations on the Phoneme recognizer 37
4 MELP-BASED SPECTRAL MODIFICATION FOR VOICE TRANSFORMATION 40
4.1 Introduction 40
4.2 Speech Model 41
4.2.1 LPC-based Speech Model 41
4.2.1.1 Line Spectrum Frequency (LSF) Pairs 43
4.2.2 The MELP Speech Coder 44
4.3 Speech Data and Time-Alignment 47
4.4 Training 51
4.4.1 MSVQ Representation of MELP and Observations 51
4.4.1.1 Speaker Individualities in the First Stage of MSVQ of MELP 53
4.4.1.2 Personalizing the MSVQ Codebook 55
4.4.1.3 Mutual Information Computations 57
4.4.2 Obtaining the Mapping Histograms 60

4.5 Transformation - the Baseline System 61
4.5.1 Dynamic Programming for LSF Transformation 62
4.5.2 Speech Synthesis 68
4.6 Results 69
4.6.1 Objective Evaluations 70
4.6.1.1 Errors and Performance Indices . . 70
4.6.1.2 Results 71
4.6.2 Subjective Evaluations 73
5 LSF QUANTIZATION FOR VOICE TRANSFORMATION . . 74
5.1 Introduction 74
5.2 Quantizing Line Spectral Frequencies for VT 74
5.2.1 Speaker-specific LSF Quantization 75
5.2.1.1 PCA Application 77
5.2.1.2 k-means Clustering Application . . 79
5.2.2 Training 80
5.3 Transformation 81
5.4 Evaluations 84
5.4.1 Objective Evaluations 84
5.4.2 Subjective Evaluations 90
6 CONCLUSIONS 92
6.1 Summary 92
6.2 Directions for Future Research 93
APPENDICES 95
A SPEECH SYNTHESIS TOOLS DEVELOPED FOR TURKISH SPEECH RESEARCH 96
A.1 Introduction and Background 96
A.1.1 Text Analysis 98
A.1.2 Prosody Generation 99
A.1.3 Unit Selection 100
A.1.4 Waveform Synthesis 101
A.2 Diphone Synthesis Tools Developed for Concatenative Speech Synthesis 102
A.2.1 Diphone Corpus Construction 103

A.2.2 Recording the Diphone Corpus 104
A.2.3 Developing Natural Language Processing Modules for Festival 107
A.2.3.1 Language Specific Modules 108
A.2.3.2 Speaker Specific Modules 109
A.2.4 Waveform Generation 112
A.2.5 Evaluations 112
B EXAMPLE OF A TRIPHONE-BALANCED CORPUS COLLECTION LOG FILE 114
C MUTUAL INFORMATION 115
D PRINCIPLE COMPONENT ANALYSIS 118
E K-MEANS CLUSTERING ALGORITHM 121
REFERENCES 122
VITA 131

LIST OF TABLES

3.1 The most frequent 100 triphones and their normalized occurrence rates in Turkish with METUbet symbols. 27

3.2 Comparison of the triphones in the 2.5 million-word text corpus and the 2462-sentence text corpus. Occurrence rates are normalized to add up to unity. Rankings show the descending order numbers. 29

3.3 File arrangement in the 193-speaker audio corpus. 30

3.4 Speaker grown-up region distributions in the 193-speaker audio corpus. 30

3.5 Word error rate for *Sonic* on several tasks: TI-Digits, DARPA Communicator, Nov'92 Wall Street Journal (WSJ) 5k test set and Swithboard task. Real-time factors are for first-pass decoding on an 800 MHz Intel Pentium III. 32

3.6 Mapping from Sphinx-II to METUbet phone set. 33

3.7 Decision Tree Questions for Turkish. 35

3.8 Comparison of the percentages of automatically placed phoneme boundaries within a fixed distance from the hand-labelled reference. 37

3.9 Phone error rates for 20 Turkish speakers. Results are shown for a baseline system and the same system with iterative unsupervised MLLR adaptation. 38

3.10 System summary percentages per speaker. Cor (correct), Sub (substitution), Del (deletion), Ins (insertion), Err (error) percentages are given. 38

4.1 Examples of codewords for multistage LSF quantization of MELP rounded to two-digit decimals. First two codewords from each MSVQ stage are illustrated. The first column shows the stage number (stg), the second column shows the index number (ind), and the other columns show the frequency vectors (vect). 52

4.2 Mutual Information (MI) between the 1^{st} and the 2^{nd} stages of the source and the target speakers. MP-MI stands for maximum possible MI 58

4.3 Mutual Information (MI) between the reduced $1^{st} + 2^{nd}$ stages of the source and the target speakers. MP-MI stands for maximum possible MI. 60

5.1 Eigenvalues of the covariance matrices obtained from the zero-mean data matrices. Eigenvalues are listed in descending order. 79
5.2 Mutual information in bits, obtained from the mapping histograms of the voiced frames of the two speakers. 90

A.1 New symbols added to the METUbet alphabet for speech synthesis. 103
A.2 Examples of diphones and their carrier words 104
A.3 Diphone index list (.est file) for *Festival* 107
A.4 Turkish vowel definitions in *Festival* text module (V/C: vowel or consonant, length: **S**hort or **L**ong, height: 1/2/3 levels, frontness: 1/2/3 levels, roundness: **R**ound or **F**lat) 109
A.5 Some examples of letter-to-sound rules in *Scheme* 110
A.6 Phonemes and their mean durations for our female TTS speaker. 111
A.7 Turkish Words for Diagnostic Rhyme Test 112

LIST OF FIGURES

2.1 Waveforms and spectrograms from two different male speakers of Turkish, uttering the phrase - *Onun Kaptanı*. Sampling frequency is 8 kHz and time axis is in seconds. 10

3.1 Mapping from IPA to METUbet. 24
3.2 Block diagram of the Turkish phonetic aligner. 36
3.3 Block diagram of the Turkish phoneme recognizer. 37

4.1 Overview block diagram for the voice transformation system. . 40
4.2 LPC-based model of speech production. 42
4.3 MELP decoder block diagram. 45
4.4 4-stage vector quantization of LSF's in MELP with phoneme alignments for the two speakers. The first column shows the frame number, the next 4-columns show the corresponding MSVQ indices. The final column shows the associated METUbet phonemes. 49
4.5 Frames in Figure 4.4 after time-alignment by DTW. 50
4.6 Effect of adding the 1_{st} vectors of 2^{nd}, 3^{rd}, and 4^{th} stages of MSVQ to the 1^{st} LSF vector of the 1^{st} stage. Corresponding filter responses are black: stage1, blue: stage1+stage2, red: stage1+stage2+stage3, dashed: stage1+stage2+stage3+stage4. 52
4.7 MSVQ 1^{st}-stage index histogram for phoneme AA of speaker-1. 54
4.8 MSVQ 1^{st}-stage index histogram for phoneme AA of speaker-2. 54
4.9 Histogram matrix mapping occurrence rates of the 1^{st} stage indices of MSVQ of the two speakers. 55
4.10 Means of the absolute values of the frequencies in 2^{nd}, 3^{rd} and 4^{th} stages of MELP's MSVQ, $msvq_i(k)$ where $i = 2, 3, 4$. 56
4.11 Empirical occurrence probabilities of the MSVQ Stage-1 indices obtained from X^{MSVQ} and Y^{MSVQ} data matrices of the two speakers, $p_1(x)$ and $p_1(y)$. 59
4.12 Comparison of the converted speech output of the baseline system and the target speaker's speech. Illustrated part is the word *kazandı* in Turkish, sampling rate is 8 kHz. 63
4.13 Flow chart of the dynamic programming procedure. 64
4.14 Dynamic programming along the frames of one sentence for LSF transformation. 65

4.15 Illustration of the transformation method in 2-dimensional space. 68
4.16 Comparison of the waveforms and spectrograms of the baseline system and the improved system with dynamic programming. (a) is target speaker's speech, (b) is conversion with baseline system with $L = 128$, (c) is conversion with dynamic programming with $L = 128$ and $D = 0.16$. Displayed is the Turkish word "aydı". 69
4.17 Performance indices of the system for transformation from Speaker-1 to Speaker-2. 72
4.18 Interactive window of the speaker-similarity test. 73

5.1 Plots of the first two components of the normalized (zero-mean) LSFs (upper plots) and the first two components of the transformed data (lower plots) for both speakers. LSF values are given in Hz for a sampling rate of 8 kHz. The upper figures are the mean-subtracted LSFs for voiced frames, while the lower figures are the first two components after the transformation. xT_x stands for the transformed vectors for Speaker-1, and yT_y stands for the transformed vectors for Speaker-2. . . . 76
5.2 Plots of the third and fourth components of the normalized (zero-mean) LSFs (upper plots) and the third and fourth components of the transformed data (lower plots) for both speakers. LSF values are given in Hz for a sampling rate of 8 kHz. The upper figures are the mean-subtracted LSFs for voiced frames, while the lower figures are the first two components after the transformation. xT_x stands for the transformed vectors for Speaker-1, and yT_y stands for the transformed vectors for Speaker-2. 77
5.3 Mapping histogram for quantized voiced frames of the source and the target speakers, $\tilde{X}^V$ and $\tilde{Y}^V$. 81
5.4 Block diagram of the training system. 82
5.5 Block diagram of the LSF transformation system. 85
5.6 LSF performance indices, P_{LSF}, for N values 4, 6, and 8, with respect to the interval, D. The $\otimes$ marked plots are P_{LSF} of the transformation given in Equation 5.8, others are P_{LSF} of the transformation in Equation 5.9. 86
5.7 Detailed plot of the results given in Figure 5.6, with $L = 64$, $N = 4$ and Equation 5.8. 87
5.8 LSF performance indices, P_{LSF}, for transformation from Speaker-1 to Speaker-2, $L = 64$ and the transformation given in Equation 5.8 88
5.9 LSF performance indices, P_{LSF}, for transformation from Speaker-1 to Speaker-2 with $L = 96$ and the transformation given in Equation 5.8 89

5.10 LSF performance indices, P_{LSF}, for transformation from Speaker-1 to Speaker-2 with $L = 128$ and the transformation given in Equation 5.8 89

A.1 Structure of corpus-based TTS. 98
A.2 Quality and task-independence in speech synthesis. 102
A.3 Diphone recording user interface. 105
A.4 Speech waveform (upper panel) and the glottal waveform (lower panel). This is female speech collected at 16kHz, presented speech part is the [eş] boundary in Turkish. Time axis is in seconds. 106
A.5 Diphone recording set-up. 106
A.6 Natural language processing modules in *Festival*. 108

B.1 Beginning part of a triphone-balanced corpus collection log-file. 114

CHAPTER 1

INTRODUCTION

One of the most natural, convenient and useful means of communication for most people is speech, which conveys not only a message, but also information such as emotion, attitude and speaker individuality. As the computer technology advanced within the past two decades, the realization of a man-machine interface to facilitate communication between people and computers has gained importance and, naturally, speech has been focused on as a medium for such communication. Therefore, various speech technologies such as speech synthesis, recognition, voice transformation and coding for different languages, have been worked on widely in recent years.

In this dissertation, we consider new approaches in the design of a newly emerging speech technology called *voice transformation* (VT) for Turkish. The goal of VT is to modify a *source speaker's* speech such that it is perceived as if a *target speaker* had spoken it. Our objectives in this thesis are two-fold. The first objective is to develop a standard speech corpora and segmentation tools for Turkish speech research, which are required to develop a VT system. The second is to consider new approaches for VT.

In this chapter, we first motivate our need for developing speech corpora and speech recognition tools for Turkish. Then, we motivate the use of VT systems by a number of example applications, followed by a brief description of current VT approaches. We then present a summary of our approach to the problem of VT and finally give the organization of the thesis.

1.1 Motivation

Developing any kind of speech technology, such as a VT algorithm, in a new language requires new tools and strategies to be developed specific to that language. Speech and text corpora designed, for the acquisition of acoustic-phonetic knowledge, are required in addition to speech segmentation tools to organize the corpora. Turkish has been lacking a standard corpus of read speech, which is similar to TIMIT [1], the American English corpus of read speech. TIMIT has been designed for development and evaluation of automatic speech recognition systems and it contains a total of 6300 microphone quality recorded sentences, 10 sentences spoken by each of 630 speakers from 8 major dialect regions of the United States. There have been recent attempts to develop audio and text corpora for Turkish. Interactive Systems Laboratories at Carnegie Mellon University has collected a multilingual audio corpus, *GlobalPhone* [2], which covers 9 of the 12 most widespread languages of the world. It includes 22.2 hours of read Turkish speech from 100 speakers of native speakers. Text is chosen from the political and economical articles selected from national newspapers. TÜBİTAK, The Scientific and Technical Research Council of Turkey, has collected an audio corpus of 65 speakers, each reading 373 words and 15 sentences [3]. This corpus was designed to be triphone-balanced; however, it is telephony speech. Another telephony speech Turkish corpus has been collected through the *OrienTel* Project [4], which aims to enable the project's participants to design and develop multilingual interactive communication services for the Mediterranean and the Middle East. This corpus includes segmented telephony speech from 1700 speakers, each uttering about 6 minutes of speech [5]. Turkish has been lacking a triphone-balanced high quality microphone speech.

Our motivation to develop tools for automatic segmentation of Turkish speech raised from our need to segment the database collected for our VT research. There are numerous applications in which large quantities of phonetically segmented speech is necessary. For example, model parameters for large

vocabulary speech recognizers are estimated from thousands of phonetically segmented speech. Speech synthesizers based on waveform concatenation can also require thousands of segmented speech segments. Since manual segmentation of such databases is time consuming, automatic speech segmentation algorithms have been proposed in order to provide efficient phonetic labelling.

The segmentation tools we have developed have also been used to develop a Turkish *text-to-speech* (TTS) engine which is given in Appendix A. This work is not in the scope of this thesis; however, it has been developed as a side-product using the segmentation tools developed in this thesis.

Our second goal is to consider new approaches in the design of a VT system. The sound of a person's voice plays an important part in daily communication. This is called *speaker identity* and it allows us to recognize people from their voices alone. It also makes it possible to differentiate between speakers, for example, on a radio program. Speaker identity can be controlled by means of a VT system. There are various applications of a VT system: An example is the integration of a VT system with a TTS synthesizer. Today's state-of-the-art TTS systems are based on concatenative synthesis method in which segments from a natural database are combined together to generate a new utterance. The creation of such a database requires a significant amount of recording, segmenting and labelling effort. For example, a speaker may be required to talk in a restricted way for several hours to collect even a relatively small speech inventory of 2500 diphones [6]. For an inexperienced speaker, this may even take more than 10 hours [7]. In addition, trained labellers can spend 10-100 hours for every hour of recorded speech, depending on the complexity of transcriptions [6]. Using VT technology, new synthesis voices can be created by transforming the voice of the speaker of the inventory to new speakers' voices. Speaker model of the source and the target speakers and hence the transformation model can be extracted using a much smaller inventory than the original TTS inventory. Using different speaker models, the synthesis system can generate speech signals with different speaker identities from a single

speaker database which belongs to the source speaker [8]. This approach can also be used to develop the voice of a speaking-impaired person who can provide limited amount of speech data, or the voice of an unavailable speaker, whose small amount of previous speech data is owned. For concatenative TTS systems, VT technology can also be used to preserve the voice quality of the speaker during the long recording session. The perceptual quality of a speaker often changes during long recording hours, and VT can be applied on the resulting corpus such that the final parts of the database have the same speech quality with the beginning part of the database [9].

In today's state-of-the-art VT systems there are two basic modes: Training and transformation. In the training mode, the system uses speech samples from the source and the target and it tries to estimate a transformation function from the source speaker's speech to that of the target speaker. Once the training part has been completed, the system is ready to transform the source speaker's speech to make it sound like the target speaker. One solution to VT problem can be thought of as a system obtained by cascading a speech recognizer and a text-to-speech synthesizer. Although this seems to be an interesting solution to the VT problem, because of the lack of good quality speech recognition and synthesis systems such approaches seem to be feasible in the future.

1.2 Contributions

The major contributions of this thesis can be summarized as follows:

- A phonetic alphabet for Turkish which considers not only phonemes but also all allophones listed in the phonetic dictionary given in [10] has been developed. A triphone-balanced (in terms of the phonetic alphabet developed) 2462-sentence text has been prepared. Then a triphone-balanced audio corpus of 193 speakers, uttering 40 balanced sentences each, for Turkish has been collected.

- Automatic segmentation tools (a phonetic aligner and a phoneme recognizer) for Turkish have been developed and the phonetic aligner has been used to segment our audio corpus.

- A new VT approach based on *Mixed Excitation Linear Prediction*, MELP [11], speech coding algorithm framework has been developed and evaluated.

- A dynamic programming approach which considers the dependence between transformed frames has been developed and used to improve the synthetic speech quality. The allocation of the spectral peaks independently in each frame has been reported as a problem remained to be solved in VT systems.

- A new method for quantizing line spectral frequencies has been used, which includes dimension reduction of the feature space. Quantization of the reduced dimensions has improved the synthetic speech quality.

1.3 Outline of the Thesis

Chapter 2 gives a detailed survey on works related to the objective of this thesis together with that on the application field of our results, that is: VT models and evaluations. We present the phonetic alphabet, audio corpus, the phonetic aligner, and phoneme recognizer designed for Turkish in Chapter 3. We also present the diphone corpus collected for the Turkish TTS system developed using the Festival speech synthesis system in Appendix A. The baseline VT system using MELP speech coding framework and the improvements on this system have been explained in Chapter 4. The dynamic programming approach is also explained in Chapter 4. Chapter 5 introduces another method for quantizing line spectral frequencies with dimension reduction applied on our previous framework for VT. Objective and subjective evaluation results are given and compared for both systems in Chapters 4 and 5. Chapter 6 concludes the thesis, providing also notes on future works.

CHAPTER 2

AN OVERVIEW OF VOICE TRANSFORMATION SYSTEMS

In this chapter, a literature survey on published works in the area of voice transformation (VT) systems is presented. The first section introduces the basic properties of the speech signal which should be considered while developing a VT system. Next section introduces components of a VT system and summarizes the current VT techniques. Finally, the last section presents evaluation methods for a VT system.

2.1 Basic Properties of the Speech Signal

Basics of the speech production will be presented in the first part of this section. The relationship between the speech signal features and speaker individuality is the subject of the second part.

2.1.1 Speech Production

Human speech is produced by air-pressure emanating from the mouth and the nostrils of a speaker. The compression of the lungs induces a stream of air flowing through the vocal tract, which begins at the vocal cords, or *glottis*, and ends at the lips. This air flow is the source of four types of sounds [6] :

- **Aspiration noise**, which is the sound of air rushing through the entire vocal tract, similar to breathing through the mouth.

- **Frication noise**, which is the sound of the turbulent air at a point of narrow constriction, for example during the initial sound in "fair".

- **Plosion**, which is the sound of air-burst, for example during the initial consonant in "ton".

- **Voicing**, which is a quasi-periodic vibration of the glottis, for example during the vowel in "key". The frequency of the vibration is called the *fundamental frequency* or F_0, and it is perceived as *pitch*.

These four types of sounds may occur in any combination. These sounds are further modified by the vocal tract shape, which is determined by the following organs:

- **Vocal cords (glottis)**: When the vocal cords are close together and oscillate against one another during a speech sound, the sound is said to be *voiced*. When the folds are too slack or tense to vibrate, the sound is said to be *unvoiced*.

- **Velum (soft Plate)**: It operates as a valve, which opens to allow passage of air through the nasal cavity. Sounds produced when it is open are called *nasals*, which are m and n.

- **Hard palate**: This is the long and relatively hard surface at the roof inside mouth, and it enables consonant articulation when the tongue is placed against it.

- **Tongue**: This is the flexible articulator. It is shaped away from the palate for vowels, and it is placed close or on the palate or teeth for consonant articulation.

- **Teeth**: This is another place of articulation used to brace tongue for certain consonants.

- **Lips**: Lips can be rounded or spread to affect vowel quality. They are completely closed to stop the oral air flow in certain consonants (p, b,

and m).

Different vocal tract shapes have different resonant frequencies, called *formants*, which are instrumental in developing the nature of the different speech sounds, called *phonemes*.

Many different models have been postulated for quantitatively describing certain factors involved in the speech process. One of the most successful models of acoustical speech behavior is the linear *source-filter model*, which satisfies the basic criterion of modelling (being able to find mathematical relations which can be used to represent a limited physical situation with minimum complexity and maximum accuracy) [12]. In this model, which will be used throughout this thesis, a *source* or *excitation waveform* is input into a *time-varying filter*. This view of speech production is very powerful, because it can explain the majority of the speech phenomena [6]. The excitation waveform accounts for the physiological sound sources listed above. For example, aspiration and frication noise can be modelled as random processes, plosion as a step-function, and voicing as a pulse train. A number of glottal pulse models have been proposed to describe the details of the pulse shape during voicing [13], [14]. In most systems today, the excitation waveform is usually classified into a *voiced* and an *unvoiced* signal, which can sometimes be modelled in their simplest form as either a random signal or an impulse train with varying *pitch*, respectively. The time-varying filter represents the contribution of the vocal tract shape by selectively attenuating certain frequencies of the excitation spectrum resulting in a speech waveform with a particular formant structure for various speech sounds.

2.1.2 Voice Individuality

The speech signal contains many types of information. The signal carries information about the message (what is said), about the speaker (who said it) and the environment (where it was said). The task of VT is to change the speaker information, i.e., the *voice individuality*, while preserving other

information types. The factors that are relevant to voice individuality can be categorized in terms of socio/psycho-logical versus physiological dimensions [15]. *Speaking style*, which depends on factors like age, social status, dialect and the community to which the speaker belongs, is socially conditioned. The *sound of voice*, on the other hand, comes mainly from the physical properties of the speech organs. Physical properties of the speaker affect the glottal source frequency range (i.e., pitch range) and its frequency spectrum, and also the power spectrum of the vocal tract for various positions. The characteristics of a speaker are commonly divided into the following types of cues [6]:

- **Segmental cues:** Acoustic descriptors of segmental cues include formant locations and bandwidths, spectral tilt, F_0, and energy. Segmental cues depend on the physical properties of the speech organs.

- **Suprasegmental cues:** These are related to the style of speaking, for example the duration of phonemes and the evolution of F_0 (intonation) and energy (stress) over an utterance. These cues are influenced by social and physiological conditions. Suprasegmental cues can be easily changed at will. For example, it is easy for a speaker to slow his or her speech, lower the voice or speak more softly. Therefore, impersonators usually mimic suprasegmental characteristics [15]. However, some segmental cues can be mimicked by impersonators who are specially skilled in changing some part of their vocal tract physically or in modifying the behavior of their glottal pulse [6]. Even formant frequencies and bandwidths can be affected in this manner.

- **Linguistic cues:** These include particular choice of words, dialects and accents. Our work in this thesis will focus on segmental and suprasegmental cues only. Linguistic cues are beyond the scope of this thesis.

Some of the segmental and suprasegmental cue differences between different speakers can be illustrated by comparing the waveforms and spectrograms of the same phrase "Onun kaptanı" in Turkish uttered by two male

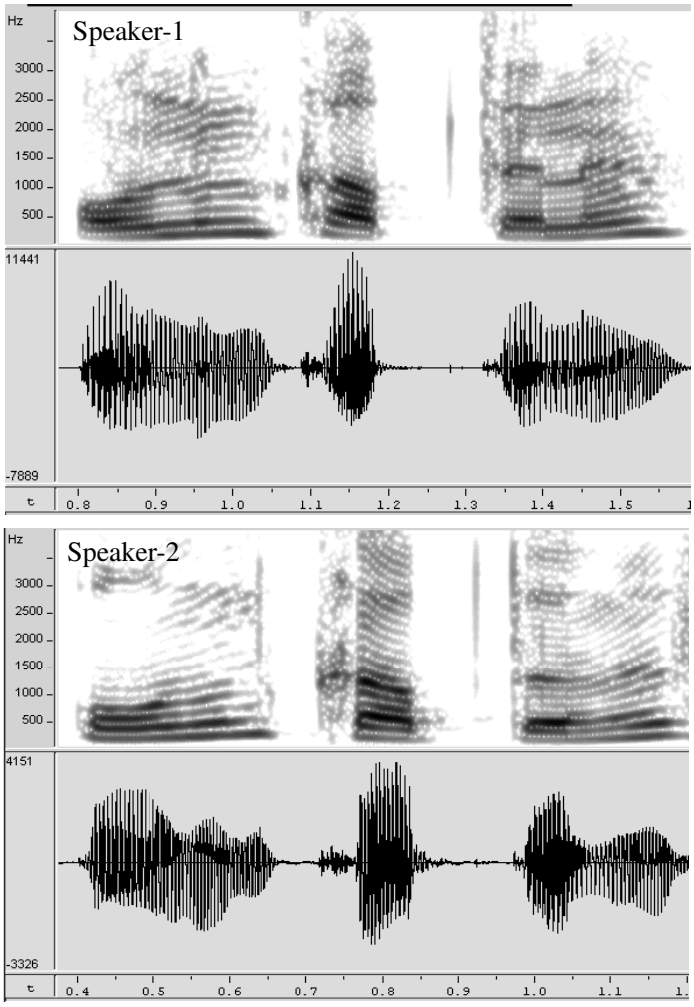


Figure 2.1: Waveforms and spectrograms from two different male speakers of Turkish, uttering the phrase - *Onun Kaptanı*. Sampling frequency is 8 kHz and time axis is in seconds.

speakers in Figure 2.1. Durations of phonemes and amplitude contours are different. Speaker-2 exhibits a more voiced structure in the spectrogram at high frequency bands, while Speaker-1 has more discontinuities in his formant structures as seen in the figure.

2.1.2.1 Speaker recognition by humans

Before investigating the effect of segmental or suprasegmental cues on voice individuality, it is important to understand how accurate human listeners can identify voices.

Humans are capable of identifying voices under various conditions and con-

texts with a fairly high degree of accuracy if the voices are familiar to the listener; however, human speaker-recognition is actually far from being perfect. The ability of one listener to recognize voices familiar to him or her from a set included 29 familiar and 24 unfamiliar voices, has been measured in an experiment, which had shown that 31% of the 29 familiar voices were correctly identified from a single word, 66% from a single sentence and only 83% from 30 seconds of speech [6].

2.1.2.2 Relative effects of segmental and suprasegmental cues on speaker individuality and human perception

This section summarizes the work in the literature which investigates the effects of various segmental and suprasegmental cues on speaker discrimination by humans.

In [16], a subjective listening experiment made on synthetic speech compared the effects of the LPC spectral envelope and LPC residual on speaker identity. The authors concluded that LPC spectral envelope has a greater effect on speaker identification than the LPC residual.

In another experiment, the relation between the perceptual discrimination of speaker identity and the difference in elementary acoustical parameters has been explored [17]. The authors have studied hybrid voices produced by interchanging the approximated glottal source wave and vocal tract spectrum among speakers. The results suggest that the vocal tract has a relatively greater contribution than the glottal source characteristics, to the ability of humans to discriminate speakers. Another experiment made on the *TIMIT* continuous speech corpus [1] has shown that median pitch and vocal tract length for males; median pitch, glottal tilt and average duration of unvoiced speech segments for females are the major dimensions of subjective speaker similarity judgements [18]. These results confirm that average pitch is the most identifying cue in discriminating between speakers followed by segmental cues.

In [19], it is reported that voice personality is more sensitive to the formant frequency shift than to the pitch frequency and bandwidth changes. Voice personality is lost for the uniform shift of approximately five percent according to subjective perception tests [19]. It has also been reported that for formant shifts, personality is more sensitive to the lower three formants than to the higher ones, while for bandwidth manipulation it is more sensitive to the higher formants. Compared to the formant frequency manipulation, perception of voice-personality is reported to be far less sensitive to the pitch manipulation. Personality is retained for the increase or decrease of average pitch frequency less than 1.45 to 0.6 times of the original speech. In [20], Kuwaba also reported that it is necessary to change the pitch frequency by as much as 50% or more toward either low or high frequency regions to lose voice individuality. Another study in [21] reports that there are also speaker individualities in pitch frequency contours and some specific parameters related to the dynamics of pitch contours. Another interesting result they have reported is that time averaged pitch contours play a much smaller role in speaker individuality than dynamics of the pitch contours.

2.2 An Overview of Voice Transformation Systems

This section introduces parts of a VT system briefly and summarizes work on VT research in the literature. In a VT system, there are two basic modes: Training and transformation. In the training mode, the system uses speech samples from the source and the target and it tries to estimate a transformation function from the source speaker's speech to that of the target speaker. Once the training part has been completed, the system is ready to transform the source speaker's speech to make it sound like the target speaker.

A VT system should have at least three basic components: A speech corpus, a speech model and a transformation function. The speech corpus supplies

speech data from both the source and the target speakers. This corpus is used both for training and testing. Speech model is a mathematical model of the speech signal. The model parameters are selected according to the desired transformation. The model parameters, which are called *features*, are obtained during training. In the transformation mode, these features are transformed using the transformation function and the speech is re-synthesized using the transformed features.

2.2.1 Speech Corpus

The speech corpus provides necessary speech data for training the transformation function and for testing the performance of the VT system using objective and subjective evaluation measures. The speech corpora may contain as little as five vowels [22, 23], a set of words [24, 25], short sentences [26], or one hour of read speech [27]. Alternatively, speech databases created for text-to-speech synthesizers have been used [8, 28]. The optimal size of the speech corpus depends on the application. In [6], a special corpus, which consisted of 50 phonetically-balanced sentences of English, has been designed. Voice has also been recorded carefully, such that the subjects were asked to mimic pre-recorded utterances, which were slow and with flat intonation. This helped the speech of different speakers to be aligned naturally as much as possible. Reducing the alignment errors increases the transformation performance.

2.2.2 Speech Modelling

In VT systems, the ideal speech model should produce a wide variety of speech that is intelligible, as well as natural and accurate with respect to speaker recognizability. These can be achieved with numerous number of speech parameters, however, as the number of parameters increase, it is harder to find a robust transformation function. Therefore, it is important that a well-matched speech model and transformation function are used.

In Section 2.1.2, it has been shown that voice individuality is found in all

acoustic cues with varying degrees. However, researchers have found evidence that segmental features and the average behavior of suprasegmental features (rate of speech and F_0) are sufficient for a high degree of speaker discrimination by humans [6]. Studies on speaker identification have shown that the spectral envelope alone contain a great deal of information to identify speakers with the help of a computer [29]. Therefore, VT systems today usually focus on transforming a representation of the short term spectral envelope, while adjusting the source speaker's F_0, energy and rate of speech to match those of the target speaker on the average. The speech processing in VT systems is usually performed on small sections of speech, which are called *frames*, at a time.

The source-filter model which approximates the vocal tract as a slowly varying filter by fitting a spectral envelope to the magnitude spectrum of a short segment of speech for representing the speech spectrum is commonly used in VT systems. This approach has been mentioned in Section 2.1. The model parameters of the source-filter model are often obtained by linear prediction, which will be defined in Chapter 4. The filter coefficients are called *linear prediction coefficients* (LPC). LPCs are converted to a number of alternative representations with more desirable properties such as the ability to be interpolated robustly. For example, cepstral coefficients [30], line spectral frequencies (LSFs) [26, 8, 31], formant frequencies and bandwidths [32], and log area ratios [24] have been used in VT systems.

In LPC analysis, inverse filtering a speech segment with its corresponding LPC filter gives the approximate glottal excitation waveform. It is possible to keep the residual signal unchanged spectrally in a VT algorithm as in [25, 8]. The result is a more natural sounding speech signal [6]. However, since the residual signal also carries a certain degree of speaker information, modifying the residual is also necessary to obtain more similar speech with the target. Several methods have been proposed to modify the LPC residual in addition to the LPC spectrum. In [26] and [27], a codebook-based transformation of

LPC residuals using a weighted combination of excitation filters has been formulated. The excitation filters have been derived from the average source and target residual spectra within one class. Both the source speech spectrum envelope and the LPC residual spectrum are transformed based on a single classification. This method, however, has been reported to cause speech quality degradations, so the authors have applied a bandwidth modification method to increase speech quality [27]. In [6], a new approach for LPC residual modification has been proposed. Instead of transforming the source residual directly, a relation between the LPC spectrum and the LPC residual of the target speaker is obtained. Then, that relation is used to predict the target LPC residual from the transformed LPC spectral envelopes during voiced speech. It has been reported that, modifying the LPC residual together with the LPC spectral modification produces perceptually closer speech to the target; however, voice quality is quite below the natural stimuli [6].

2.2.3 Transformation Methods

A transformation function captures the relationship between the source speaker and the target speaker during training, and it is applied on the source speaker's speech during the transformation mode of a VT system. The transformation function can be a continuous function to be applied on the speech features [6, 9, 33], or it can be a discrete mapping from the source speaker's feature space to the that of the target [24, 27, 34].

The transformation function is obtained from the training speech corpus of both the target and the source speakers. Naturally, the durations of the linguistic units differ between speakers, even when producing the same utterances. Therefore, stream of features stemming from both speakers should be aligned in time before training. The time-alignment has been implemented by a *dynamic time warping* (DTW) algorithm [35] in most of the previous approaches [25, 24, 32, 28, 6]. Some approaches use unsupervised Hidden Markov Model (HMM) [26], or forced-alignment speech recognition [8]. Some approaches use

forced-alignment and DTW together [31, 34], to warp segments of speech inside phoneme boundaries, or some use phonetic classifiers as in [27].

One of the earliest approaches to the spectral conversion problem is the *mapping codebook* method [24], which was originally introduced for speaker adaptation [36]. In this approach, a clustering procedure (vector quantization) is applied to the spectral parameters of both the source and the target speakers. The resulting vector quantization codebooks are used to obtain a mapping codebook whose entries represent the transformed spectral vectors corresponding to the centroids of the source speaker codebook. The main shortcoming of this method is the fact that the parameter space of the converted envelope is limited to a discrete set of envelopes. Several variations of this basic scheme have been investigated in order to overcome this limitation. Using piecewise linear conversion rules has been suggested to modify the formant frequencies and the spectral intensity [32]. The conversion rules are generated statistically using vector quantized parameters of the two speakers. A similar approach is given in [37], where an orthogonal vector space conversion technique is proposed to transform LPC cepstral coefficients. This technique consists of principle component decomposition by applying the Karhunen-Loeve transformation and minimum mean-square error coordinate transformation. The spectral interpolation approach described in [38] uses interpolation among the spectrum of several speakers to determine the converted spectrum. For adaptation to a target speaker, the optimal interpolation coefficients are determined so as to minimize the distance between the target speaker's spectrum and the spectrum generated by interpolation. The pre-stored data of the multiple speakers provide the information on speech spectral consistency, characteristics and dynamics of the spectral structure of speech. Some researchers suggest that a possible way to improve the quality of the converted speech is to modify only some specific aspects of the spectral envelope, such as formant frequency locations [22, 32]. In [23], using artificial neural networks has been suggested for voice conversion. Formants are used to represent the vocal tract system fea-

tures and a formant vocoder is used for synthesis and formant transformation is captured by a neural network. In [27], mapping codebooks have been used to map the LSFs of source and target speakers. In a recent publication [39], a sub-band based version of the same method has been proposed, which ensures higher quality conversion at sampling rates higher than 16 kHz. The same conversion scheme as in [27] is applied, but only the lower band of the speech signal is transformed while keeping the higher frequency sub-bands untouched. This method is reported to be providing higher quality speech output than the full-band conversion for high sampling rates.

Several researchers have proposed to use an individual transformation function for each class of speech sound. Each transformation function represents a relationship between source and target features of one class. Two types of local transformation approaches have been used: linear regression and dynamic frequency warping (DFW) [25]. For each class, an algorithm computed the optimal transformations for both linear regression and DFW during the training process. Similarly, in [32], a set of linear transformation rules, which depend on the input class, has been used. The techniques in [25] and [32] use discrete transformation functions which are capable of producing an infinite number of target features. This is more advantageous than the codebook based methods, which result in a discrete set of converted envelopes. However, discontinuities can still occur in the output due to the discrete nature of selecting a single local transformation function.

Continuous transformation functions have also been offered [23, 6, 28]. In [23], formant frequencies have been transformed with the help of an *artificial neural network*. They have found that the network generalized properly to unseen data. Another popular continuous transformation technique is using *Gaussian mixture models* (GMM) to describe and map the source and target feature distributions. In [9], a classification of the source feature space has been performed by constructing a GMM that modelled the source feature space. Then parameters of a mixture of locally linear transformation functions are

estimated by solving normal equations for a least-squares problem based on the correspondence between source and target features. Discrete *mel-frequency cepstrum coefficients* have been used to model the source and target features.

In [8, 6] probabilistic and locally linear transformation functions have been proposed using a GMM which is estimated on the joint density of source and target features. Modelling the joint density rather than the source density alone can lead to a better allocation of mixture components and avoids certain numerical problems when inverting large and poorly conditioned matrices [6].

2.3 Evaluation of VT Systems

Evaluation of VT systems are realized by both objective scoring tests and subjective listening experiments. Objective evaluation is useful in comparing algorithmic alternatives within the same VT system. On the other hand, a perceptual system evaluation is inevitable, because the output of a VT system is intended to be heard by human-beings.

2.3.1 Objective Evaluation

Objective measures are usually based on computing the distortion between two speech signals. In VT systems, the average *spectral distortion* (SD) between the target and the transformed speech is compared to the SD between the target and source. $SD(target, converted)$ should be smaller than $SD(target, source)$ for a successful VT system. This comparison has been used by many researchers [24, 27, 30, 8, 6]. Another method is using a simple speaker verification system to test the performance of the VT system objectively [40]. Comparing the log-likelihood of the source and the transformed speech using the target's probability density function gives the amount of reduction in the distance from the source speaker to the target by the transformation [27, 31].

2.3.2 Subjective Evaluation

The perceptual evaluation of a VT system could measure several dimensions: Intelligibility (Do listeners understand what is said?), naturalness (Does the transformed speech sound natural or synthetic?), and speaker recognizability (Does it sound like the target speaker speaking?).

The most popular perceptual test for VT systems in the literature is the **ABX** test, which measures speaker recognizability. In this test, subjects listen to three stimuli, **A**, **B**, and **X**. They are asked to decide whether **X** is closer to **A** or **B**. **X** is typically the transformed speech, and **A** and **B** are the source and the target speakers. Utterances can be short phonetic units [30], 2-3 word phrases [27], or whole sentences [6, 34]. One useful property of **ABX** test is the elimination of any response bias, i.e. people are equally likely to choose **A** or **B** if indeed **A**=**B** [6]. The disadvantage is that it involves 3 stimuli, which may cause memory effects, such as forgetting the first utterance while listening to the third. Also, a score of 100% does not determine whether the transformed speech is indistinguishable from the target speaker's speech, it only indicates that the transformed speech is closer to the target speech. In [8], during the evaluation of a VT system in conjunction with a TTS system, scores of 97.5% for male-female conversion and 52% for male-male conversions have been obtained. On the other hand, a result of 100% for male-female and 78% for male-male transformations with 3 subjects listening to ten 2-3 word phrases has been reported in [27]. 80% and 77% have been reported for two different methods, using 20 subjects listening to 10 sentences in [34].

An alternative to the ABX test is the *similarity test*. Subjects listen to pairs of utterances instead of triads in this type of test, and they rate the similarity of the utterances in each pair. This test has also been used widely in the literature [24, 6]. This test can also be used as a preference test to check the naturalness of the transformed speech with respect to various system parameters [30].

Naturalness of the transformed speech can be measured by carrying out

a *mean opinion score* (MOS) test [35, page 336]. In a standard MOS test for characterizing the quality of a speech signal, listeners rate the transformed speech with ratings from 1 to 5 ("bad", "poor", "fair", "good", "excellent"). In [8], a MOS test has been used to test the quality of the transformed speech. In [27], the intelligibility of the transformed nonsense speech has been measured. It has been found that the phone accuracy of the transformed speech was similar to that of the source speaker's speech.

In [6], some problems of the existing subjective listening tests (i.e. tests being in small scale, lack of standardization, shortcomings of the ABX tests, lack of a standard VT corpus for testing, and etc.) have been addressed and a new evaluation strategy has been proposed for measuring the transformation performance with a focus on speaker recognizability. The natural ability of humans to distinguish and recognize the speakers of the speech corpus were measured, and these measurements served as a baseline against which the system's transformation performance was compared.

CHAPTER 3

SPEECH CORPORA AND RECOGNITION TOOLS DEVELOPED FOR TURKISH SPEECH RESEARCH

In this chapter, we present our work on speech corpora, analysis and recognition tools developed for Turkish, which we have also used for our later work on speech synthesis and VT.

3.1 Introduction

To develop a speech synthesizer and a VT system for Turkish, we needed to collect and segment a large audio corpus of at least two speakers. Since segmenting a speech database manually is both time consuming and prone to human errors, we had to focus on developing an automatic speech segmenter for Turkish. Moreover, the audio corpus to be used for VT training and evaluations had to have maximal phonetic coverage for our VT research to be reliable. This chapter focuses on our preliminary work on Turkish speech, which we believe to be useful also for researchers working in the area of Turkish speech[1].

Developing speech analysis/synthesis tools specific to a language requires

[1] The part of the research in this chapter and the work on speech synthesis given in Appendix A were achieved during a 17-month-period spent as a visiting-researcher at the Center for Spoken Language Research (CSLR) of University of Colorado at Boulder in USA. The visit was supported by TÜBİTAK, the Scientific and Technical Research Council of Turkey, for one year through a combined doctoral scholarship program. The rest of the visit was supported by CSLR.

some questions to be answered for that language. For example;

- What phonemes, diphones and triphones exist in the language?
- What sentences should be recorded such that the maximal phonetic coverage is achieved in the audio corpus?

No direct research to answer such questions has ever been made for Turkish. A phonetic alphabet which represents all phonemes and maybe some allophones is required before answering those questions. After developing a convenient phonetic alphabet, we have obtained those answers using a large text corpus, which had been collected from online Turkish newspapers. Then a 2462 triphone-balanced Turkish sentence set has been determined for developing an audio corpus. Those sentences have been used to collect an audio corpus at the Middle East Technical University (METU).

This corpus has been used to port CSLR's speech recognition system, *Sonic* [41], which had been developed for American English to Turkish. This system is able to align speech with its text and can also recognize phonemes. As future work, this system can be improved to work as a continuous speech recognizer. A detailed and a robust language modelling for Turkish, which is an agglutinative language, however, is remained to be added to the system.

3.2 Phonetic Alphabet for Turkish

Modern standard Turkish is a phoneme-based language like Finnish or Japanese, which means that phonemes are represented by letters in the written language [42]. It is also true to say that there is nearly one-to-one mapping between the written text and its pronunciation. However, some vowels and consonants have variants depending on the place they are produced in the vocal tract [10]. For example the letter *a* in the word *laf* is pronounced predorsal, while in the word *almak* *a*'s are pronounced postdorsal. So, 29 letters in the Turkish alphabet are represented by 43 phonetic symbols in [10]. In order to be more accurate

for both recognition and synthesis tools to be developed in our research, we have decided to consider those variances, which are most often the allophones of the Turkish phonemes. The representations in [10] are in International Phonetic Alphabet (IPA) symbols [43]. However, the orthography of IPA makes extensive use of letters that are not available in our computer programming applications; therefore, we have decided to use a mapping of the IPA symbols to a more computer-friendly set of ASCII symbols. Turkish form of a computer readable phonetic alphabet called *SAMPA* [44] has already been developed on the initiative of the OrienTel Project [4], but Turkish-SAMPA [45] does not cover all the allophones in [10]. So we have mapped IPA symbols for Turkish phonemes to a new set of phonetic symbols, which we have called *METUbet* [46]. The choice of symbol formatting in METUbet is similar to that used within Sphinx-II, which has been designed as a mapping of IPA symbols for American English [47]. METUbet symbols, corresponding IPA symbols and example Turkish words are given in Figure 3.1. This is a complete reference to the phonetic symbols used throughout this thesis.

3.3 Design of a Phonetically Rich Audio Corpus

Design of the text for an audio corpus requires careful selection of utterances such that the corpus represents the general phonetic behavior of the language. To achieve the phonetic-balance of Turkish, we have considered triphones as basic units, because it has been reported that for *Hidden Markov Model* (HMM) based continuous-speech-recognition and keyword-spotting systems, triphone-modelling is more powerful than phone, word or syllable modelling in terms of consistency and inclusion of co-articulatory effects [48]. The only problem with using triphones as speech units is that there exists a large number of triphones in any language. For example, Turkish has been reported to have approximately 27 thousand triphones [3]. This problem can be overcome

IPA	METUbet	Example	IPA	METUbet	Example
ɑ	AA	**a**nı	l	L	**l**eylek
a	A	**l**af	ɬ	LL	ku**l**
e	E	**e**lma	m	M	da**m**
ɛ	EE	d**e**re	n	NN	a**n**u
i	IYG	**i**ğde	ŋ	N	sü**n**gü
I	IY	s**i**mit	p	P	**p**p
İ	I	**ı**sı	r	R	**r**af
ɔ	O	**s**oru	ɾ	RR	u**r**mak
o	OG	**o**ğlak	ɣ	RH	bi**r**
U	U	k**u**lak	s	S	**s**es
u	UG	**u**ğur	ʃ	SH	a**ş**ı
Œ	OE	**ö**rtü	t	T	ü**t**ü
Ø	OEG	**ö**ğren	v	V	**v**ar
Y	UE	**ü**mit	ʋ	V	ta**v**uk
y	UE	d**ü**ğme	j	Y	**y**at
b	B	**b**al	z	Z	a**z**ık
d	D	**d**ede	ʒ	ZH	yo**z**
g	GG	**k**arga	ʤ	C	**c**am
ɟ	G	**g**enç	ʧ	CH	se**ç**im
h	H	**h**asta	f	F	**f**asıl
k	KK	a**k**ıl	ʒ	J	mü**j**de
c	K	**k**edi			

Figure 3.1: Mapping from IPA to METUbet.

using two different methods: The first method determines all the triphones in a large audio corpus and the triphones with acoustic similarities are quantized into groups. Then a subset of the utterances in the corpus which contains these selected groups are used as the balanced audio corpus. The second method determines the most frequently used triphones in a large text corpus which is supposed to be large enough to model the language well. Then, a set of balanced sentences are designed depending on those triphone frequencies. The audio corpus is collected later using the balanced sentence set. The first method has the advantage of using the already-recorded audio corpus at hand;

however, the audio corpus may not be representing the language properly. Moreover, the triphone models should be robust for determining the correct grouping of the triphones. The second method is more time-consuming, but it has more control on the text to be prepared for audio recording. We have considered the second method to exactly determine the sentence set to be recorded.

3.3.1 Construction of a Phonetically Rich Sentence Set

The number and frequencies of the triphones in Turkish have been obtained from a large text corpus collected at METU. This corpus had been collected from national newspapers and some Turkish document pages on the web [7]. We have normalized the corpus (numbers and abbreviations are expanded, repetitions of headings and foreign words are cleared up) and ended up with a text corpus of 2.5 million words (2,529,850 words exactly). Next, it had to be converted into METUbet symbols to determine the triphone frequencies in Turkish. Since there are 29 graphemes and 43 METUbet symbols, the mapping from grapheme to phonemes is not one-to-one, but they are usually context or part-of-word dependent. For example, the mapping from grapheme k to the allophones K or KK, and that from a to A or AA are context dependent as illustrated in the example given below:

Word	**METUbet**
laf	L A F
akıl	AA KK I LL
çekil	CH E K IY L

As seen in the above examples, a coming after l is predorsal and it is represented by A in METUbet, while postdorsal a is represented by AA. When k is between back vowels (in the second row of the table), it is back-pronounced and is represented by KK. When k is between front vowels (given in the third row of the table), it is frontal and represented by K in METUbet.

A set of mapping rules (letter-to-sound rules) [7], which had been previously developed at METU from the Turkish pronunciation dictionary in [10], has been used to convert the 2.5 million-word text corpus into METUbet phonetic representation. This has provided some insight into the frequency of the occurrences of triphones within the Turkish language. The beginning and the end of the sentences have been marked with *SIL*, meaning "silence". *SIL* has also been counted as a phoneme in triphone occurrence counting (i.e. SIL-phoneme-phoneme, phoneme-SIL-phoneme, and phoneme-phoneme-SIL occurrences have also been counted). The total number of triphones in terms of METUbet symbols has been found to be 16,977,808 and the number of unique triphones is 29,266, which is very close to the number suggested in [3]. The most frequent 100 triphones of Turkish (assuming that 2.5 million-word text corpus represents general triphone occurrence tendencies in Turkish), and their normalized occurrences rates are given in Table 3.1. Normalization is achieved by dividing the number of occurrences by the total number of triphones, 16,977,808.

Construction of a phonetically balanced set of sentences for Turkish has been achieved in the following steps:

- First 2000 sentences of the TIMIT corpus have been translated into Turkish.
- These sentences are converted into METUbet symbols using letter-to-sound rules developed at METU.
- Triphone occurrence rates in those sentences are obtained and compared with those found from the 2.5 million-word text corpus.
- An extra 462-sentence set has been added to 2000 sentences in order to ensure coverage of the most frequent 5000 triphones in the 2.5 million-word text corpus.

The resulting 2462-sentence set includes most frequently used 11,033 triphones in Turkish (before augmenting the 2000-sentence set, that number was

Table 3.1: The most frequent 100 triphones and their normalized occurrence rates in Turkish with METUbet symbols.

METUbet Triphone	Occurrence Rate	METUbet TRIPHONE	Occurrence Rate	METUbet TRIPHONE	Occurrence Rate
EE RR IY	0.0047	S IY NN	0.0018	IY CH IY	0.0013
LL AA RR	0.0046	B IY L	0.0018	AA Y I	0.0013
L EE RR	0.0045	I G I	0.0018	L EE NN	0.0013
B IY RH	0.0044	AA NN LL	0.0018	EE K IY	0.0013
NN D AA	0.0038	EE NN IY	0.0017	KK LL AA	0.0013
IY NN IY	0.0034	IY G IY	0.0017	T AA NN	0.0013
RR IY NN	0.0029	AA NN AA	0.0017	KK AA NN	0.0013
AA RR I	0.0029	IY NN EE	0.0017	IY S IY	0.0013
IY L EE	0.0029	L AA RH	0.0016	M AA S	0.0012
I NN I	0.0028	IY Y O	0.0016	M AA N	0.0012
AA S I	0.0028	AA RR I	0.0016	RR AA KK	0.0012
IY NN D	0.0027	LL M AA	0.0016	AA B IY	0.0012
I NN D	0.0027	B AA SH	0.0016	Y AA P	0.0012
NN D EE	0.0026	L EE RH	0.0016	IY RR IY	0.0012
AA RR AA	0.0026	AA LL I	0.0015	M AA KK	0.0012
LL AA NN	0.0025	Y L EE	0.0015	K L EE	0.0012
AA NN I	0.0024	AA KK AA	0.0015	EE T IY	0.0012
AA D AA	0.0023	I NN AA	0.0015	CH IY N	0.0012
RR I NN	0.0023	Y O RR	0.0015	EE C EE	0.0012
AA Y AA	0.0022	K AA RR	0.0015	S AA NN	0.0012
NN L A	0.0021	O RH SIL	0.0015	IY M IY	0.0012
IY Y EE	0.0021	O NN U	0.0015	N D IY	0.0011
NN IY NN	0.0020	O LL AA	0.0015	I Y O	0.0011
D EE NN	0.0020	EE G IY	0.0014	SIL B IY	0.0011
EE S IY	0.0020	NN B IY	0.0014	IY Y AA	0.0011
NN I NN	0.0020	U NN U	0.0014	T UE RR	0.0011
D AA NN	0.0020	Y AA NN	0.0014	EE D IY	0.0011
S I NN	0.0019	EE L EE	0.0014	LL IY K	0.0011
AA M AA	0.0019	B IY RR	0.0013	L A M	0.0011
IY L IY	0.0019	O RR U	0.0013	RR L EE	0.0011
AA LL AA	0.0019	AA NN D	0.0013	IY S T	0.0011
IY B IY	0.0019	IY K IY	0.0013	AA D I	0.0011
Y O RH	0.0019	SIL B U	0.0013	EE Y EE	0.0010
				M AA Y	0.0010

9,492). A comparison of most frequent triphones in our large text corpus and the new 2462-sentence-set is given in Table 3.2. The most frequent triphones from both corpora have been found to be highly correlated and the ranking orders are quite close to each other. This shows that the new sentence set highly reflects the phonetic behavior of Turkish.

3.3.2 Collecting the Audio Corpus

The audio corpus has been collected in the Department of Electrical and Electronics Engineering, at METU. For each speaker, a set of 40 sentences among 2462 sentences are selected randomly. This is the method which had been used to collect the TIMIT corpus [1]. To-date, speech from 193 speakers (89 female and 104 male) has been collected. The age range is from 19 to 50 years with an average of 23.9 years. Speakers are collected from students, faculty and staff at METU. The speech had been collected in office quality with a *Sennheiser* microphone, model *ME 102*. The data had been digitally recorded with a *Sound Blaster* sound card on a PC at a 16 kHz sampling rate. Each recording session is accompanied by a text file, which lists the 40 randomly selected sentences. In addition, the recording date, the age of the speaker and the geographical region of Turkey where the speaker has grown up is recorded. Part of an example text file can be seen in Appendix B. The geographical region distributions in the whole database are given in Table 3.4. The final audio corpus consists of audio files, associated text transcriptions and phone-level, word-level and HMM-state-level alignments. The files in the corpus are arranged as presented in Table 3.3. These alignments are provided by the phonetic aligner, which we have developed by porting CSLR's speech recognition tool *Sonic* [41] to Turkish. The aligner is explained in detail in Section 3.4. Each audio file in the audio corpus has been checked for misreadings and repetitions. In cases of misreadings, either the corresponding text file has been corrected or the sentence has been deleted completely.

Table 3.2: Comparison of the triphones in the 2.5 million-word text corpus and the 2462-sentence text corpus. Occurrence rates are normalized to add up to unity. Rankings show the descending order numbers.

METUbet TRIPHONE	RANKING IN 2.5M corpus	OCCURRENCE RATE (2.5M corpus)	RANKING IN 2462 sentences	OCCURRENCE RATE (2462 sentences)
EE RR IY	1	0.0047	2	0.0056
LL AA RR	2	0.0046	3	0.0052
L EE RR	3	0.0045	4	0.0050
B IY RH	4	0.0044	1	0.0060
NN D AA	5	0.0038	9	0.0033
IY NN IY	6	0.0034	6	0.0036
RR IY NN	7	0.0029	13	0.0029
AA RR I	8	0.0029	8	0.0034
IY L EE	9	0.0029	10	0.0030
I NN I	10	0.0028	21	0.0027
AA S I	11	0.0027	23	0.0026
IY NN D	12	0.0027	24	0.0026
I NN D	13	0.0027	39	0.0021
NN D EE	14	0.0026	19	0.0028
AA RR AA	15	0.0026	15	0.0029
LL AA NN	16	0.0025	17	0.0028
AA NN I	17	0.0024	20	0.0028
AA D AA	18	0.0023	5	0.0037
RR I NN	19	0.0023	28	0.0024
AA Y AA	20	0.0022	22	0.0027
NN L A	21	0.0021	27	0.0025
IY Y EE	22	0.0021	71	0.0016
NN IY NN	23	0.0021	61	0.0017
D EE NN	24	0.0020	38	0.0021
EE S IY	25	0.0020	30	0.0023
NN I NN	26	0.0020	70	0.0016
D AA NN	27	0.0020	44	0.0019
S I NN	28	0.0020	60	0.0017
AA M AA	29	0.0019	16	0.0028
IY L IY	30	0.0019	11	0.0030

Table 3.3: File arrangement in the 193-speaker audio corpus.

FILE TYPE	DESCRIPTION
SPK-ID-INFO.txt	speaker information and sentences
SPK-ID.raw	speech waveform file
SPK-ID.txt	orthographic transcription of the words the person said
SPK-ID.wrd	time-aligned orthographic word transcription
SPK-ID.phn	time-aligned phonetic transcription (METUbet)
SPK-ID.stat	time-aligned HMM-state transcription (METUbet)

Table 3.4: Speaker grown-up region distributions in the 193-speaker audio corpus.

GEOGRAPHICAL REGION	PERCENT in the AUDIO COPRUS
İç Anadolu	48%
Ege	16%
Marmara	16%
Akdeniz	10%
Karadeniz	8%
Güney Doğu Anadolu	1%
Doğu Anadolu	1%

3.4 Speech Recognition Tools Developed for Turkish

Two types of tools have been developed for Turkish: Phonetic aligner and phoneme recognizer. CSLR's speech recognition toolkit, *Sonic*, [41], has been ported to Turkish. The resulting port has aided in the development of our audio corpus that has been phonetically labelled at word, phoneme and HMM-state level. In Section 3.4.1, the structure of the *Sonic* speech recognizer toolkit will be explained briefly. Then, we will discuss the systems developed by porting *Sonic* to Turkish in Section 3.4.2.

3.4.1 The Sonic Continuous Speech Recognizer

Sonic is a toolkit for enabling research and development of new algorithms for continuous speech recognition, which is developed and used as a test bed for

research activities that include speech recognition as core components at the Center for Spoken Language Research (CSLR) of University of Colorado at Boulder [41].

The system acoustic models are decision-tree state-clustered HMMs with associated gamma probability density functions to model state-durations [46]. The recognizer implements a two-pass search strategy. The first pass consists of a time-synchronous Viterbi token-passing search. Cross-word acoustic models and trigram language models are applied in the first pass of search. During the second pass, the resulting word-lattice is converted into a word-graph. Sonic incorporates speaker adaptation and normalization methods such as Maximum Likelihood Linear Regression (MLLR), Parallel Model Adaptation, and Vocal Tract Length Normalization (VTLN).

Phonetic aligner module of *Sonic* provides word, phone, and HMM state-level boundaries for acoustic training. It includes decision-tree based trainable letter-to-sound prediction module, and multilingual lexicon support. Turkish has been the first language, which *Sonic* has first been ported to, through our work in this thesis [46]. Then it has been ported to many other languages (German, Spanish, French, Italian, Croatian, Arabic, Russian, Portuguese, Korean, and Japanese) [49].

Sonic has been benchmarked on several standard continuous speech recognition tasks for American English and has been shown to have competitive recognition accuracy to other recognition systems evaluated on similar data [46]. Performance metrics are shown in Table 3.5 [49].

TI-Digits contains speech which was originally designed and collected at Texas Instruments, Inc. (TI) for the purpose of designing and evaluating algorithms for speaker-independent recognition of connected digit sequences. It is microphone speech collected from 326 speakers each pronouncing 77 digit sequences [50]. DARPA Communicator is a real time spoken dialog system designed to recognize telephone quality speech in travel domain and respond [51]. Wall Street Journal (WSJ) is a corpus which contains microphone quality speech of approximately 78,000 training utterances (73 hours of speech), 4,000

Table 3.5: Word error rate for *Sonic* on several tasks: TI-Digits, DARPA Communicator, Nov'92 Wall Street Journal (WSJ) 5k test set and Swithboard task. Real-time factors are for first-pass decoding on an 800 MHz Intel Pentium III.

TASK	VOCABULARY SIZE	WORD ERROR RATE no adaptation	WORD ERROR RATE with adaptation	REAL-TIME FACTOR
TI-Digits	11	0.4%	0.2%	0.1
DARPA Communicator	2.1k	10.9%	-NA-	1.6
WSJ	5k	3.9%	3.0%	1.5
Switchboard	40k	41.9%	31.0%	9.1

of which are the result of spontaneous dictation by journalists with varying degrees of experience in dictation [52]. Switchboard is a corpus of conversational telephone speech which consists of 3,638 5-minute telephone conversations involving 657 participants [53].

3.4.2 Systems Developed for Turkish Speech Research

3.4.2.1 Phonetic Aligner

Sonic uses the Carnegie Mellon University's Sphinx-II phoneme symbol set [47]. Initialization of the recognizer's acoustic models to Turkish was performed by mapping Sphinx-II symbols to the acoustically nearest equivalents in METUbet. The mapping is shown in Table 3.6. This has been done simply by mapping the symbols which correspond to the nearest IPA symbols in the IPA chart [43]. We found that there was no acceptable mapping for the Turkish phoneme *GH* (ğ in orthography). *GH* usually lengthens the vowel it precedes or acts as a weak *Y* when it is between front vowels [10]. Therefore, we have not used it for the recognizer and the aligner applications. The aligner outputs *ğ* in word-level, but not in phone-level alignments, and instead outputs the previous vowel with its lengthened phoneme boundary.

Table 3.6: Mapping from Sphinx-II to METUbet phone set.

SPHINX-II PHONEME	ENGLISH EXAMPLE	METUbet PHONEME	SPHINX-II PHONEME	ENGLISH EXAMPLE	METUbet PHONEME
AA	*o*dd	AA, A	JH	*g*ee	C
AE	*a*t	EE	K	*k*ey	K, KK
AH	h*u*t		KD	lic*k*	
AO	*ou*ght		L	*l*ee	L, LL
AW	c*ow*		M	*m*e	M
AX	*a*bide	OE	N	*kn*ee	NN, N
AXR	use*r*		NG	pi*ng*	
AY	h*i*de		OW	*oa*t	O
B	*b*e	B	OY	t*oy*	
BD	du*b*		P	*p*ee	P
CH	*ch*eese	CH	PD	li*p*	
D	*d*ee	D	R	*r*ead	R, RR, RH
DD	du*d*		S	*s*ea	S
DH	*th*ee		SH	*sh*e	SH
DX	ma*tt*er		T	*t*ea	T
EH	*E*d	E	TD	li*t*	
ER	h*ur*t		TH	*th*eta	
EY	*a*te		TS	bi*ts*	
F	*f*ee	F	UH	h*oo*d	U, UE
G	*g*reen	G, GG	UW	t*wo*	
GD	ba*g*		V	*v*ee	V, VV
HH	*h*e	H	W	*w*e	
IH	*i*t		Y	*y*ield	Y
IX	ac*i*d	I	Z	*z*ee	Z, ZH
IY	*ea*t	IY	ZH	sei*z*ure	J

The 193-speaker Turkish audio corpus has been used to improve the accuracy of phonetic alignment system originally developed for English. Letter-to-sound (i.e., letter-to-METUbet) rules for Turkish have been used to develop a dictionary of the words in the 2462-sentence corpus. This dictionary has been integrated into Sonic. A set of decision tree questions has been developed for Turkish phonemes and it has been used for acoustic model training. 36 questions have been determined based on the place and the manner of the articulation of the METUbet phonemes based on the explanations given in [10]. Decision tree questions are shown in Table 3.7.

The first 100 speakers of the audio corpus were used to train the Turkish acoustic models of the aligner. Using the initial mapping, the corpus was aligned at the HMM-state level and models were then retrained using decision tree state clustering. The resulting aligner is capable of providing word-level and phoneme-level boundaries for Turkish. The phonemes are represented by METUbet symbols at the output of the aligner. For phoneme recognition experiments, the 2.5 million-word text corpus has been converted to METUbet phonemes using text-to-phoneme rules that we have developed. This corpus was used to develop a back-off trigram phoneme language-model. Figure 3.2 illustrates the block diagram of the aligner.

3.4.2.2 Phoneme Recognizer

For phoneme recognition experiments, the 2.5 million-word text corpus has been converted to METUbet phonemes using the letter-to-sound rules. This corpus was used to develop a back-off trigram phoneme language-model. Figure 3.3 illustrates the block diagram of the phoneme recognizer.

3.4.2.3 Evaluations on the Aligner

Experiments on the phonetic aligner and the phoneme recognizer were conducted by randomly selecting 20 speakers (10 male and 10 female) from the 193-speaker audio corpus. Those 20 speakers are not among the first 100 speak-

Table 3.7: Decision Tree Questions for Turkish.

QUESTION	ANSWER
silence	SIL
voiced	A, AA, E, EE, I, IY, O, OE, U, UE, B, C D, G, GG, L, LL, M, N, NN, R, RR, V, VV, Y
voiceless	CH, F, H, J, K, KK, L, LL, P, RH, S, SH, T, ZH
vowel	A, AA, E, EE, I, IY, O, OE, U, UE
back-vowel	A, AA, I, O, U
front-vowel	E, EE, IY, OE, UE
round-vowel	O, OE, U, UE
flat-vowel	A, AA, E, EE, I, IY
high-vowel	I, IY, U, UE
low-vowel	A, AA, E, EE, O, OE
back-flat-vowel	A, AA, I
front-flat-vowel	E, EE, IY
back-round-vowel	O, U
front-round-vowel	OE, UE
low-flat-vowel	A, AA, E, EE
high-flat-vowel	I, IY
low-round-vowel	O, OE
high-round-vowel	U, UE
plosive-consonant	B, P, T, D, K, KK, G, GG
nasal-cons	M, N, NN
stop-fricative	RH, ZH
rolled-cons	R, RR
lateral	L, LL
fricative	C, CH, F, H, J, S, SH, V, VV, Y, Z, ZH
bilabial	B, P, M
labiodental	F, V, VV
dental	D, T
palato-alveolar	NN, R, RR, S, Z, ZH
alveo-palatal	C, CH, J, SH, Y
palatal	L, LL
velar	K, KK, G, GG
glottal	H
front-cons	G, K, L
back-cons	GG, KK, LL

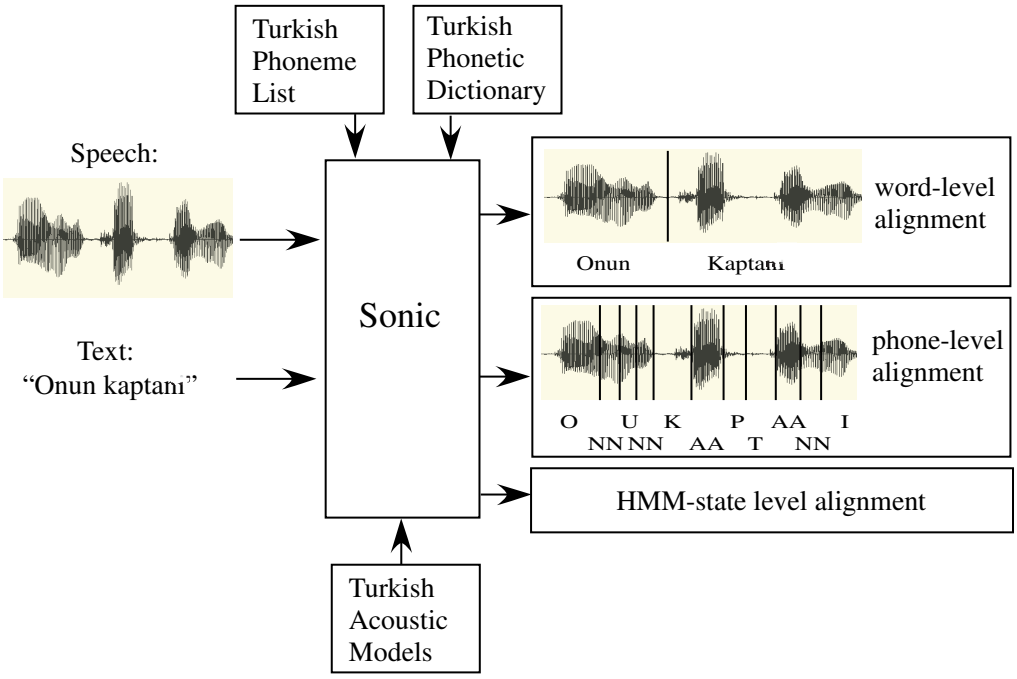


Figure 3.2: Block diagram of the Turkish phonetic aligner.

ers which were used for training. 40 sentences from each speaker were aligned using the Sonic-Turkish-phonetic-aligner. The alignments for each speaker were corrected by hand and compared to the alignments produced by the aligner. The quality of the automatic alignment has been measured by computing the absolute distance (in msec.) between the automatically determined and hand-labelled METUbet boundaries. For a boundary aligner-labelled at time τ and the hand-labelled boundary at time $\tilde{\tau}$, the misalignment is defined as $\varepsilon = |\tau - \tilde{\tau}|$. The overall segmentation performance is obtained by computing the percent of boundaries labelled correctly, where correctness means $\varepsilon < \triangle$, with $\triangle$ being a fixed distance [54]. Results are presented in Table 3.8. For comparison purposes, results from the same system for English are also provided in Table 3.8 [54]. For Turkish, 91.2% of the misalignments have been obtained within 20 msec. of the hand-labelled locations. The results obtained for the Turkish phonetic-aligner are quite comparable to those obtained for English [54].

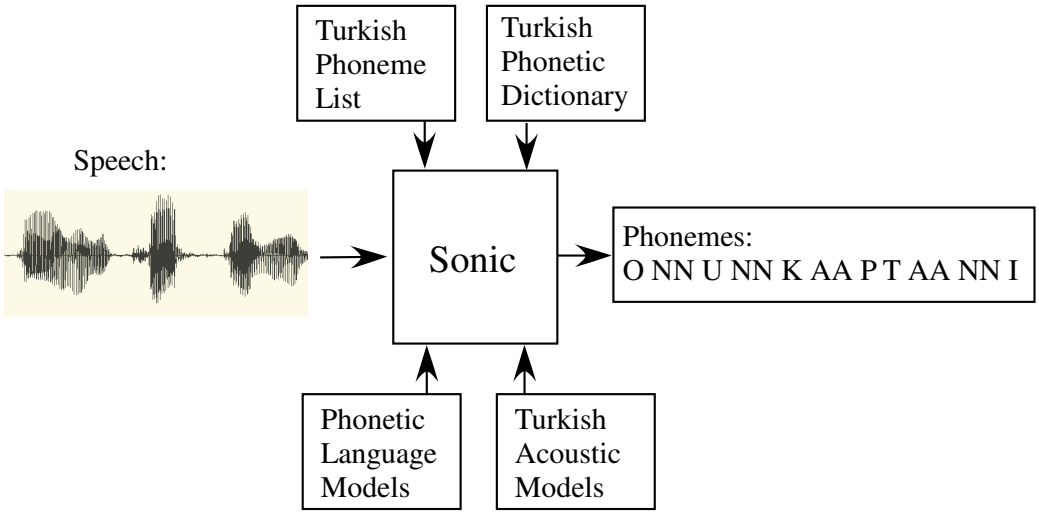


Figure 3.3: Block diagram of the Turkish phoneme recognizer.

Table 3.8: Comparison of the percentages of automatically placed phoneme boundaries within a fixed distance from the hand-labelled reference.

MISALIGNMENT FROM REFERENCE △	PERCENT OF AUTOMATICALLY PLACED PHONEME BOUNDARIES FOR TURKISH	PERCENT OF AUTOMATICALLY PLACED PHONEME BOUNDARIES FOR ENGLISH
≤5msec	53.7%	47.9%
≤10msec	67.6%	69.9%
≤20msec	91.2%	85.9%
≤40msec	98.1%	95.9%
≤60msec	99.3%	98.4%

3.4.2.4 Evaluations on the Phoneme recognizer

Phoneme recognition using decision tree state clustered HMMs has also been performed on the same test-set using a back-off trigram phoneme language model trained from the newspaper text corpus. Results of phoneme recognition experiments both with and without iterative unsupervised MLLR adaptation are shown in Table 3.9. Here we see that the overall phone error rate was found to be 29.3%. To the best of our knowledge, the only phone error rate of Turkish that has been reported is 44.1% [2] with 29 phonemes and without a phoneme language modelling. System summary percentages by speaker after the third pass are provided in Table 3.10. The average error rate decreased from 31.1% on the average to 29.3% from the first pass to third pass by speaker adaptation.

Table 3.9: Phone error rates for 20 Turkish speakers. Results are shown for a baseline system and the same system with iterative unsupervised MLLR adaptation.

GENDER	NON-ADAPTED	ADAPTED (MLLR)
Male	30.7%	29.1%
Female	31.5%	29.6%
Overall	31.1%	29.3%

Table 3.10: System summary percentages per speaker. Cor (correct), Sub (substitution), Del (deletion), Ins (insertion), Err (error) percentages are given.

SPEAKER ID	SENTENCE NUMBER	PHONE NUMBER	COR%	SUB%	DEL%	INS%	ERR%
s1082	38	1679	67.5	15.3	17.2	4.1	36.6
s1086	40	1885	71.6	15.3	13.1	2.4	30.8
s1087	40	1877	77.0	13.1	9.9	2.4	25.4
s1088	40	2004	67.0	16.3	16.7	1.3	34.3
s1089	39	1655	77.9	13.1	9.0	2.1	24.2
s1091	40	1837	73.4	16.2	10.4	2.0	28.6
s1093	40	1753	67.3	16.9	15.8	1.4	34.1
s1094	40	1891	77.0	13.1	9.9	2.0	24.9
s1096	40	1806	75.5	16.2	8.3	2.7	27.2
s1097	40	1801	76.8	14.3	8.9	1.7	24.8
s1129	40	1920	73.2	15.1	11.7	1.5	28.2
s1131	40	1825	70.1	16.7	13.2	1.3	31.2
s1132	38	1923	75.4	14.6	10.0	2.5	27.1
s1134	39	2054	64.1	20.4	15.5	2.2	38.1
s1135	39	1904	75.6	15.1	9.2	2.0	26.4
s1136	40	1877	71.1	16.8	12.1	1.4	30.3
s1137	40	1778	76.5	15.1	8.4	1.5	24.9
s1138	40	1935	67.4	19.9	12.6	2.0	34.6
s1140	40	2074	77.0	16.2	6.8	2.7	25.8
s1141	39	1804	74.1	15.2	10.6	2.5	28.4
Sum/Avg	792	37282	72.2	15.8	11.5	2.1	29.3

Table 3.10 presents correct detection, substitution, deletion, insertion and error percentages. The following are the explanations of these terms:

- A **substitution** error occurs when one phoneme in the *reference* (this is what should be recognized) is replaced by another phoneme in the *hypothesis* (this is what is recognized).

- An **insertion** error occurs when an extra phoneme is inserted in the hypothesis.
- A **deletion** error occurs when a phoneme is missing in the hypothesis.
- **phoneme_no** is the total number of phonemes in the reference.
- **Error percentage** is equal to $[(\sum(sub + ins + del))\ /phoneme\_no] \times 100$, where *sub, ins*, and *del* stand for the total numbers of substitution, insertion and deletion errors respectively.
- **Correct detection percentage** is the number of matching phonemes compared to the phoneme number as a percent.

CHAPTER 4

MELP-BASED SPECTRAL MODIFICATION FOR VOICE TRANSFORMATION

4.1 Introduction

This chapter introduces the baseline VT system we have implemented. Then we introduce the improvements applied on the baseline system. The system is designed to transform the spectral envelope of speech by changing spectral parameters of a source-filter model, using a mapping histogram matrix obtained during training.

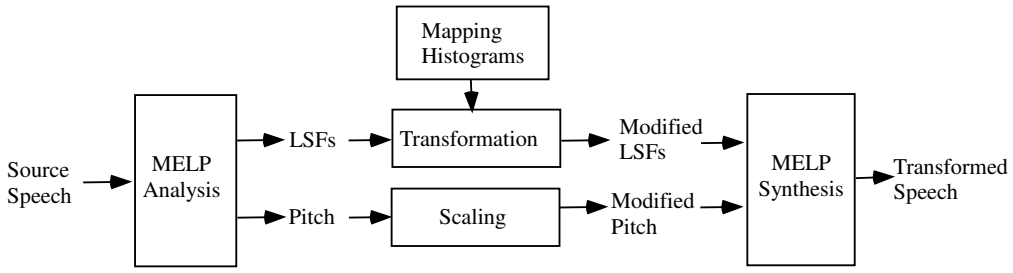


Figure 4.1: Overview block diagram for the voice transformation system.

Figure 4.1 illustrates an overview block diagram of the VT system in the transformation mode. Speech model is based on the *Mixed Excitation Linear Prediction* (MELP) coder [11]. The system first analyzes the source speaker's speech, then applies transformation on the analysis parameters, which are line spectral frequencies (LSFs), then re-synthesizes speech with the transformed

spectral parameters. Excitation parameters in this system are not modified, except for the pitch-period. Synthetic speech is expected to sound like the target speaker due to the transformation applied.

4.2 Speech Model

Speech model is based on the MELP coder [11] whose analysis and synthesis modules are based on the traditional *Linear Prediction Coding* (LPC) parametric model [12]. Therefore, the first part of this section explains the LPC-based source-filter model briefly. Basics of MELP speech coding algorithm are given in Section 4.2.2.

4.2.1 LPC-based Speech Model

The linear prediction method [12, 55] is a source-filter motivated approach for both analyzing and synthesizing the speech signal. The model assumes that a sample of speech can be approximated by a linear combination of P previous samples plus an additive excitation term,

$$s(n) = \sum_{k=1}^{P} a_k s(n-k) + Gu(n) \tag{4.1}$$

where P is referred to as the analysis order, a_k represents the k^{th} predictor coefficient, and $u(n)$ is the excitation scaled by a gain-factor, G. The excitation can be thought of as driving a passive linear vocal tract shaping filter, which is described by the predictor coefficients. During synthesis, the excitation can be either a series of periodic pulses for voiced speech or noise-like excitation for unvoiced sounds. The separation between the pulses for voiced speech determines the resulting pitch of the output signal. By expressing Equation 4.1 in terms of an input/output relationship, a $z-$transform domain expression for the vocal tract filter can be realized as,

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{A(z)} = \frac{G}{1 - \sum_{k=1}^{P} a_k z^{-k}} \tag{4.2}$$

where $S(z)$ and $U(z)$ represent the $z-$transforms of the speech signal and excitation, respectively. The denominator polynomial in $H(z)$ is often referred to as the analysis filter and is denoted by $A(z)$. A short-time speech waveform passed through the analysis filter will have an output which is approximately white noise for unvoiced speech and a periodic pulse train during voiced speech. A simple block diagram of the LPC-based model of speech is given in Figure 4.2.

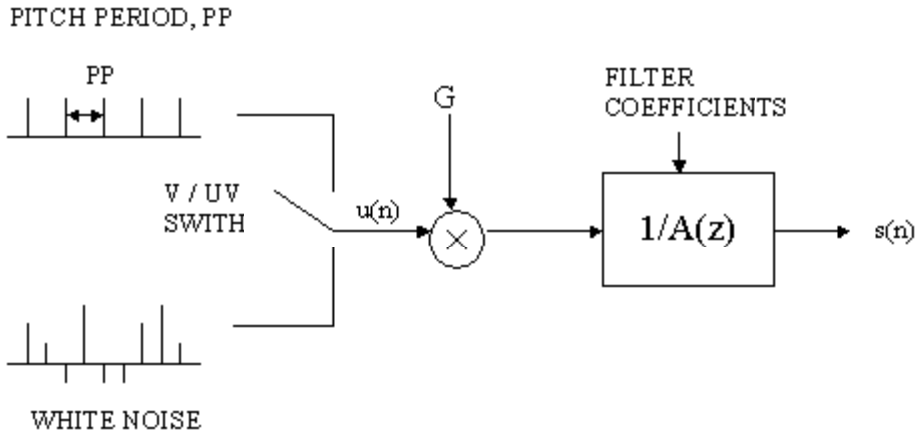


Figure 4.2: LPC-based model of speech production.

It has been shown that the LPC model of speech closely relates the acoustic tube model of the vocal tract, which assumes the vocal tract to consist of a set of P interconnected sections of equal length [12]. Using this relationship, an equation to determine the prediction order, P, for a given sampling rate and a vocal tract length has been obtained. This equation is given as [12, page 75]:

$$P = \frac{2Lf_s}{c}, \tag{4.3}$$

where L is the vocal tract length, f_s is the sampling frequency, and c is the speed of sound. If we assume that L is 17cm, which is reported as vocal tract length of an average male [55, page 39], and that the speed of sound is 340 m/s, then the prediction order, P, is obtained as 8 and 16, for sampling frequencies of 8 kHz and 16 kHz, respectively. MELP uses $P = 10$ for LPC analysis, which is also used throughout this thesis.

4.2.1.1 Line Spectrum Frequency (LSF) Pairs

The determination of the LPC parameters is straight forward and computationally efficient. But they have poor interpolation and quantization properties [56]. Consequently, alternative representations of LPCs such as reflection coefficients [12], log-area ratios [57, 12], etc. have been used for both speech coding and synthesis. LSFs also provide an equivalent representation of the linear predictor coefficients. LSFs have been used to represent the vocal tract parameters for VT throughout this thesis. The reason for selecting LSFs is that these parameters are closely related to formant frequencies [27] which carry speaker individualities. They have good interpolation properties and they are stable [58], [56, page 216]. In addition, they have a fixed dynamic range which makes them attractive for real-time DSP implementation [27]. MELP also uses LSFs for coding the vocal tract filter response [11].

LSFs were originally formulated by Itakura [59]. They are computed from the P^{th} order LPC analysis with the analysis filter given by,

$$A(z) = 1 - \sum_{k=1}^{P} a_k z^{-k}. \tag{4.4}$$

The filter order can be extended to $(P+1)$ without introducing any new information [60]. The resulting prediction error filter polynomials become,

$$P(z) = A(z) + z^{-(P+1)} A(z^{-1}) \tag{4.5}$$

$$Q(z) = A(z) - z^{-(P+1)} A(z^{-1}), \tag{4.6}$$

such that

$$A(z) = \frac{1}{2}\left[P(z) + Q(z)\right]. \tag{4.7}$$

The two resulting polynomials, P(z) and Q(z), can be thought of as the lossless acoustic tube representation of the vocal tract transfer function when the glottis is either completely closed or completely open [60]. It can be shown that the zeros of the polynomials $P(z)$ and $Q(z)$ are located on the unit circle and they are interlaced with each other [59]. Furthermore, the corresponding LPC filter is guaranteed to be stable if and only if these two conditions are

satisfied [59]. Since the zeros of $P(z)$ and $Q(z)$ lie on the unit circle, each can be expressed as e^{jw_i} where w_i is known as an LSF. A P^{th} order LPC analysis results in $2P$ frequencies which are positioned in complex conjugate locations on the unit circle. Therefore, it is sufficient to consider the first P frequencies ranging from $[0, \pi]$. Since zeros of $P(z)$ and $Q(z)$ are interleaved along the unit circle, the polynomial $P(z)$ will have roots at frequencies $\{w_1, w_3, \ldots, w_{P-1}\}$ while the polynomial $Q(z)$ will have roots at the frequencies $\{w_2, w_4, \ldots, w_P\}$. The roots found inside the interval $[0, \pi]$ are ordered such that $\{0 < w_1 < w_2 < w_3 < \ldots < w_{P-1} < w_P < \pi\}$. Thus a P^{th} order LPC analysis will result in a vector of LSF parameters containing P ordered frequency locations spanning the frequency range $[0, \pi]$.

In [61], it has been shown that, a cluster of (2 or 3) LSFs characterizes a formant frequency, and the bandwidth of a given formant depends closely on the closeness of the corresponding LSFs. In addition, the spectral sensitivities of LSFs are localized; i.e., a change in a given LSF produces a change in the LPC power spectrum only in its neighborhood [61]. These properties make LSFs proper for spectral modification applications such as VT.

4.2.2 The MELP Speech Coder

MELP is designed to convert analog voice to 2,400 bits/s digitized voice and to reconvert back to analog voice. The analysis and synthesis of MELP is based on the traditional LPC parametric model [12]. MELP also uses some additional features such as mixed excitation and aperiodic pulses to model the excitation, and this allows the coder to mimic some characteristics of natural human speech [62]. MELP is reported to produce natural sounding speech even in a difficult noise environment [62].

The MELP decoder block diagram is given in Figure 4.3. The features illustrated in the figure are explained briefly below [11]:

The **mixed excitation** is generated using a multi-band mixing model. The primary effect of the mixed excitation is to reduce the buzz usually associated

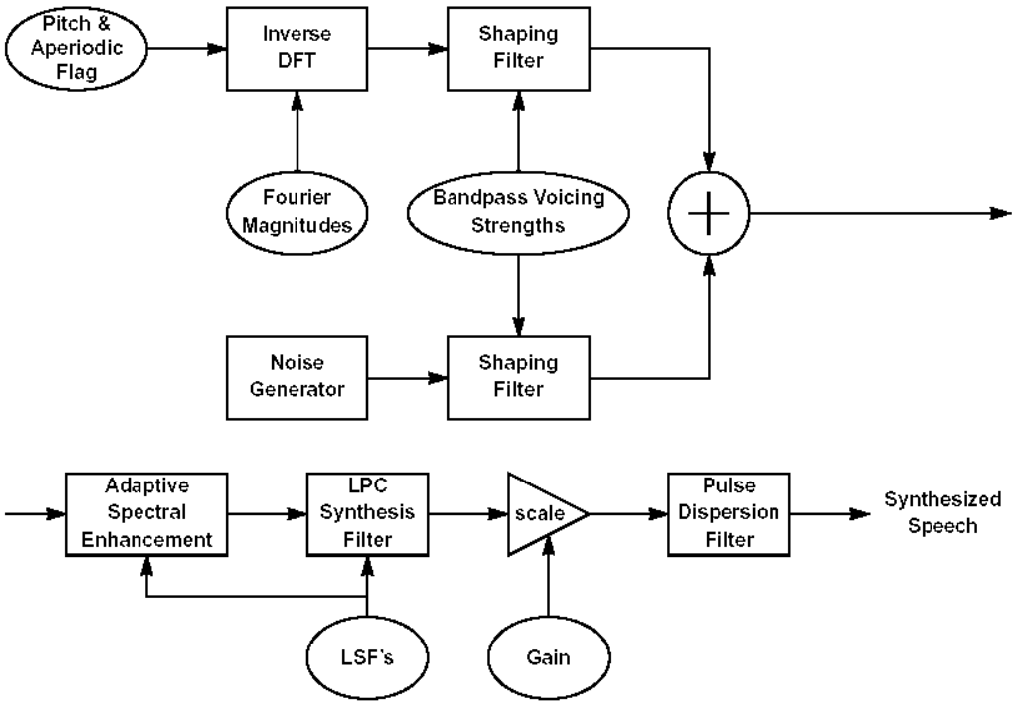


Figure 4.3: MELP decoder block diagram.

with LPC-based synthesizers. The periodic pulse train and the noise excitation are passed through bandpass filters and then added together to give a full-band excitation. For each frame (180 voice samples with a sampling rate of 8,000 samples/sec), the frequency shaping filter coefficients are generated by a weighted sum of each of the bandpass filters. These filters are 6^{th} order *Butterworth* filters with passbands of $0-500$, $500-1000$, $1000-2000$, $2000-3000$, and $3000-4000$ Hz. The pulse filter is calculated as the sum of each of the bandpass filters weighted by the voicing strength in that band. The noise filter is generated by a similar weighted sum, with weights set to keep the total noise and pulse power constant in each frequency band. The weights are determined by the voicing strengths in each band.

When the input speech is voiced, the MELP decoder can use either **periodic** or **aperiodic pulses**. Aperiodic pulses are most often used during transition regions between voiced and unvoiced segments of the speech signal. The voicing strength of the first band determines whether periodic or aperiodic pulses should be used. This feature enables the decoder to reduce erratic

glottal pulses without introducing tonal sounds [62].

The **adaptive spectral enhancement** filter is based on the poles of the linear prediction synthesis filter. Its use enhances the formant structure of the synthetic speech, and it is applied to give the synthetic speech a more natural quality [11].

The **pulse dispersion** is implemented using a fixed filter based on a spectrally-flattened triangle pulse. This filter spreads the excitation energy within a pitch period reducing the harsh quality of synthetic speech.

A 10^{th} order LPC analysis is performed on the input speech using a 200-sample (25 ms) Hamming window centered on the last sample in the current frame. Then the linear prediction coefficients are converted into LSFs. For each frame, an LSF vector f of length 10, is obtained. f is quantized using a multi-stage vector quantizer (MSVQ). The MSVQ codebook consists of four stages of 128, 64, 64, and 64 levels respectively. The quantized vector, $\hat{f}$, is the sum of the vectors selected by the search process, where one vector selected from each stage. The MSVQ search finds the codebook vector which minimizes the square of the weighted Euclidian distance, d^2, between the unquantized and quantized LSF vectors [11]:

$$d^2(f, \hat{f}) = \sum_{i=1}^{10} w_i (f_i - \hat{f}_i)^2, \tag{4.8}$$

where

$$w_i = \begin{cases} P(f_i)^{0.3}, & 1 \leq i \leq 8 \\ 0.64 P(f_i)^{0.3}, & i = 9 \\ 0.16 P(f_i)^{0.3}, & i = 10 \end{cases} . \tag{4.9}$$

f_i is the i_{th} component of the unquantized LSF vector, and $P(f_i)$ is the inverse prediction filter power spectrum evaluated at frequency f_i. The search procedure is an *M-best approximation* to a full search [63], in which $M = 8$ best code vectors from each stage are saved for use with the next stage. Then a process to ensure ascending order of LSFs and minimum separation of 50 Hz between

LSFs is applied to the quantized LSF vector [11]. This resulting vector is used in the Fourier magnitude computation in Figure 4.3.

Equation 4.8 represents a weighted Euclidian distance measure. The weights are assigned to a given LSF vector proportional to the value of LPC power spectrum at this LSF. Thus this distance measure allows for quantization of LSFs in the formant regions better than those in the non-formant regions. Also, the distance measure gives more weight to the LSFs corresponding to the high-amplitude formants than to those corresponding to the lower amplitude formants. The LSFs corresponding to the valleys in the power spectrum get the least weight. Since human ear cannot resolve differences at high frequencies as accurately as at low frequencies, more weights are assigned to the lower LSFs than to the higher LSFs. Similar distance measures had also been used for speech coding and recognition purposes, and good results had been obtained [61, 64]. Note that weights are specific to every LSF vector and vary from frame-to-frame.

Bandpass voicing quantization is achieved on the voicing strengths (which is the normalized autocorrelation value computed over the speech signal in that band) on each one of the five passbands of the speech signal. When the first band ($0 - 500$ Hz) is unvoiced (i.e., the passband strength is found to be smaller than 0.6), the rest of the bands are quantized to 0. When the first band is voiced, the remaining voicing bands are quantized to 1 if their value is greater than or equal to 0.6, and to 0 otherwise [11].

4.3 Speech Data and Time-Alignment

Speech data for our VT research consists of 235 sentences collected from two male speakers of Turkish. Sentences are selected randomly from the phonetically-balanced 2462-sentence text corpus described in Chapter 3. This makes approximately 15 minutes of speech and more than 30,000 non-overlapping MELP analysis frames for each speaker, after silence frames are removed. Both speakers have read the same sentence set. 230 sentences have been used as the

training set and the rest 5 sentences as the test set. Speech has been collected in a quite office environment with a *Sehnheiser ME-64* microphone and sampled at 16 kHz.

In natural speech, the durations of the speech units vary from speaker to speaker. During training, the system tries to estimate a transformation function which can predict the features of the target speaker from the features of the source speaker. Therefore, the feature streams from the source and the target speaker should be time-aligned to obtain the transformation function that gives the relationship between the source and target features of equal phonetic context. The goal of time-alignment is to modify the source and the target speaker feature stream in such a way that the resulting feature streams describe approximately the same phonetic content. Time-alignment in this thesis has been achieved by selectively deleting or repeating frames from the target speaker feature stream to match the number of source frames within phonetically equivalent regions, through a *dynamic time warping* (DTW) algorithm [35, page 383].

All sentences of each speaker have been aligned in phoneme level using the Turkish phonetic aligner described in Chapter 3. After obtaining the phoneme-level alignments, the speech has been re-sampled at 8 kHz for MELP analysis. For every MELP frame, quantized components of multistage vector quantization (MSVQ), and bandpass voicing values (BPVC) have been extracted. Using the phoneme-level alignments, every MELP frame has been associated to a phoneme. Figure 4.4 illustrates examples of phoneme-aligned MSVQ frame files for each speaker. The aligner analyzes signals sampled at 16 kHz, with frames of length 320 samples. The frames are overlapping by 50% (shifted by 160 samples). MELP, on the other hand, analyzes speech sampled at 8 kHz and with non-overlapping frames of length 180 samples. Association has been achieved by mapping the nearest MELP and alignment frame centers. This file preparation phase has been repeated separately for each speaker.

The next step is the time-alignment between two speakers which has been

Frame	msvq1	msvq2	msvq3	msvq4	phone
:	:	:	:	:	:
34	43	14	14	58	SIL
35	43	22	21	3	SIL
36	114	51	21	34	0
37	13	34	3	13	0
38	93	41	19	58	0
39	20	0	50	63	0
40	76	56	58	5	0
41	125	0	56	37	NN
42	61	0	27	40	NN
43	125	8	49	53	NN
44	61	0	29	14	U
45	124	0	55	24	NN
46	116	56	61	12	NN
47	108	39	9	57	NN
48	120	39	34	21	KK
49	91	19	8	29	KK
50	25	3	32	50	KK
51	93	47	53	19	KK
52	93	47	35	62	AA
:	:	:	:	:	:

Speaker 1

Frame	msvq1	msvq2	msvq3	msvq4	phone
:	:	:	:	:	:
16	111	14	26	14	SIL
17	111	14	24	15	SIL
18	111	2	6	37	0
19	12	57	43	12	0
20	20	50	32	25	0
21	20	33	44	29	0
22	20	34	44	28	0
23	92	40	45	53	0
24	28	35	0	28	0
25	61	40	32	20	NN
26	20	47	14	17	U
27	20	17	35	2	U
28	4	17	38	0	U
29	52	16	43	30	NN
30	125	18	50	37	NN
31	61	31	52	5	KK
32	108	21	44	41	KK
33	91	3	42	29	KK
34	123	38	33	0	KK
:	:	:	:	:	:

Speaker 2

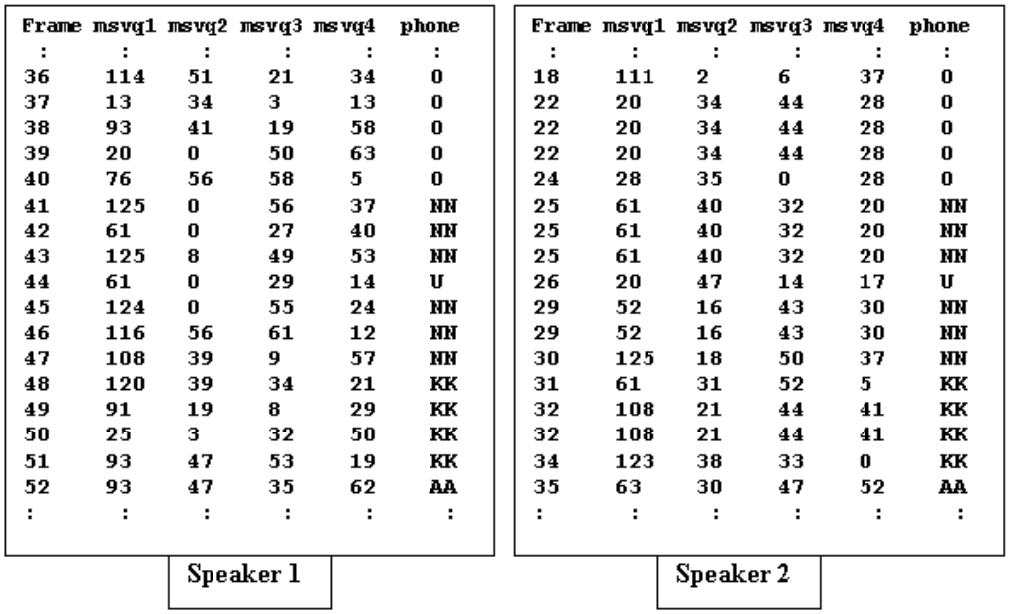
Figure 4.4: 4-stage vector quantization of LSF's in MELP with phoneme alignments for the two speakers. The first column shows the frame number, the next 4-columns show the corresponding MSVQ indices. The final column shows the associated METUbet phonemes.

achieved by DTW algorithm applied between frames of the phoneme groups of the two speakers. The time-warped form of the MSVQ frame lists given in Figure 4.4 are presented in Figure 4.5. Note that the frame numbers in each phoneme group are equated after time-alignment. The distance criterion we have used in the DTW algorithm is the Bark-weighted RMS error in dB between the power spectra of the two speakers at those frames [65]. This distance is given as:

$$SD(A_s(\omega), A_t(\omega)) = \sqrt{\frac{1}{\pi W_0}\int_0^{\pi} |W_B(\omega)|^2 \left| 10\log_{10}\frac{|A_s(\omega)|^2}{|A_t(\omega)|^2}\right|^2 d\omega} \qquad (4.10)$$

where $A_s(\omega)$ and $A_t(\omega)$ are the source and target LPC filters obtained from a MELP frame. W_0 normalizes $W_B(\omega)/W_0$ to unity RMS. The weights $W_B(\omega)$ are the Bark weights given as:

$$W_B(\omega) = \frac{1}{25 + 75\left(1 + 1.4(\frac{F_s\omega}{2000\pi})^2\right)^{0.69}} \qquad (4.11)$$



Frame	msvq1	msvq2	msvq3	msvq4	phone
:	:	:	:	:	:
36	114	51	21	34	0
37	13	34	3	13	0
38	93	41	19	58	0
39	20	0	50	63	0
40	76	56	58	5	0
41	125	0	56	37	NN
42	61	0	27	40	NN
43	125	8	49	53	NN
44	61	0	29	14	U
45	124	0	55	24	NN
46	116	56	61	12	NN
47	108	39	9	57	NN
48	120	39	34	21	KK
49	91	19	8	29	KK
50	25	3	32	50	KK
51	93	47	53	19	KK
52	93	47	35	62	AA
:	:	:	:	:	:

Speaker 1

Frame	msvq1	msvq2	msvq3	msvq4	phone
:	:	:	:	:	:
18	111	2	6	37	0
22	20	34	44	28	0
22	20	34	44	28	0
22	20	34	44	28	0
24	28	35	0	28	0
25	61	40	32	20	NN
25	61	40	32	20	NN
25	61	40	32	20	NN
26	20	47	14	17	U
29	52	16	43	30	NN
29	52	16	43	30	NN
30	125	18	50	37	NN
31	61	31	52	5	KK
32	108	21	44	41	KK
32	108	21	44	41	KK
34	123	38	33	0	KK
35	63	30	47	52	AA
:	:	:	:	:	:

Speaker 2

Figure 4.5: Frames in Figure 4.4 after time-alignment by DTW.

where F_s is the sampling frequency in Hz.

Bark weighting has been reported to have greater correlation with subjective evaluations because of down-weighting the higher frequencies [65]. LPC filters are computed directly over the speech files using the autocorrelation method, instead of using the quantized LSFs of MELP, to obtain a more accurate warping.

After the alignment, we collect the aligned feature vectors (MSVQ and BPVC vectors from MELP analysis) into the N frames of source data,

$$X^{MSVQ} = \begin{bmatrix} x_1^{MSVQ} \\ x_2^{MSVQ} \\ \vdots \\ x_N^{MSVQ} \end{bmatrix}_{N\times 4}, \; X^{BPVC} = \begin{bmatrix} x_1^{BPVC} \\ x_2^{BPVC} \\ \vdots \\ x_N^{BPVC} \end{bmatrix}_{N\times 5}, \tag{4.12}$$

and, respectively, the target data,

$$Y^{MSVQ} = \begin{bmatrix} y_1^{MSVQ} \\ y_2^{MSVQ} \\ \vdots \\ y_N^{MSVQ} \end{bmatrix}_{N\times 4}, \; Y^{BPVC} = \begin{bmatrix} y_1^{BPVC} \\ y_2^{BPVC} \\ \vdots \\ y_N^{BPVC} \end{bmatrix}_{N\times 5}. \tag{4.13}$$

The data matrices X^{MSVQ} and Y^{MSVQ} contain 4-stage MSVQ quantization indices as illustrated in Figure 4.5, and similarly X^{BPVC} and Y^{BPVC} include BPVC quantization values (0 for an unvoiced band and 1 for a voiced band) for 5 frequency bands explained in Section 4.2.2. Beginning and end silences of the sentences are not included in the training data sets. The value of N depends on the amount of available data from both source and target speaker. In our experiments, $N \geq 30,000$.

4.4 Training

In this section, we first present our observations on the LSF quantization of MELP speech coding algorithm. Then we explain our method of training. During the training mode of the system, we obtain mapping histograms relating the source feature vectors to those of the target. Those histograms are used to transform speech, as will be presented in the next section.

4.4.1 MSVQ Representation of MELP and Observations

In the training mode, the system obtains a transformation function to map the source speaker's speech features, X^{MSVQ}, to an estimate of the corresponding target speaker's speech features, Y^{MSVQ}. Note that the elements of feature matrices in our case are not the LSF values themselves, but the stage indices corresponding to the multi-stage LSF codebook of MELP. First, two codewords of each stage are illustrated in Table 4.1 to give an insight about LSF coding of MELP.

Table 4.1: Examples of codewords for multistage LSF quantization of MELP rounded to two-digit decimals. First two codewords from each MSVQ stage are illustrated. The first column shows the stage number (stg), the second column shows the index number (ind), and the other columns show the frequency vectors (vect).

stg	ind	vect (1) Hz	(2) Hz	(3) Hz	(4) Hz	(5) Hz	(6) Hz	(7) Hz	(8) Hz	(9) Hz	(10) Hz
1	0	355.24	492.66	635.39	980.35	1837.26	2145.00	2413.63	2740.71	3093.75	3368.61
1	1	484.73	640.39	823.55	1338.89	1880.24	2096.15	2435.88	2765.30	3111.36	3418.14
2	0	-1.08	-11.58	2.31	-14.00	-69.46	37.77	-72.56	-159.22	-339.82	-152.12
2	1	-49.65	-46.25	-78.10	-42.88	87.14	6.81	-84.40	-169.55	114.69	84.98
3	0	32.98	12.95	-77.52	3.46	21.12	-18.52	8.59	-32.46	79.44	57.10
3	1	20.02	21.87	-34.91	79.79	42.79	37.29	53.15	-12.29	-122.97	-61.57
4	0	13.86	-19.74	20.32	-1.37	-24.03	55.40	-33.43	-7.16	-76.74	-66.68
4	1	-22.29	-34.19	20.70	-14.17	28.37	-25.98	-43.56	15.94	46.56	34.64

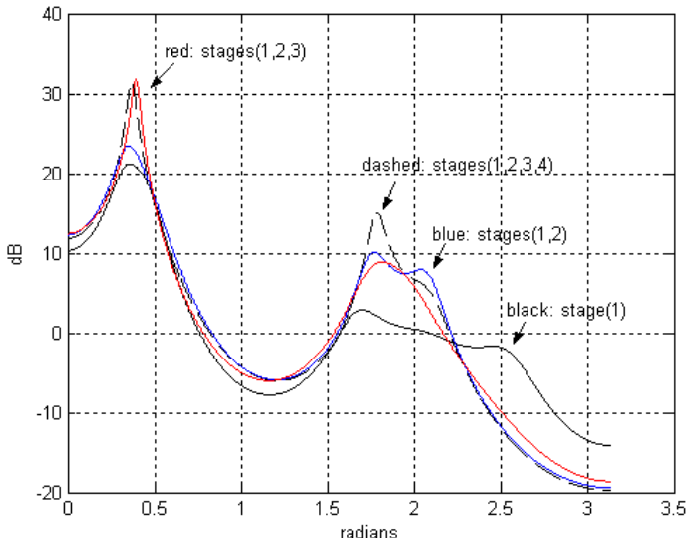


Figure 4.6: Effect of adding the 1_{st} vectors of 2^{nd}, 3^{rd}, and 4^{th} stages of MSVQ to the 1^{st} LSF vector of the 1^{st} stage. Corresponding filter responses are black: stage1, blue: stage1+stage2, red: stage1+stage2+stage3, dashed: stage1+stage2+stage3+stage4.

As presented in Table 4.1, the first stage vectors are the main LSF frequencies, and the vectors in the other stages are added to the first vector to fine-tune the LPC spectrum obtained from the first stage. Figure 4.6 presents the LPC

filter response obtained from the first LSF vector of the first stage. The effect of adding the other stages is illustrated by plotting their filter responses on top of it.

Before determining the method for mapping histograms to map the LSF indices of one speaker to the other, we have made some observations on the LSF quantization of MELP to investigate whether MSVQ indices of MELP carry speaker individualities.

4.4.1.1 Speaker Individualities in the First Stage of MSVQ of MELP

After the DTW process, number of frames belonging to every phoneme of the source and the target speakers are equated. Although they are uttering the same phonemes, MSVQ indices assigned to the same frames of the two speakers are different due to the speaker individualities. This is observed in Figure 4.5. Another observation is that since spectral behavior of phonemes are affected by the context, same phonemes in different context are not quantized to the same MSVQ indices for one speaker. This can be observed in Figure 4.7 and Figure 4.8 which illustrate phoneme occurrence rates (histograms) in our VT corpus for the first stage of MSVQ belonging to the first and the second speakers, respectively. The same phonemes tend to gather around certain MSVQ indices, which is expected since their LPC spectrums should present similar formant structures. The existence of the distribution observed in Figures 4.7 and 4.8, on the other hand, is due to the context difference and also the multi-stage vector selection of MELP (i.e., the first index of MSVQ does not have to be the nearest first stage vector to the original speech spectrum, the selection of the first is effected by the other stages during the M-best selection). Histograms for the same phonemes are different for different speakers, which means that the first stage LSFs of MSVQ carry speaker individualities. Figure 4.9 presents the occurrence rates of the corresponding indices of the two speakers in the aligned VT corpus.

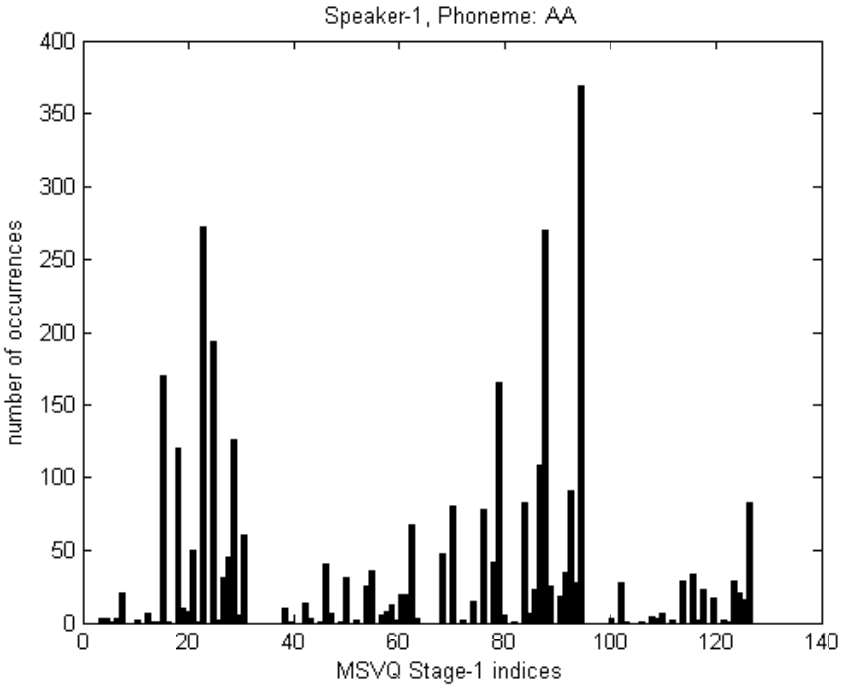


Figure 4.7: MSVQ 1^{st}-stage index histogram for phoneme AA of speaker-1.

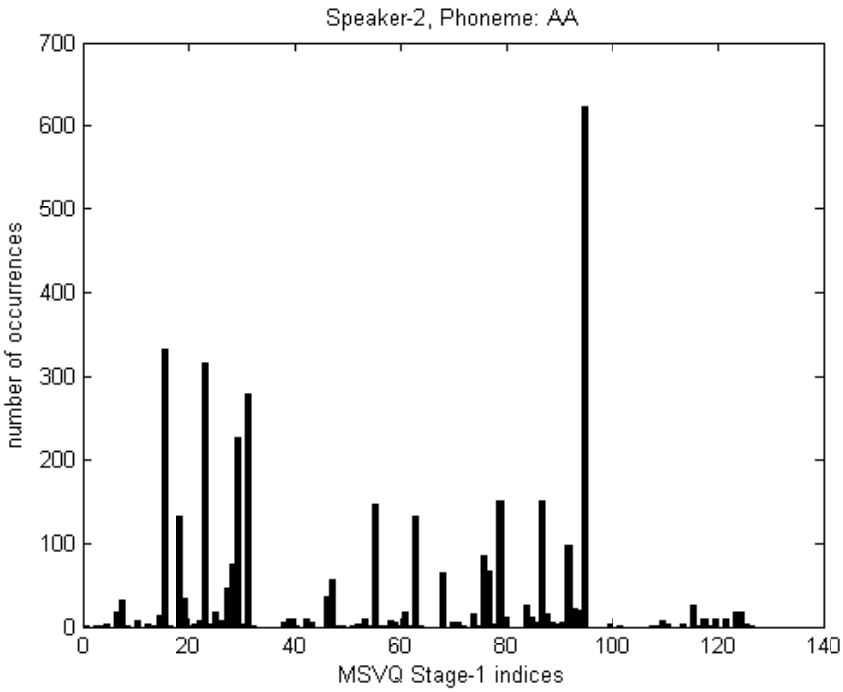


Figure 4.8: MSVQ 1^{st}-stage index histogram for phoneme AA of speaker-2.

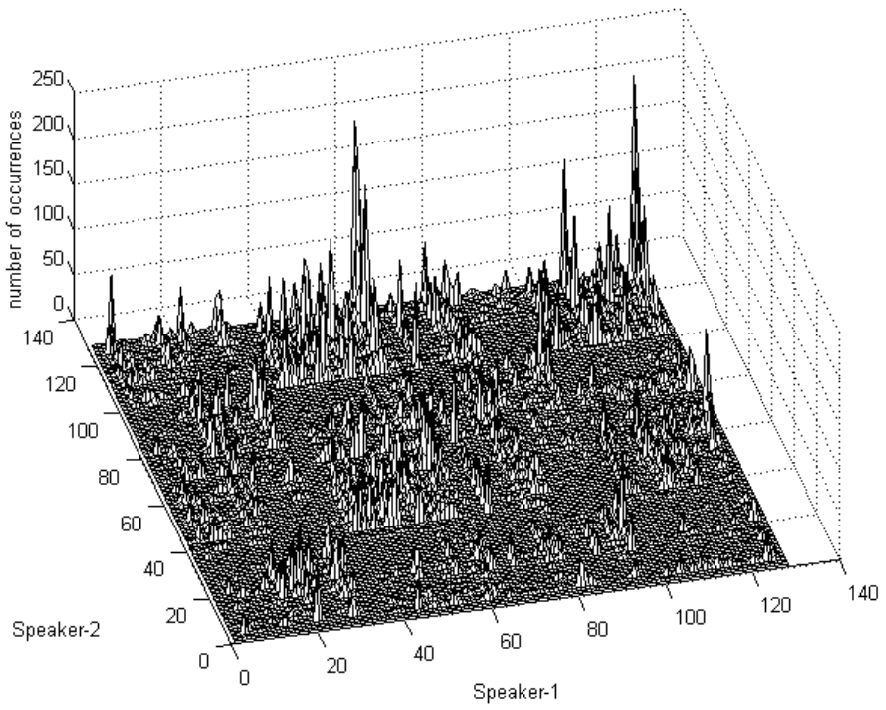


Figure 4.9: Histogram matrix mapping occurrence rates of the 1^{st} stage indices of MSVQ of the two speakers.

4.4.1.2 Personalizing the MSVQ Codebook

The idea of mapping codebooks for VT has been used widely in the literature as presented before in Chapter 2. Codebooks are usually obtained individually from speakers [24, 27]. If we use only the first stage of MSVQ for mapping the LSF space of the two speakers, we will be encountering two types of problems. The first problem is that 128 LSFs of the first stage are not speaker-specific; therefore, some indices are never used by some speakers as observed in Figure 4.9. This is inefficient in terms of quantizing the LSF space of a speaker, and hence in terms of capturing the speaker individualities. The second problem is that the parameter space of the converted envelope is limited to a discrete set of envelopes, which are only the first stage LSFs. Neglecting the spectral details in the synthetic speech will result in a degradation in the speech quality. Therefore, at least two or more of the MSVQ stages should be considered.

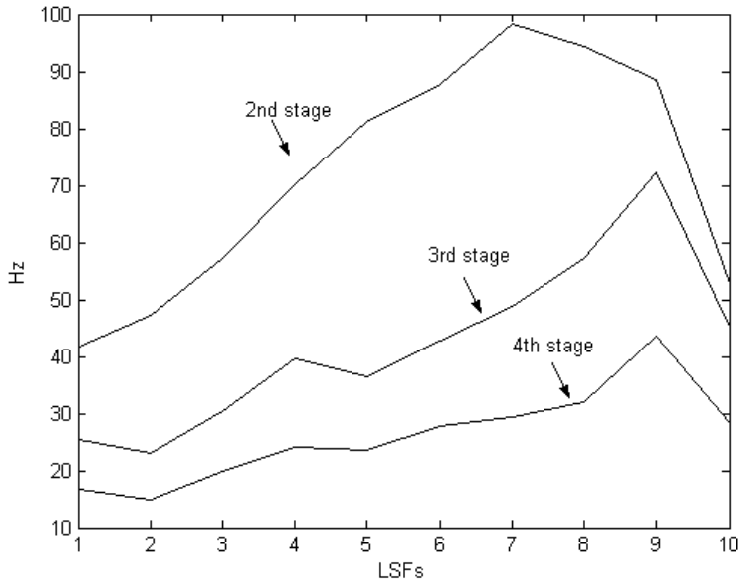


Figure 4.10: Means of the absolute values of the frequencies in 2^{nd}, 3^{rd} and 4^{th} stages of MELP's MSVQ, $msvq_i(k)$ where $i = 2, 3, 4$.

To observe the effect of each stage on the final LPC spectrum of speech, we have computed the mean of the absolute values of the frequencies included in 2^{nd}, 3^{rd}, and 4^{th} stages of MSVQ. If we represent the vectors in the i^{th} stage in the MSVQ codebook with $msvq_i(k)$, where k runs from 1 to 64 for the 2^{nd}, 3^{rd}, and 4^{th} stages, then the means of the absolute frequency values are computed as:

$$\overline{msvq}_i = \frac{1}{N} \sum_{k=1}^{N} |msvq_i(k)|, \tag{4.14}$$

where $N = 64$. The result is presented in Figure 4.10. This figure shows that, each stage of MSVQ is more important than the next one in terms of contribution to the LPC spectrum.

MSVQ codebook of MELP includes 128 LSF vectors for the first stage, and 64 frequency vectors for each of the remaining 3 stages. This makes $128 \times 64 \times 64 \times 64 = 33,554,432$ possible codewords for LSF quantization. With our sample space of size approximately $30,000$ vectors obtained from the VT corpus, it is impossible to obtain occurrence histograms which reflect the ac-

tual usage probabilities for a speaker. The data problem could be overcome by considering the first two or three stages, and neglecting the effect of the 4^{th} stage, and limiting the number of quantized LSFs used by the speaker according to the usage distribution specific to that speaker. When only the 1^{st} and the 2^{nd} stages are considered, it has been observed that out of $128 \times 64 = 8192$ LSF vectors, only 1600 were enough to cover more than 80% of the LSF space for both speakers. Those 1600 vectors are different for each speaker, which means that they carry more speaker-specific information. Moreover, we could force MELP to use only those 1600 vectors and determine the 3^{rd} and 4^{th} stage indices accordingly. We have computed the mutual information between the LSF vector indices of the source and the target speakers to support these ideas.

4.4.1.3 Mutual Information Computations

In this section, mutual information computations on the VT corpus will be presented to investigate the speaker individualities on LSF quantization of MELP. Preliminary basic information on the mutual information concept is given in Appendix C.

Definition: *Mutual Information*

Mutual information is a measure of the amount of information that one random variable contains about another random variable. It is the reduction in the uncertainty of one random variable due to the knowledge of the other. Let X and Y be two discrete random variables with alphabets χ and ψ. Their probability mass functions are given as $p(x) = Pr\{X = x\}$, $x \in \chi$, and $p(y) = Pr\{Y = y\}$, $y \in \psi$, respectively. Mutual information, $I(X;Y)$, is the relative entropy between the joint distribution, $p(x,y)$, and the product distribution, $p(x)p(y)$, i.e.,

$$I(X;Y) = \sum_{x \in \chi} \sum_{y \in \psi} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}. \tag{4.15}$$

It can also be shown that $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$ [66], where $H(X)$ and $H(Y)$ are the entropy functions of the random variables

Table 4.2: Mutual Information (MI) between the 1^{st} and the 2^{nd} stages of the source and the target speakers. MP-MI stands for maximum possible MI

	MI in bits	MP-MI	MI/MP-MI
$I(X_1;Y_1)$	1.17	7	0.17
$I(X_1;Y_1)$ (voiced frames)	1.14	7	0.16
$I(X_1;Y_1)$ (unvoiced frames)	1.19	7	0.17
$I(X_2;Y_2)$	0.21	6	0.04
$I(X_2;Y_2)$ (voiced frames)	0.24	6	0.04
$I(X_2;Y_2)$ (unvoiced frames)	0.44	6	0.07

X and Y respectively. $H(X|Y)$ and $H(Y|X)$ are the conditional entropies.

Mutual Information Computations

In our experiments, we have considered the distribution of the MSVQ stage indices of the first speaker as $p_X(x)$ and that of the target as $p_Y(y)$ (we will refer to them as $p(x)$ and $p(y)$ for convenience). x and y will belong to a set running from 1 to 128 for the first stage, from 1 to 64 for the second stage and so forth. We are approximating the actual probability functions from the empirical probabilities obtained from those histograms which give the number of occurrences of each frame in the VT corpus.

Let us define the empirical probabilities of the source and the target as $p_i(x)$ and $p_i(y)$, respectively, corresponding to the random variables X_i and Y_i, where i denotes the stage number. The joint probabilities $p_i(x, y)$ are obtained from the cross occurrence histograms (see Figure 4.9). $p_1(x)$ and $p_1(y)$ are shown in Figure 4.11. Mutual information, $I(X_i;Y_i)$, is computed using Equation 4.15. The results are given in Table 4.2.

Results in Table 4.2 show that the second stages of the MSVQ indices of the two speakers do not give enough information about each other. Mapping the stages independently would not give satisfactory results in terms of VT.

Then we have investigated the situation when the first two stages of MSVQ are considered dependently. For this experiment, all 8192 possibilities of the

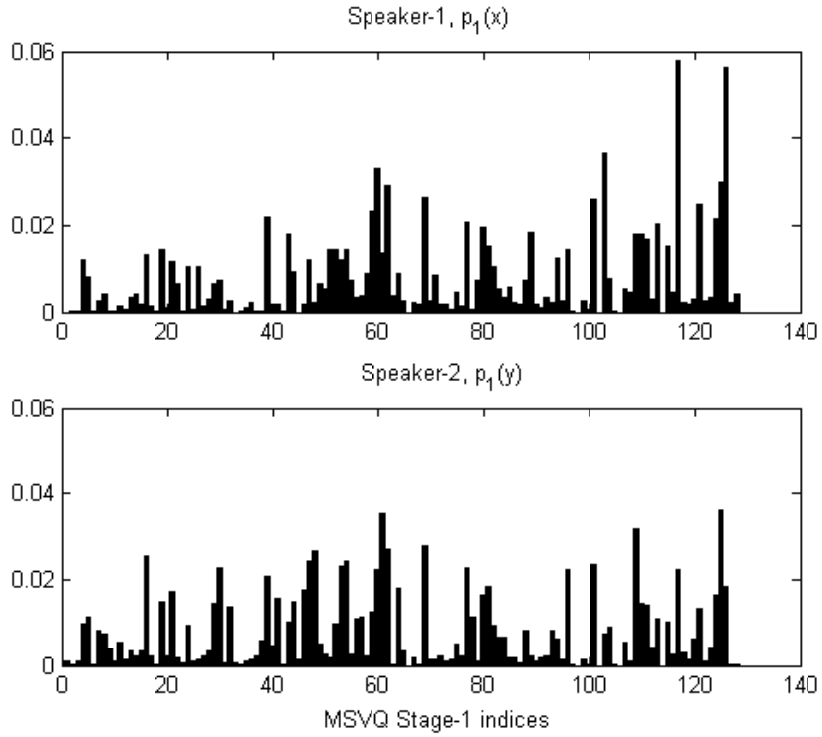


Figure 4.11: Empirical occurrence probabilities of the MSVQ Stage-1 indices obtained from X^{MSVQ} and Y^{MSVQ} data matrices of the two speakers, $p_1(x)$ and $p_1(y)$.

summation of the two stages are considered for both of the speakers as:

$$\{msvq_1(1) + msvq_2(1),\ msvq_1(1) + msvq_2(2), \cdots, msvq_1(1) + msvq_2(64),$$
$$\cdots, msvq_1(2) + msvq_2(1), \cdots, msvq_1(128) + msvq_2(64)\}.$$

$msvq_i(k)$ are MSVQ frequency vectors of length 10, belonging to i^{th} stage and k^{th} index, as used before in Equation 4.14. To omit the unused vectors and make the vectors speaker-specific, we have reduced the vector size from 8192 to 256. This also results in better probability estimates. The method for reduction is a simple one: The most frequently used 256 vectors from 8192 2-stage combinations are selected for each speaker. This covers approximately 40% of all the frames for both speakers. The rest of the 8192 vectors are mapped to the nearest vector among the 256 selected ones, using the spectral distance measure given in Equation 4.10. Note that, in contrast to the previous mutual information computation, the LSF vectors compared for the two speakers are

Table 4.3: Mutual Information (MI) between the reduced $1^{st} + 2^{nd}$ stages of the source and the target speakers. MP-MI stands for maximum possible MI.

	MI in bits	MP-MI	MI/MP-MI
$I(X_{1,2}; Y_{1,2})$	2.23	8	0.28
$I(X_{1,2}; Y_{1,2})$ (voiced frames)	2.37	8	0.30
$I(X_{1,2}; Y_{1,2})$ (unvoiced frames)	2.29	8	0.29

different, because the selected 256 LSF vectors are speaker-specific. Let us call those selected 256 2-stage vectors of source and target as $X_{1,2}$ and $Y_{1,2}$ respectively. Mutual information computed between the random variables $X_{1,2}$ and $Y_{1,2}$, is given in Table 4.3. The increase in the mutual information compared to the results in Table 4.2 indicates that it is easier to predict the LSF space of the target speaker from the LSF space of the source speaker when more details in the spectrum are considered with speaker-specific quantization.

4.4.2 Obtaining the Mapping Histograms

The observations in the previous section are important, because the method we have used for VT in this chapter is based on them. The observations can be summarized as follows:

- Every stage of MSVQ has more effect on the LPC spectrum than the stages following it in terms of spectral distortion.
- To be able to use the MSVQ of MELP for spectral mapping, the codebook size should be reduced due to corpus size limitations.
- Reducing the codebook size considering the mostly used LSF quantization values by the speakers during reduction is advantageous in terms of speaker individuality.
- Reducing the codebook size carefully increases the mutual information between the LSF spaces of the two speakers, which will result in a better mapping.

The method we have used to obtain a speaker specific quantization out of MELP's MSVQ quantization can be summarized in the following steps:

- The first two stages provide 8192 different LSF vectors. 1600 of them, specific to a speaker, are enough to cover 80% of the LPC spectrum space of each speaker. The mostly used 1600 two-stage combinations for each speaker are obtained.

- New 3^{rd} and 4^{th} stage indices for the whole corpus are determined once more, forcing MELP to use those 1600 1^{st} and 2^{nd} stage combinations (i.e. $X_{N\times 4}^{MSVQ}$ and $Y_{N\times 4}^{MSVQ}$ are updated).

- Considering the 3^{rd} stage with those 1600 makes $1600\times 64 = 102400$ LSF combinations. 102400 LSF vectors have been reduced to L by choosing the most frequently used L vectors from X^{MSVQ} and Y^{MSVQ}. The rest of the LSF combinations are mapped to one of those L vectors using the distance criterion given in Equation 4.10. The procedure has been experimented with L values 256, 128, 96, and 64.

We have obtained a reduced set of L LSF vectors out of MELP's 3 stages of MSVQ. Contribution of the 4^{th} stage has been neglected during training. During transformation, instead of replacing the 4^{th} stage with zeros, we are using the 4^{th} stage frequencies of the source speaker, without transforming.

From the updated source and target data, X^{MSVQ} and Y^{MSVQ}, whose indices include L speaker specific LSF indices; a $L\times L$ matrix which includes occurrence numbers in the 30,000-frame corpus has been obtained. The elements of the histogram matrix, $Hist(i, j)$, show how many times the LSF vector corresponding to the i^{th} index of the source encountered to the LSF vector corresponding to the j^{th} index of the target speaker.

4.5 Transformation - the Baseline System

Transformation of the LPC spectrum is achieved using the histogram matrix. The method for the baseline system is mapping the LSF vector corresponding

to the i^{th} index of the source to j_m,

$$j_m = \arg_j \max Hist(i, j) \tag{4.16}$$

which is the target index, that corresponds to the most frequently occurring index when the source index i occurs in the corpus. In the next section, we will consider some innovations on this system to overcome some shortcomings of it. The main shortcoming of this mapping method is that the parameter space of the converted envelope is limited to a discrete set of envelopes. This reduces the quality of the transformation. This method may also result in high distortions between the LPC spectrums of the neighboring frames, which causes audible buzzy sounds or clicks. In Figure 4.12 spectrograms of the output of the baseline system and the original target speaker's utterance are given. Spectral discontinuities are observed in the converted speech. Also, limitation of the target spectrum has caused unvoiced frames to occur in the middle of the illustrated part of the converted speech. In order to obtain a higher quality synthetic speech, we have applied a dynamic programming approach to determine the best target index that corresponds to the source index. This approach aims to reduce the distortion between the neighboring frames, while it is giving chance to every target index, j, to be used depending on its occurrence rate corresponding to the i^{th} index of the source speaker in the histogram matrix.

4.5.1 Dynamic Programming for LSF Transformation

Assume we have the source speaker's data of one sentence, $X^{sen}_{4\times M}$, where M is the number of frames in the sentence. The columns of X^{sen} denote the 4 stage MSVQ indices. Let the codebook of L codewords (described in Section 4.4.2) of the source be denoted by $\bar{x} = [x_1, x_2, \cdots, x_L]$, and those of target $\bar{y} = [y_1, y_2, \cdots, y_L]$. The first step is the preparation of the source data by quantizing it into the codewords $\bar{x}$ as described in Section 4.4.2. Indices obtained for each LSF vector with respect to new quantization are indicated by

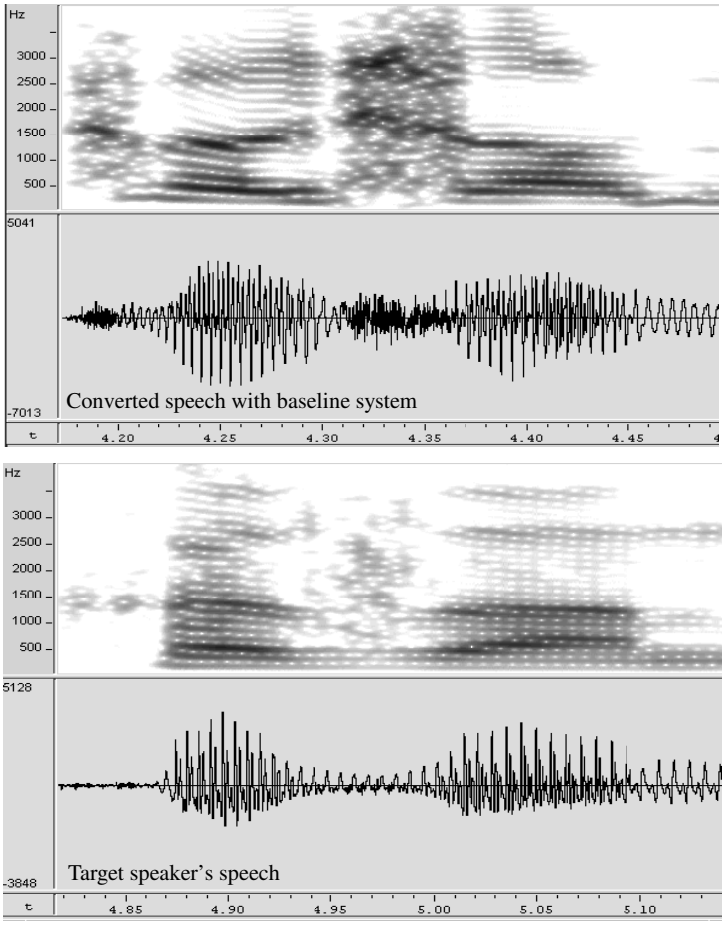


Figure 4.12: Comparison of the converted speech output of the baseline system and the target speaker's speech. Illustrated part is the word *kazandı* in Turkish, sampling rate is 8 kHz.

$x[n]$, whose elements are one of $\bar{x}$. n is the frame number in the sentence, which runs from 1 to M. Using $x[n]$, a sentence dependent histogram matrix is obtained as:

$$H^{sen} = \begin{bmatrix} Hist(x[1],1) & Hist(x[2],1) & \cdots & Hist(x[M],1) \\ Hist(x[1],2) & Hist(x[2],2) & \cdots & Hist(x[M],2) \\ \vdots & \vdots & \ddots & \vdots \\ Hist(x[1],L) & Hist(x[2],L) & \cdots & Hist(x[M],L) \end{bmatrix}_{L \times M} \tag{4.17}$$

where M is the frame size and L is the reduced vector size. Dynamic programming is achieved on the elements of this matrix to determine the best path from frame number 1 to frame number M. We also apply constraints and

transition probabilities obtained from the corpus of target speaker, Y^{MSVQ}, between frames to prevent spectral discontinuities in the transformed speech. This will also allow the chance of using all elements of the matrix in Equation 4.17, which will increase the spectral variability in the output causing more natural sounding transformed speech. A simple flow chart of the dynamic programming procedure is given in Figure 4.13.

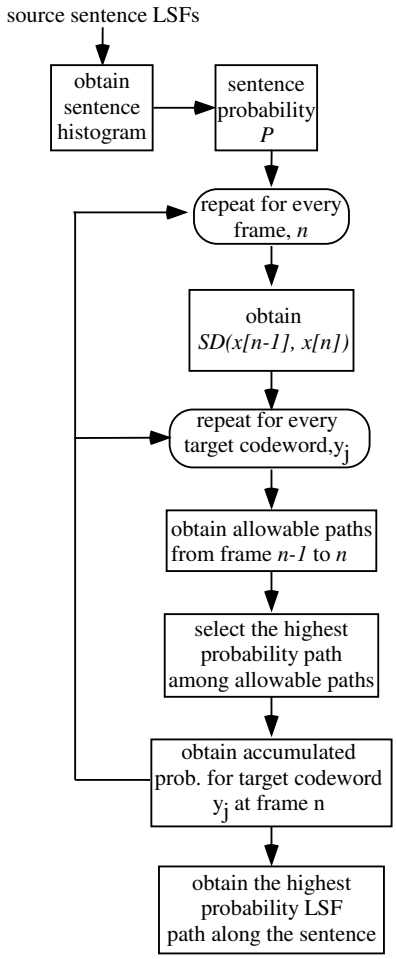


Figure 4.13: Flow chart of the dynamic programming procedure.

To obtain the probability matrix for the M-frame sentence, we normalize the histogram matrix, H^{sen}, in Equation 4.17 such that its columns add up to unity. This new matrix is called P and it is of size $L \times M$. Every row of this matrix corresponds to one of L LSF vectors of the target speaker as illustrated in Figure 4.14. The idea is to determine the best path from frame-1 to frame-M, while allowing only certain transitions among the possible target codewords,

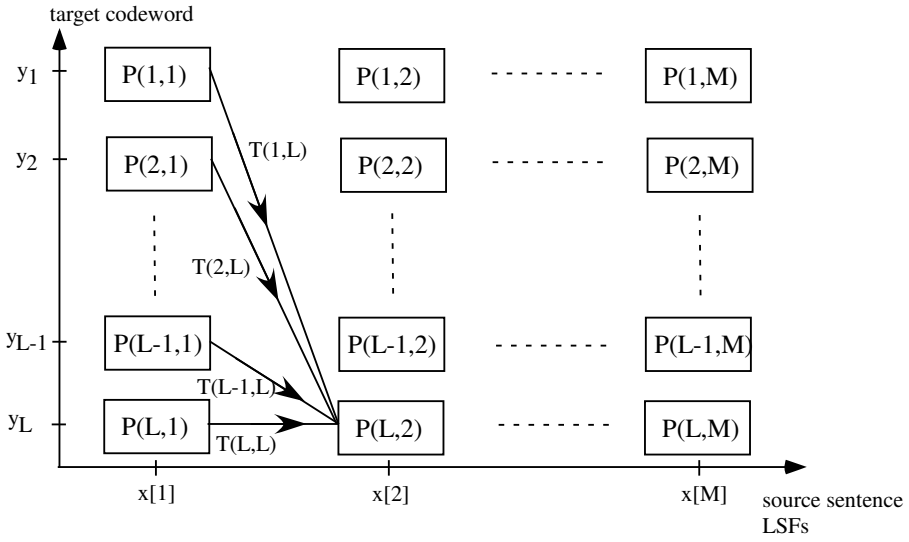


Figure 4.14: Dynamic programming along the frames of one sentence for LSF transformation.

y_i. $T(i,j)$ in the figure represents the transition probabilities from the target codeword y_i in one frame, to another target codeword y_j in the next frame. $T(i,j)$'s are the estimated probabilities based on the empirical probabilities obtained from the target data and T is an $L \times L$ matrix. To determine the path towards $P(L,2)$, for example, first the spectral distortion between the source vectors, $SD(x[1],x[2])$, is computed. SD is the distortion measure d given in Equation 4.8. Paths which satisfy the criterion given in (4.18) are selected as *allowable paths* to $P(L,2)$. This is a search on possible j values ($j = 1 \cdots L$) with an allowed distance interval, D, such that:

$$SD(x[1],x[2]) - D \leq SD(y_j, y_L) \leq SD(x[1],x[2]) + D \qquad (4.18)$$

is satisfied. Best path to $P(L,2)$ is than selected among the allowable paths. It is the path which results in the maximum probability along the path. Let us represent the set of j values satisfying (4.18) with ψ. The path probability is computed by multiplying the accumulated probability on the path towards $P(L,2)$ with the transition probability $T(j,L)$ and $P(L,2)$ itself for all $j \in \psi$. For example, the probability of the path towards y_L at $P(L,2)$ which we define as path probability, $PathProb(L,2)$, is:

$$
\begin{aligned}
PathProb(L,2) &= \max_{j}\{PathProb(j,1) \times T(j,L) \times P(L,2)\}, j \in \psi \\
&= \max_{j}\{P(j,1) \times T(j,L) \times P(L,2)\}, j \in \psi. \qquad (4.19)
\end{aligned}
$$

Equality of $P(j,1)$ to $PathProb(j,1)$ in Equation 4.19 is a special case, because of the initialization of the path-probability matrix, $PathProb_{L\times M}$. Once this algorithm is applied to all frames and $PathProb_{L\times M}$ matrix is obtained with the corresponding path matrix, $Path_{L\times M}$, the best path is determined. Best path is the path corresponding to the highest probability obtained at the final column of $PathProb$ matrix. The complete pseudo code is given below.

Initialize

$$
Path = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L & 0 & \cdots & 0 \end{bmatrix} \quad PathProb = \begin{bmatrix} P(1,1) & 0 & \cdots & 0 \\ P(2,1) & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ P(L,1) & 0 & \cdots & 0 \end{bmatrix}
$$

Begin

```
for k = 2 : M;
    SDs = SD(x[k],x[k-1]);
    for i = 1 : L;
        for j = 1 : L;
            Path_allow  = all j s.t.
                SDs-D <= SD(y[i],y[j]) <= SDs+D;
        endfor
        maxSD = 0;
        for n = 1 : length(Path_allow);
            temp =
            PathProb(Path_allow(n),k-1)P(i,k)T(Path_allow(n),i);
            if temp > max_SD;
                maxSD = temp;
                Path(i,k) = Path_allow(n);
```

```
                PathProb(i,k) = temp;
            endif
        endfor
    endfor
endfor
```

Here a small scale example is provided to clarify the decoding of the path-probability matrix, $PathProb_{L\times M}$, and the path matrix, $Path_{L\times M}$. Assume we have the following path and path-probability matrices with 3 frames ($M = 3$) and 4 codewords ($L = 4$) as:

$$Path = \begin{bmatrix} 1 & \rightarrow & 1 & & 2 \\ & & & \nearrow & \\ 2 & & 3 & \rightarrow & 2 \\ & \nearrow & & & \\ 3 & & 4 & \rightarrow & 3 \\ & \nearrow & & \searrow & \\ 4 & \rightarrow & 4 & & 3 \end{bmatrix} \quad PathProb = \begin{bmatrix} 0.3 & 0.06 & 0.001 \\ & & \\ 0.1 & 0.02 & 0.002 \\ & & \\ 0.5 & 0.03 & 0.006 \\ & & \\ 0.1 & 0.01 & 0.003 \end{bmatrix}$$

The final column of the $PathProb$ matrix shows the accumulated probabilities along the paths and the highest one, which appears in the 3^{rd} row of 3^{rd} column, shows that the last target vector of the best path is the 3^{rd} codeword. Elements in the $PathProb$ matrix show from which row in the previous column, that point is reached. Following the path numbers in the $Path$ matrix, the best path is determined as [4 3 3]. The other available paths are [3 2 1], [3 2 2], and [4 3 4].

Once the path is determined, corresponding target codewords, y_i, are used to determine estimated target codeword sequence, $y_{est}[n]$, for the sentence. The 4^{th} stage source frequencies which are neglected during the transformation are added to y_i's (which are results of the dynamic programming operation) directly. This has the effect of moving the target LSF vector from the codeword

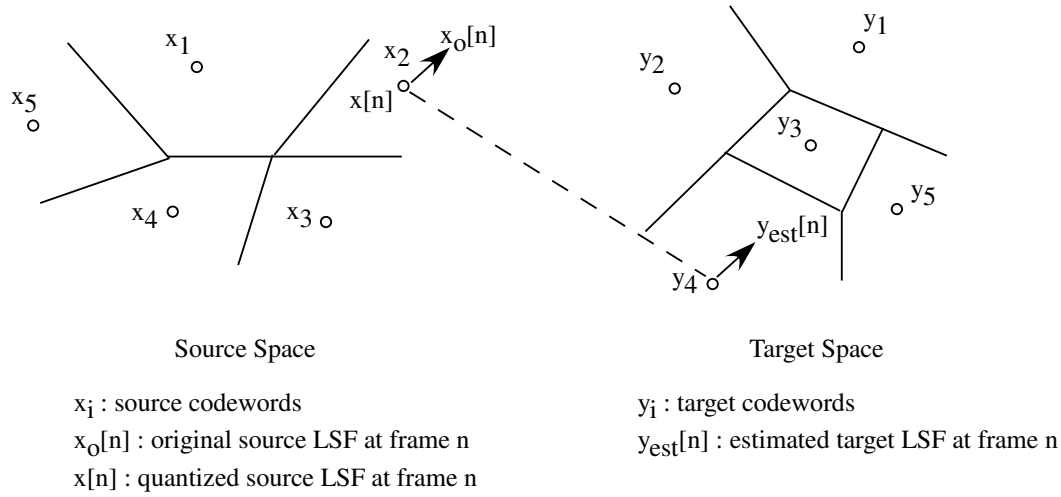


Figure 4.15: Illustration of the transformation method in 2-dimensional space.

vector, y_i, in the same direction with the vector $(x_o[n] - x[n])$ where $x_o[n]$ is the original source LSF from addition of all 4 stages of MSVQ. The situation is illustrated in Figure 4.15 with an example in the two dimensional space. In the figure the source LSF $x_o[n]$ is quantized to x_2. The transformation maps it to the target codeword y_4. The difference vector in the source space is added to the mapped codeword in the target space to obtain $y_{est}[n]$.

A comparison of the baseline system output and the improved system output with dynamic programming is given Figure 4.16. Dynamic programming reduces the discontinuities at the frame boundaries as observed in the figure.

4.5.2 Speech Synthesis

Synthesis of the transformed speech is achieved by applying the modified LSFs in the MELP speech synthesis framework presented in Section 4.2.2. Residual signal of the source is not modified except for a pitch period modification.

The maximum and the minimum pitch period samples in the VT corpus has been obtained for both of the speakers and a linear relationship has been determined between the two pitch ranges. The first speaker's pitch samples range from 38 to 83, while the second speaker's pitch sample range is from 30 to 67 for the sampling rate of 8 kHz. The relationship can be given as:

$$p_1[n] = 0.82p_2[n] - 1.24, \tag{4.20}$$

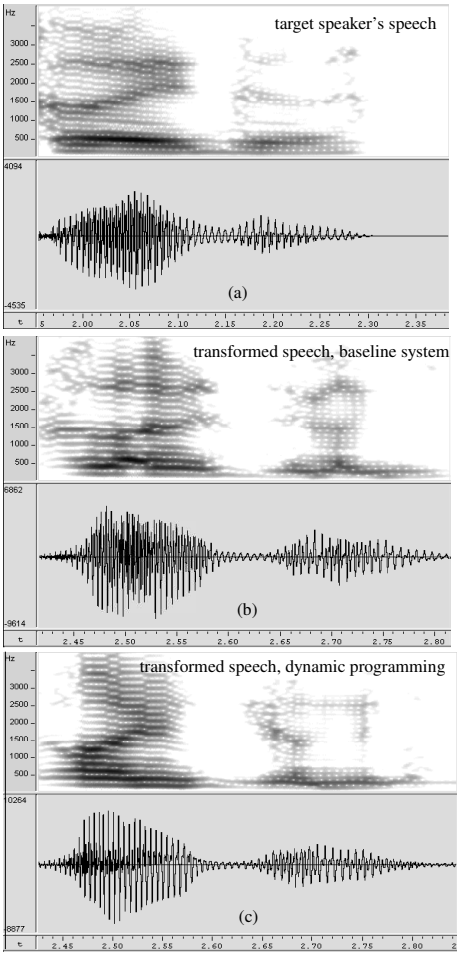


Figure 4.16: Comparison of the waveforms and spectrograms of the baseline system and the improved system with dynamic programming. (a) is target speaker's speech, (b) is conversion with baseline system with $L = 128$, (c) is conversion with dynamic programming with $L = 128$ and $D = 0.16$. Displayed is the Turkish word "aydı".

where $p_1[n]$ and $p_2[n]$ are pitch values at frame number n of the first and the second speakers respectively.

4.6 Results

We used speech data from the speech corpus introduced in Section 4.3. 5 of the 235 phonetically balanced sentences have been selected as the test set. The performance of the baseline VT system discussed in Section 4.5 has been compared to the method proposed in Section 4.5.1 with different parameter values using objective evaluations. Subjective listening tests have also been

achieved and results are reported.

4.6.1 Objective Evaluations

4.6.1.1 Errors and Performance Indices

Three kinds of distances, or errors, are of interest in a VT system: the *transformation* error $E(\hat{t}[n], t[n])$, the *inter-speaker* error $E(s[n], t[n])$, and the *intra-speaker* error $E(t_2[n], t[n])$, where $t[n]$ represents the target speaker's speech, $s[n]$ is the source speaker's speech, $\hat{t}[n]$ is the transformed speech, and $t_2[n]$ is a second rendition of the target speaker's utterance. The inter-speaker error describes the degree of difference between the source and the target speakers, while the intra-speaker error gives a measure of how much variability is present from one rendition to the next of the same sentence. All three errors are conceptual and cannot be measured directly, but can be approximated using objective and subjective evaluations [6].

To determine the transformation performance objectively, we have used the distortion measure given in Equation 4.8. Mean of this metric is obtained over all test set frames. It is given as:

$$E(x, y) = \frac{1}{M} \sum_{m=1}^{M} \sqrt{\frac{1}{10} d^2(L_x^m, L_y^m)}, \tag{4.21}$$

where M is the number of frames in the test set, L^m is the LSF vector component in frame m. Then the LSF transformation performance index is defined as:

$$P_{LSF} = 1 - \frac{E(\hat{t}[n], t[n])}{E(s[n], t[n])}. \tag{4.22}$$

This index applies a normalization to the transformation error $E(\hat{t}[n], t[n])$ across different speaker combinations. P_{LSF} is zero when the transformation error equals the inter-speaker error $E(s[n], t[n])$. It approaches to 1 as the transformation error approaches to zero. In practice, $E(\hat{t}[n], t[n])$ is never zero, because the intra-speaker error $E(t_2[n], t[n])$ is usually not zero. Actually

$E(t_2[n], t[n])$ is the lower bound for an achievable $E(\hat{t}[n], t[n])$. However, we do not have the chance to measure this lower bound, because our corpus includes only one utterance of the sentence-set from each speaker. In an effective VT system transformation error is expected to be below the inter-speaker error, which means $P_{LSF} > 0$.

4.6.1.2 Results

There exist two training parameters in our system: the number of reduced codewords, L, and the distance interval, D. We have evaluated the performance indices of the system for $L = 256, 128, 96, 64$, and
$D = 0.14, 0.16, 0.18, 0.20, 0.4, 0.6, \infty$. $D = \infty$ means all paths are allowed in the dynamic programming procedure. Transformations from Speaker-1 to Speaker-2 are tested. Figure 4.17 presents the results obtained. The maximum possible value for D is approximately 2.3 dB for each L value, which is the maximum value of the spectral distance between any LSF codewords of the source. Distance values of $D > 0.6$ give results which are very close to applying no constraints at all and values of $D < 0.14$ give no solutions, since most of the frame boundaries end up with no allowable paths when constraints are too tight.

LSF performance indices of the system with dynamic programming have been compared to the performance indices of the baseline system as given in Figure 4.17. The performance index of the baseline system has been found to be 0.010, 0.011, and 0.011 for $L = 256, 128, 64$, respectively. Dynamic programming improves the transformation performance as presented in the figure.

The results for $L = 64$ are not given in the figure, because this case gives performance indices below zero, which means that VT system is not successful. The reason for the decrease in the performance when L is reduced to 64 is thought to be the selection procedure of the reduced codewords for the speakers. When the number of codewords are reduced to a very small value,

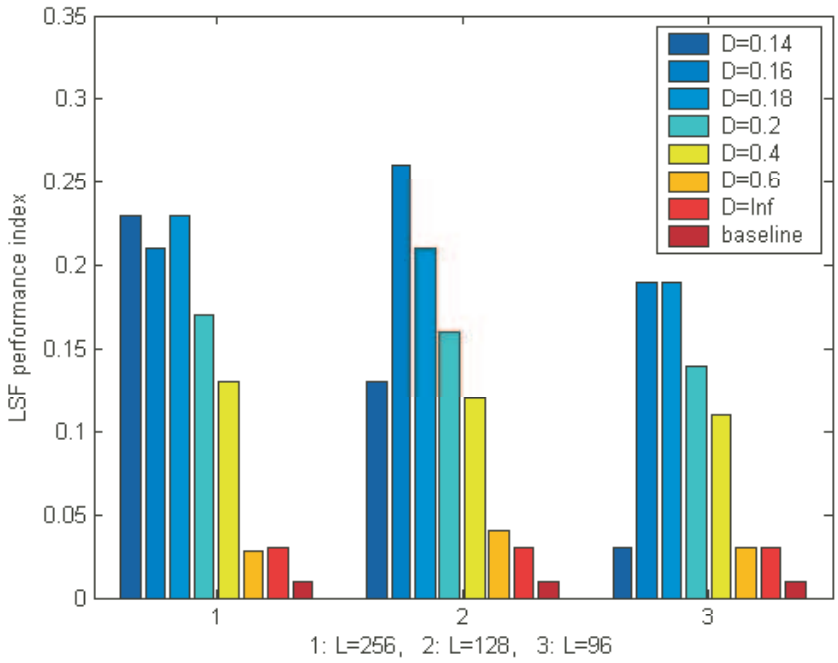


Figure 4.17: Performance indices of the system for transformation from Speaker-1 to Speaker-2.

the reduced codebook includes codewords which are very close to each other. The reduction criterion is selecting the most frequently used codewords which is not the ideal method to cover the LSF space of a speaker, when a small-sized codebook is of concern. The ideal method to select the optimum LSF codewords out of MELP's codebook, specific to a speaker, would be running a second quantization algorithm on the codewords of MELP, such as *k-means* algorithm. Chapter 5 presents our work on LSF quantization to avoid this problem, which quantizes the LSF space of the speakers directly, instead of using the pre-determined codebook of MELP. The improvements in the performances are presented in Chapter 5.

It has been observed that, best performance index is obtained for the codebook size, $L = 128$, and form the allowable distance interval in the dynamic programming, $D = 0.16$. $L = 128$ is probably a trade-off between obtaining correct probabilities from the histogram matrix, and obtaining a reduced codebook which represents the LSF space of the speakers efficiently. When L

is high, the probabilities obtained from the histogram matrix are less reliable, but the LSF space representation of the reduced codebook is more efficient.

4.6.2 Subjective Evaluations

The subjective evaluations consist of a speaker-similarity test. The speaker-similarity test is an **ABX** test.

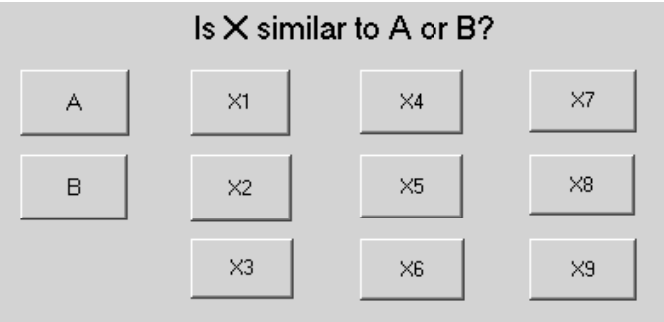


Figure 4.18: Interactive window of the speaker-similarity test.

During the **ABX** text, **A** and **B** represent the original speakers, and **X** is the transformed speech either from **A** to **B** or from **B** to **A**. Subjects are asked to determine whether **X** is more similar to **A** or **B**. The interactive window of the speaker-similarity test is shown in Figure 4.18. Only one original sentence of the speakers are provided to the subjects instead of providing the original forms of all the transformed sentences. This is to prevent the speaker-specific long-term behavior of the intonation along the sentences, from effecting the decision of the subjects. The test includes 9 transformed **X** sentences, 3 transformations for each different L values ($L = 256, 128, 96$). D is 0.18, 0.16, and 0.16 for L values 256, 128, and 96, respectively, which give the highest performance indices as observed in Figure 4.17. 20 subjects have taken the test. 174 converted sentences have been detected as the target speaker out of 180 sentences in total. 4 of the incorrect decisions were for $L = 128$ and 2 of them were for $L = 256$.

CHAPTER 5

LSF QUANTIZATION FOR VOICE TRANSFORMATION

5.1 Introduction

This chapter introduces the method we propose for quantizing LSFs to map the LSF space of one speaker to that of the other. In Chapter 4, we have investigated the usage of MELP's MSVQ codebook of LSFs for our VT system. In this chapter, instead of using MELP's LSF codebook, a new codebook for each speaker is obtained. MELP analysis and synthesis system is used as a framework, except for the LSF quantization part.

5.2 Quantizing Line Spectral Frequencies for VT

Quantization of the LSFs have been achieved by *k-means quantization algorithm* [67] after a *principle component analysis* based dimension reduction [68] applied on LSFs. Background on principle component analysis (PCA) and k-means clustering are given in Appendices D and E respectively. Application of the method will be presented in this section.

In principle component analysis (PCA), a set of data is summarized as a linear combination of an orthonormal set of vectors. For the data matrix $X_{n\times d}$, whose rows are data vectors x_i, $i = 1, \cdots, n$, the orthonormal set of

vectors are the eigenvectors of the covariance matrix of the data, X. The linear transformation T, whose columns are the eigenvectors of the covariance matrix, is used to project the normalized (zero-mean) data on the orthonormal eigenvectors. In PCA, the eigenvalues are arranged in descending order, such that the transformed zero-mean data $(X-\bar{x})T$ (where $\bar{x}$ is the mean of the data X) has the highest variance along the first eigenvector. Therefore, the data can be reconstructed using the projection of data on m eigenvectors ($m < d$) only, without sacrificing the main characteristics of it. Moreover, this transformation provides a more diverse distribution of data on the eigenvectors, which make the data more appropriate for quantization.

PCA is most appropriate for approximating multivariate normal distributions, or more generally elliptically symmetric distributions, which is almost the case in our LSF distributions. Example distributions are given in Figures 5.1 and 5.2.

In this thesis, PCA has been used to determine the principle components of the source and target LSFs to obtain a more efficient quantization of them. Only the dimensions with high variance have been quantized and those dimensions have been used to find the mapping between the two speakers.

5.2.1 Speaker-specific LSF Quantization

LPC analysis has been applied to the same 180-sample non-overlapping speech frames presented in Chapter 4. A 10^{th} order LPC analysis is performed on both the source and the target speech signal using a 200-sample Hamming window centered on the last sample in the current frame. This procedure is the same as applied in MELP analysis. LSFs have been obtained from LPC coefficients for each frame. Source and target feature stream lengths have been equated using the same DTW procedure as presented in Section 4.3. Band-pass voicing analysis of MELP has been used to label each frame as voiced or unvoiced. LSF quantization and transformation have been applied to the voiced and unvoiced frames of the speakers separately. Since DTW is

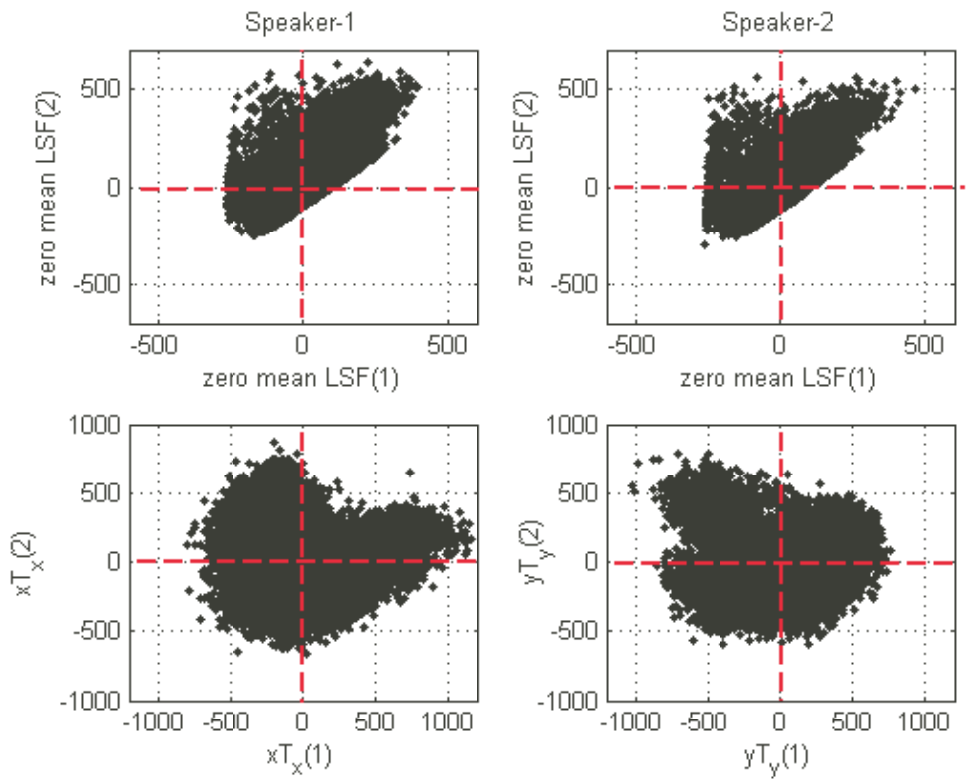


Figure 5.1: Plots of the first two components of the normalized (zero-mean) LSFs (upper plots) and the first two components of the transformed data (lower plots) for both speakers. LSF values are given in Hz for a sampling rate of 8 kHz. The upper figures are the mean-subtracted LSFs for voiced frames, while the lower figures are the first two components after the transformation. xT_x stands for the transformed vectors for Speaker-1, and yT_y stands for the transformed vectors for Speaker-2.

applied on a spectrum similarity basis for automatic alignments of the speech segments, some unvoiced frames of the source might be matched to voiced frames of the target or vice versa. These wrong-matched frames are eliminated before the training. The main reason for elimination of these frames is that during synthesis the residual of the source speaker that is not matched to the transformed filter causes degradation in speech quality. Those new LSF data sets are defined as X^V and X^{UV} for the source, and Y^V and Y^{UV} for the target. Superscripts V and UV stand for voiced and unvoiced data streams respectively. X^V and Y^V are ($N_V \times 10$) matrices with LSFs on the rows, where N_V represents the number of voiced frames. Similarly, X^{UV} and Y^{UV} are

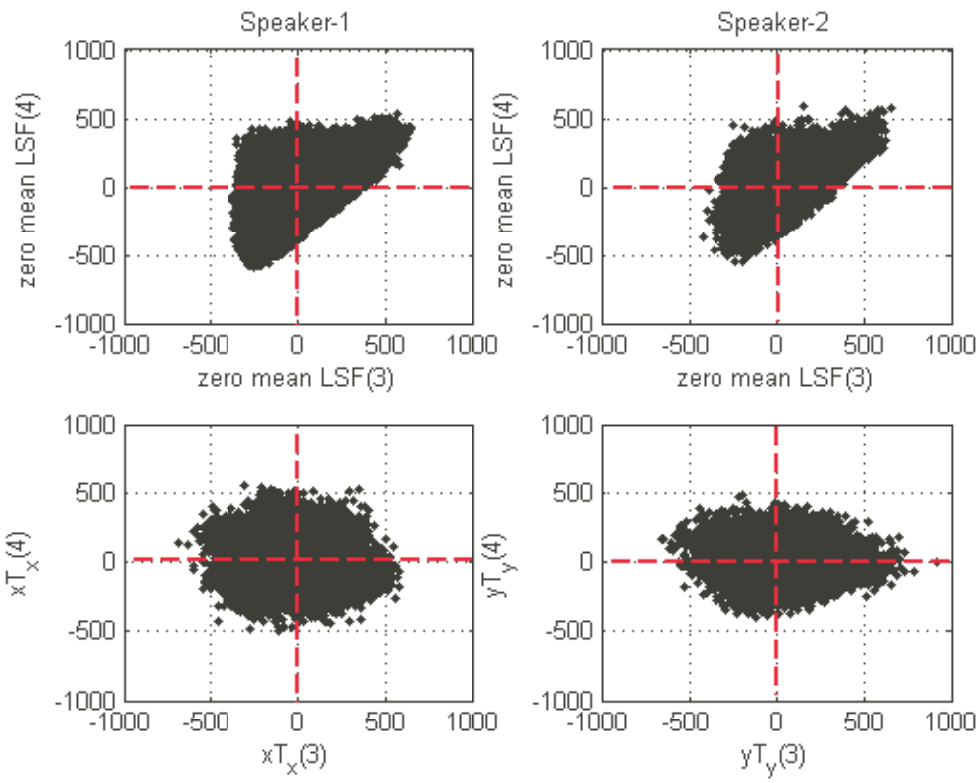


Figure 5.2: Plots of the third and fourth components of the normalized (zero-mean) LSFs (upper plots) and the third and fourth components of the transformed data (lower plots) for both speakers. LSF values are given in Hz for a sampling rate of 8 kHz. The upper figures are the mean-subtracted LSFs for voiced frames, while the lower figures are the first two components after the transformation. xT_x stands for the transformed vectors for Speaker-1, and yT_y stands for the transformed vectors for Speaker-2.

matrices of size ($N_{UV}\times 10$). Approximately one tenth of all frames are unvoiced for both speakers.

Note that, for convenience, we will assume that Speaker-1 is the source speaker and his data will be represented by X, and Speaker-2 is the target speaker and his data will be represented by Y in the following sections. During evaluations, both speakers will be used as the source and the target.

5.2.1.1 PCA Application

Principle component analysis has been applied on the four data matrices, X^V, X^{UV}, Y^V, and Y^{UV}. The aim of applying PCA is to obtain the principle

components of the data and obtain projection (or transformation) matrices, T_x^V, T_x^{UV}, T_y^V, and T_y^{UV} for voiced and unvoiced data matrices of the source and target speakers, respectively. The columns of the transformation matrices are the eigenvectors of the covariance matrices obtained from the zero-mean data as explained in Appendix D.

Let us represent mean vectors of the data matrices, X^V, X^{UV}, Y^V, and Y^{UV}, along their columns with $\bar{x}^V$, $\bar{x}^{UV}$, $\bar{y}^V$, and $\bar{y}^{UV}$, respectively. Then the zero-mean and the transformed data matrices, $\tilde{X}^V$, $\tilde{X}^{UV}$, $\tilde{Y}^V$, and $\tilde{Y}^{UV}$ are represented by:

$$\tilde{X}^V = \left(X^V - \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{N_V} \bar{x}^V \right) T_x^V, \; \tilde{Y}^V = \left(Y^V - \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{N_V} \bar{y}^V \right) T_y^V, \qquad (5.1)$$

for voiced data and similarly,

$$\tilde{X}^{UV} = \left(X^{UV} - \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{N_{UV}} \bar{x}^{UV} \right) T_x^{UV}, \; \tilde{Y}^{UV} = \left(Y^{UV} - \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{N_{UV}} \bar{y}^{UV} \right) T_y^{UV}, \qquad (5.2)$$

for unvoiced data.

The aim of using PCA here is to quantize the transformed data more efficiently. This is illustrated with examples in Figures 5.1 and 5.2. In Figure 5.1, the first two dimensions of the mean-subtracted LSF values of voiced data (X^V on the left and Y^V on the right) are plotted with respect to each other on the upper panel. They are the zero-mean forms of X^V and Y^V. The lower plots are the first two components after the transformation. Note that the transformation matrices of the two speakers are data dependent and different, and they are denoted by T_x and T_y in the figures. The dashed lines are drawn to illustrate the data distribution in each cell, when a simple four-level quantizer is used to divide the LSF space of the first two dimensions in the figures. Data dimensions will be less correlated and quantization will be much

Table 5.1: Eigenvalues of the covariance matrices obtained from the zero-mean data matrices. Eigenvalues are listed in descending order.

e_x^V	e_x^{UV}	e_y^V	e_y^{UV}
3.48	42.20	3.50	31.97
1.66	13.69	1.72	12.33
1.22	6.98	1.10	8.67
0.87	4.38	0.67	4.17
0.49	3.25	0.55	3.42
0.36	2.19	0.41	2.58
0.35	2.08	0.34	2.22
0.29	1.98	0.27	1.98
0.25	1.73	0.21	1.91
0.10	1.60	0.11	1.31

more efficient (i.e. data diversity in each cell will be more equivalent) after the transformation as illustrated in Figures 5.1 and 5.2.

Another use of PCA is that the number of dimensions to be quantized can be reduced after transformation, which may result in a more efficient quantization. The eigenvalues obtained after PCA, e_x^V, e_y^V, e_x^{UV}, and e_y^{UV}, are listed in Table 5.1. As observed in the table, the first few dimensions are very significant for the data, while the others are less significant. This means that if we use only the first four dimensions of the transformed data for reconstruction, we will be able to capture most of the characteristics of the LSFs. Eigenvectors give us the dimensions which characterize the data. If the corresponding eigenvalue of an eigenvector is high, then the data has a high variance in the dimension of that eigenvector. If the data has a small variation in one of the dimensions, then neglecting that dimension during reconstruction does not result in a big loss in the data characteristics.

5.2.1.2 k-means Clustering Application

k-means clustering algorithm has been applied on the transformed data to obtain a limited set of vectors representing the transformed LSF space of the speakers individually. The details of the algorithm are presented in Appendix E.

Let us define N as the number of principle components to be quantized and L as the number of codewords in k-means algorithm. Determination of the initial codebook, C_1, depends on the relative value of N with respect to L. If $L = 2^N$, for example, all initial codewords can be determined by binary decisions, using the mean values of the positive and negative data values in each dimension as the centroids. Then all possible values of all components make 2^N possible codewords. Note that, determination of the initial codebook is not very critical, but it is useful to start with a good approximation of the data distribution for C_1, so that k-means algorithm converges quickly to an optimal codebook. For the case, $N = 4$, and $L = 64$, for example, first two dimensions with higher eigenvalues are assigned to 4 levels, while the other two dimensions are assigned to 2 levels, giving $4^2 \times 2^2 = 64$ codewords for C_1.

k-means iteration is applied to modify C_1 until the fractional drop in the average distortion becomes equal to or less than $2.22 \cdot 10^{-16}$, and new codebooks C_x^V, C_x^{UV} (for voiced and unvoiced data of the source speaker), and C_y^V, C_y^{UV} (for voiced and unvoiced data of the target speaker) are obtained.

5.2.2 Training

Once the speaker-specific codebooks for voiced and unvoiced data of both speakers are determined, the mapping histogram matrices are obtained separately for voiced and unvoiced data of the source and the target speakers. Zero-mean and transformed data from source and target speakers, $\tilde{X}^V$, $\tilde{X}^{UV}$, $\tilde{Y}^V$, and $\tilde{Y}^{UV}$ are quantized using the codebooks C_x^V, C_x^{UV}, C_y^V, and C_y^{UV}, respectively. Mapping matrices are of size $L \times L$ similar to the mapping histogram matrices presented in Chapter 4. One example mapping histogram obtained after quantization with $N = 4$ and $L = 64$ is presented in Figure 5.3. Note that this mapping histogram is different than the mapping histogram in Figure 4.9, in the sense that there are no empty columns along the source or target dimensions in the histogram. Since, the codewords are speaker-specific, all codewords are used by the speakers. Moreover, the transformation applied

helps to obtain an equal diversity of data in each cell. Note that, mapping is obtained only for the N number of quantized principle components. The remaining components are not quantized. During transformation, they are either back-transformed using the back-transformation of the target speaker without any mapping operation or they are simply not taken into consideration. This will be explained in detail in Section 5.3.

Figure 5.4 illustrates the block diagram of the training mode of the system.

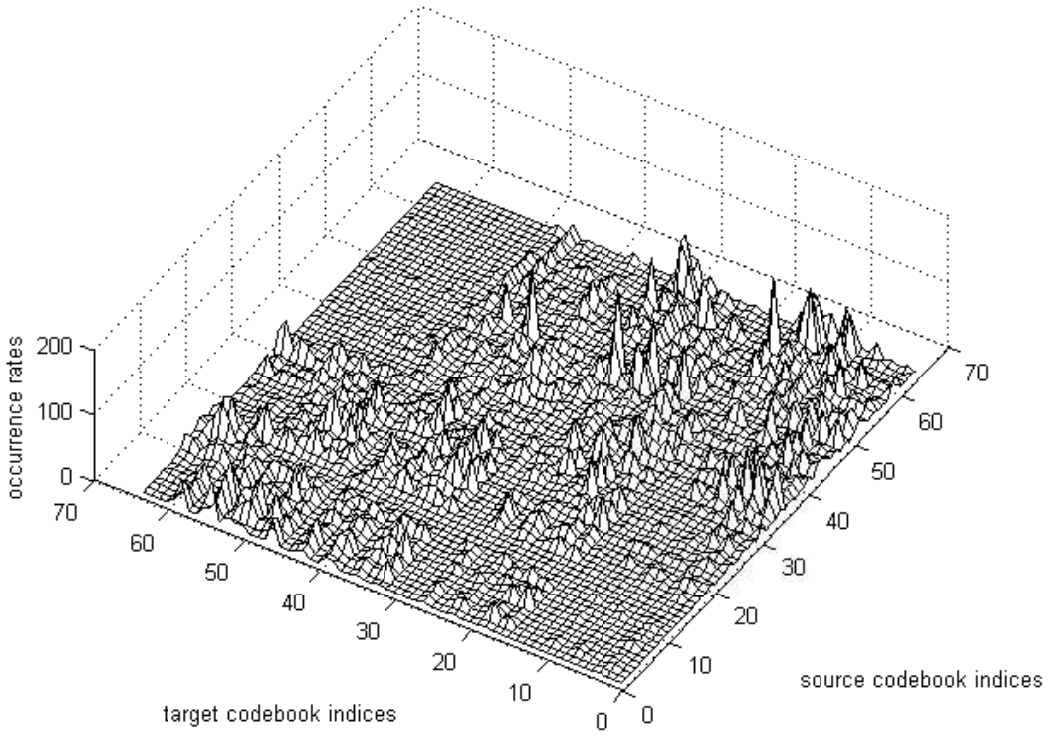


Figure 5.3: Mapping histogram for quantized voiced frames of the source and the target speakers, $\tilde{X}^V$ and $\tilde{Y}^V$.

5.3 Transformation

In the transformation mode, the system makes a voiced-unvoiced decision using the band-pass voicing values of MELP analysis. Then the mean LSF vector ($\bar{x}^V$ or $\bar{x}^{UV}$) is subtracted from the input LSF vector. It is transformed and

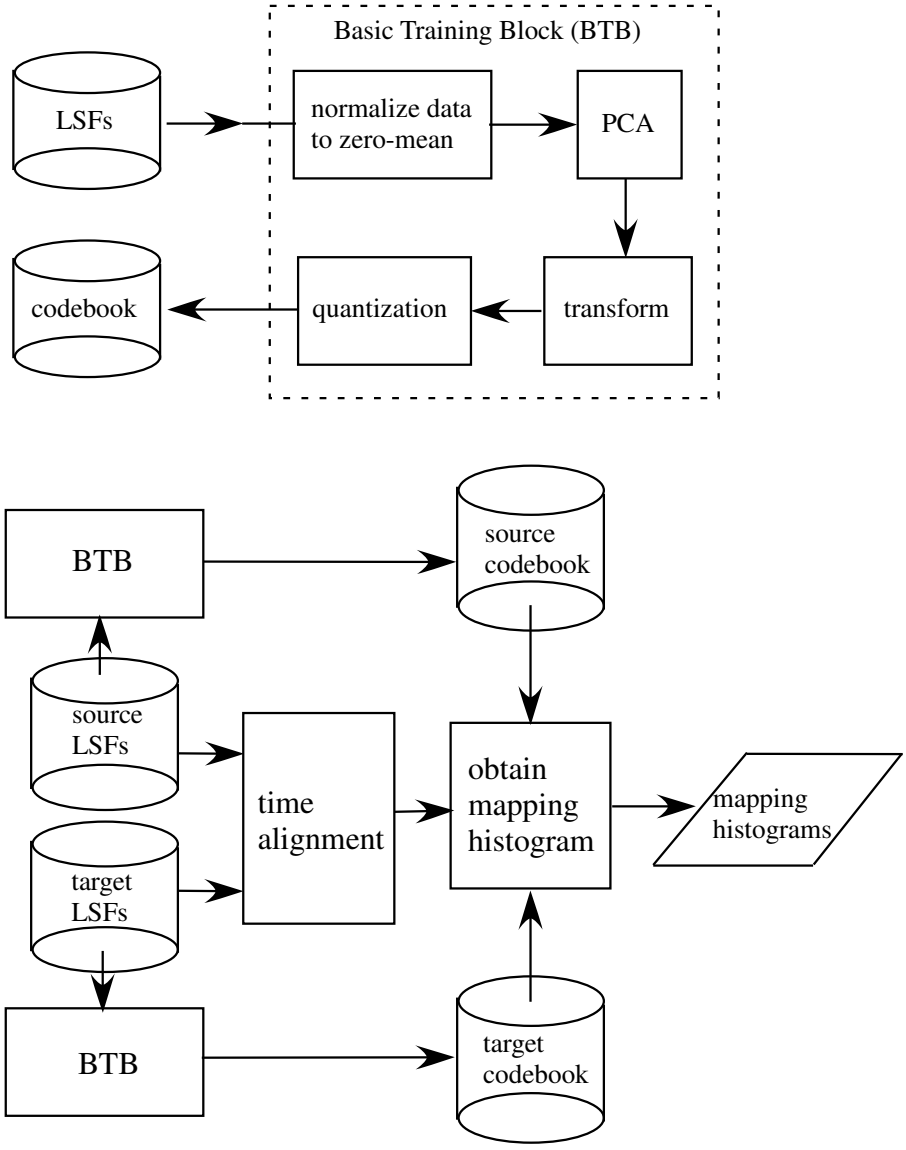


Figure 5.4: Block diagram of the training system.

the selected number of principle components (N can be from 1 to 10, $N = 10$ meaning all dimensions of the transformed matrix are used) are quantized using source's codebook. Note that quantization codebook is different for different N values, since codeword vectors are of size $N \times 1$. Those codebooks are obtained in the training mode. Each quantized frame is mapped to a target codeword based on the mapping histogram matrix. Dynamic programming is applied on a whole sentence to map the frames of the source to those of the target, which has been explained in detail in Section 4.5.1. The allowable paths are obtained by comparing the L_2 norms of the subsequent source codewords in the

sentence and the subsequent target codewords. The obtained target codewords are back-transformed using the inverse of the target speaker's transformation matrix. Mean LSF vector of the target ($\bar{y}^V$ or $\bar{y}^{UV}$) is added to the back-transformed target codeword to obtain the final transformed LSFs.

Now the method will be formulated in more detail. We will drop the superscripts V and UV for convenience from this point on. Let us define the transformation matrix for source and target as:

$$\begin{aligned} T_x &= [T_x^1 \quad T_x^2] \\ T_y &= [T_y^1 \quad T_y^2], \end{aligned} \tag{5.3}$$

where T_x^1 and T_y^1 are $10 \times N$ matrices, which are the first N columns of T_x and T_y. T_x^1 transforms the N principle components of the zero-mean source LSF row vector $(x - \bar{x})$. $N = 10$ means all dimensions are quantized. The transformation operation can be rewritten as:

$$\begin{aligned} (x - \bar{x})T_x &= (x - \bar{x})[T_x^1 \quad T_x^2] \\ &= [(x - \bar{x})T_x^1 \quad (x - \bar{x})T_x^2]. \end{aligned} \tag{5.4}$$

Since the columns of T_x and T_y are unit eigenvectors, $(T_x)^{-1} = (T_x)^T$ and $(T_y)^{-1} = (T_y)^T$, where the superscript T denotes transpose of the matrix. If we want to back-transform the transformed data in Equation 5.4 using all components without any loss of data, the below back-transformation is applied:

$$\left[(x - \bar{x})T_x^1 \quad (x - \bar{x})T_x^2\right] \begin{bmatrix} (T_x^1)^T \\ (T_x^2)^T \end{bmatrix}. \tag{5.5}$$

The expression in (5.5), can be rewritten as:

$$(x - \bar{x}) = [(x - \bar{x})T_x^1(T_x^1)^T] \; + \; [(x - \bar{x})T_x^2(T_x^2)^T]. \tag{5.6}$$

The first part of the summation in Equation 5.6 comes from the principle components, and the second part comes from the remaining components. During transformation, quantization and mapping is applied on the first addend in Equation 5.6 before back-transformation with the inverse of the target

transformation matrix, $(T_y^1)^T$, instead of the source back-transformation with $(T_x^1)^T$. The second addend in Equation 5.6 is either replaced with back transformation of $(x - \bar{x})T_x^2$ with the target back-transformation, $(T_y^2)^T$, without quantization, or it is simply neglected and equated to zero.

Let us denote the quantization operation of the source speaker by $Q_x^N(\cdot)$, which maps the space of the source speaker to an N dimensional finite subset of L components. If the mapping form source codebook to target codebook operation is denoted by $f(\cdot)$, then the target codeword obtained after transformation and mapping is given as:

$$f(Q_x^{N_p}\left((x - \bar{x})T_x^1\right)). \tag{5.7}$$

The vector obtained in (5.7) is a $N\times1$ vector selected from the target codebook, C_y^V or C_y^{UV}, therefore, it should be back-transformed with the inverse of the transformation matrix of the target, T_y. Then the target LSF vector estimation, y_{est}, obtained after transformation is:

$$y_{est} = \left[f(Q_x^{N_p}\left((x - \bar{x})T_x^1\right)\right](T_y^1)^T \ + \ \bar{y}, \tag{5.8}$$

where $\bar{y}$ is the mean vector of the target LSFs. If the unquantized components are also considered, y_{est} is computed as:

$$y_{est} = \left[f(Q_x^{N_p}\left((x - \bar{x})T_x^1\right)\right](T_y^1)^T \ + \ \left[(x - \bar{x})T_x^2(T_y^2)^T\right] \ + \ \bar{y}. \tag{5.9}$$

The estimation in Equation 5.9 back-transforms the non-principle components with the transformation matrix of the target, $(T_y^2)^T$, directly, without any quantization and mapping. Figure 5.5 presents the block diagram of the LSF transformation system.

5.4 Evaluations

5.4.1 Objective Evaluations

The same error measure between the target and the converted speech with the same test set discussed in Section 4.6.1 have been used.

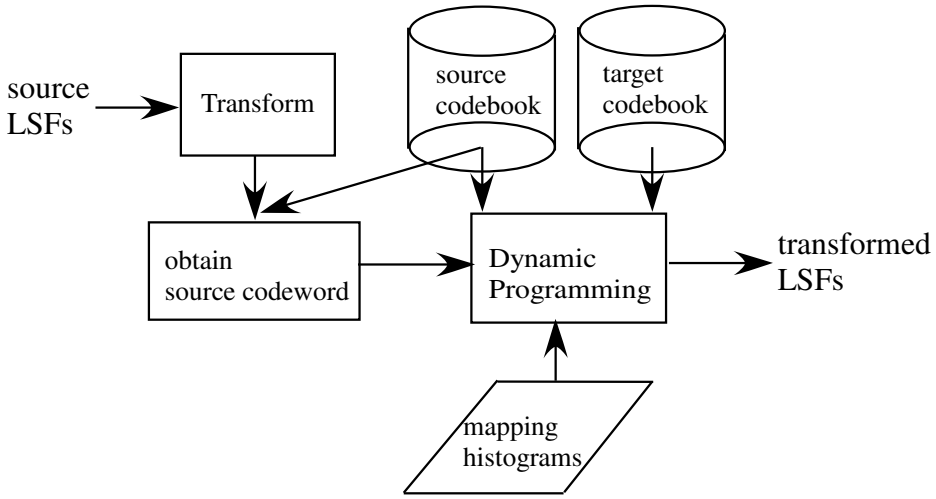


Figure 5.5: Block diagram of the LSF transformation system.

There exist two training parameters in training the VT system described in this chapter: number of principle components to be quantized, N, and the number of codewords, L, which is also the size of the mapping histogram matrix. We varied $N = 4, 6, 8, 10$ for voiced and unvoiced data. We varied $L = 64, 128, 256$ for voiced data and $L = 16$ for unvoiced data. Mapping histograms and codewords have been determined for both speakers. There is one transformation parameter, D, which is the interval to obtain the allowable paths in the dynamic programming algorithm given in Section 4.5.1. Interval, D, has been varied from various values towards ∞, which means all paths are allowed in the dynamic programming procedure. LSF performance indices (given in Section 4.6.1) of the transformation from Speaker-1 to Speaker-2 on the five test sentences are given in Figure 5.6. A more detailed plot of the performance indices with $L = 64$, $N = 4$ and Equation 5.8 is given in Figure 5.7 to illustrate the improvement in the performance index achieved by dynamic programming.

In Figure 5.6, P_{LSF} has been illustrated for the transformation from Speaker-1 to Speaker-2 for the transformations given in Equation 5.8 and Equation 5.9. The $\otimes$ marked plots are for Equation 5.8. P_{LSF} has been computed for $D = 25, 50, 75, 100, 125, 150, 200, 300, 400, 500, 600$. P_{LSF} approaches to the value of no-constraint case for values larger than $D = 500$. Note that D values

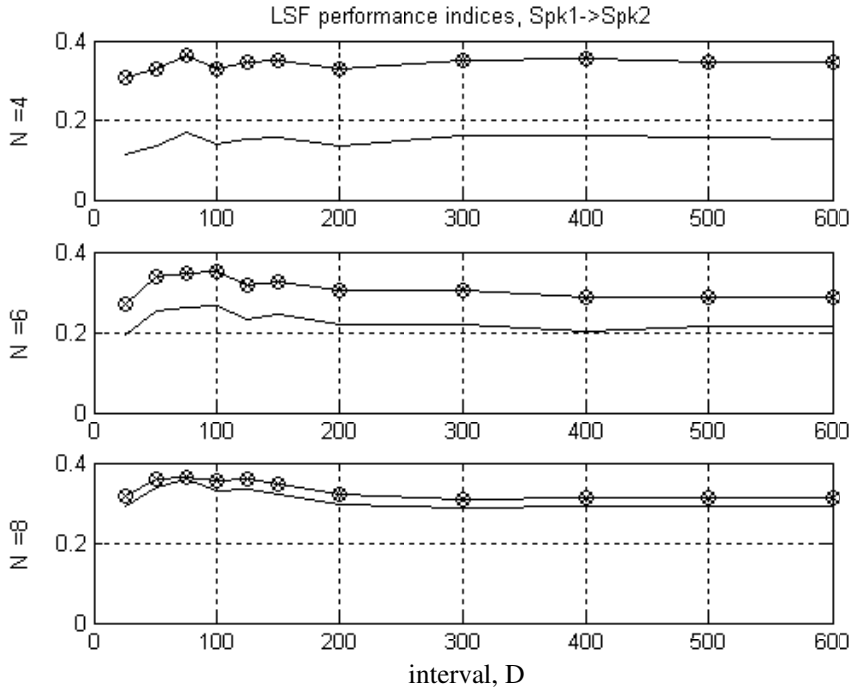


Figure 5.6: LSF performance indices, P_{LSF}, for N values 4, 6, and 8, with respect to the interval, D. The $\otimes$ marked plots are P_{LSF} of the transformation given in Equation 5.8, others are P_{LSF} of the transformation in Equation 5.9.

are quite different than those D values used in Section 4.6.1.2. This is because the distortion measures used in Equation 4.18 for the dynamic programming are different for the MELP-based method and the method proposed here. In the MELP-based method the spectral distortion measure given in Equation 4.8 has been used for LSF values in radians. Distortion measure used in this chapter is the Euclidean distance between the transformed and dimension reduced LSF vectors in Hz.

It has been observed that best results are obtained when D is a value around 75 for all N values. This shows that the dynamic programming approach increases the transformation performance compared to the case when no constraints are used. The maximum performance achieved is almost the same for all N values and they are around $P_{LSF} = 0.36$. The difference between the performances of transformation using Equation 5.8 and Equation 5.9 increases as N decreases. This is expected because as the number of principle components are decreased, more dimensions will be carried on the unquan-

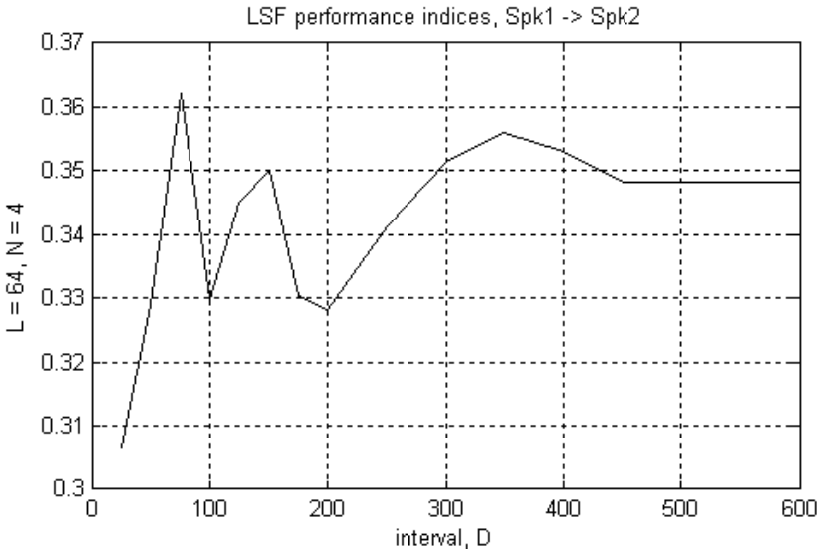


Figure 5.7: Detailed plot of the results given in Figure 5.6, with $L = 64$, $N = 4$ and Equation 5.8.

tized term in Equation 5.9, and the performance index will decrease. It is also important to note that, when Equation 5.8 is used, performance indices are almost the same for all N values. This is because the the first few principle components carry most of the information as observed from the eigenvalues obtained after PCA given in Table 5.1.

Figures 5.8, 5.9, and 5.10 illustrate the LSF performance indices for $L = 64, 96, 128$, respectively. The highest performance indices are obtained with the codebook size of $L = 64$. We have also evaluated the system with $L = 48$; however, negative performance indices have been obtained in that case. A small codebook size is advantageous in terms of approximating the real probabilities in the mapping histogram matrix using a limited database. However, below a certain L value, the quantization size is not enough to model the LSF spaces of the two speakers. The dimension reduction increases the performance index for $L = 64$ and $L = 96$ cases. With $L = 128$, on the other hand, the highest performance index is obtained with $N = 8$.

It is possible to conclude that this system performs better than the MELP-based VT system given in Chapter 4, because the maximum possible performance index for the MELP-based system has been achieved as 0.26 with

$L = 128$, while it is 0.36 with $L = 64$ in the current system. The performance of this system is also comparable with the LSF performance indices given in the literature [6, page 62]. The results given in [6] give highest performance index as approximately 0.33. However, the test sets are completely different in this work and in the one reported in [6]. Therefore, it is not possible to definitely report that this system performs better.

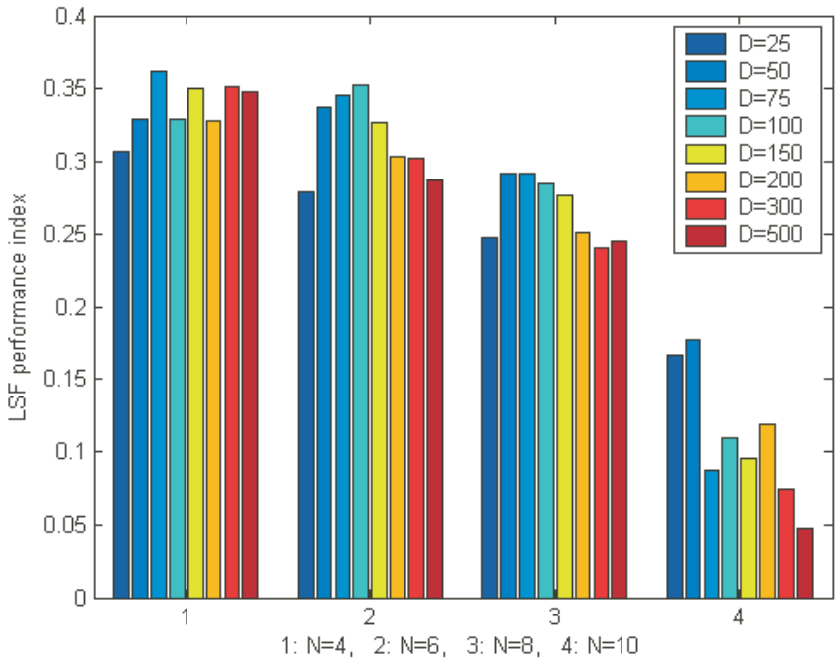


Figure 5.8: LSF performance indices, P_{LSF}, for transformation from Speaker-1 to Speaker-2, $L = 64$ and the transformation given in Equation 5.8

We have also computed the mutual information between the codeword indices of the two speakers for different codebook sizes for both systems using the mapping histograms to obtain empirical probabilities. It has been observed that the mutual information is higher in the new VT system compared to the MELP-based system. Results are given in Table 5.2.

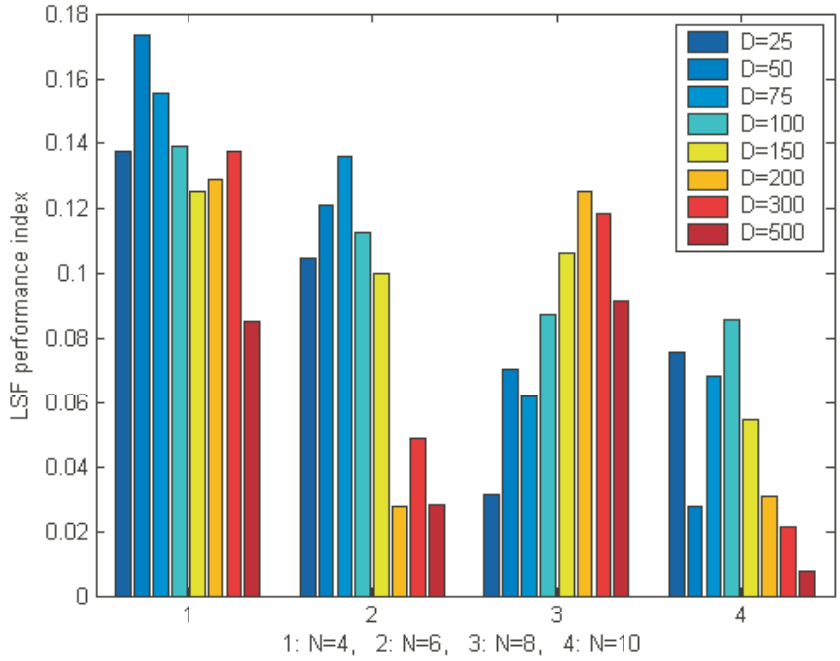


Figure 5.9: LSF performance indices, P_{LSF}, for transformation from Speaker-1 to Speaker-2 with $L = 96$ and the transformation given in Equation 5.8

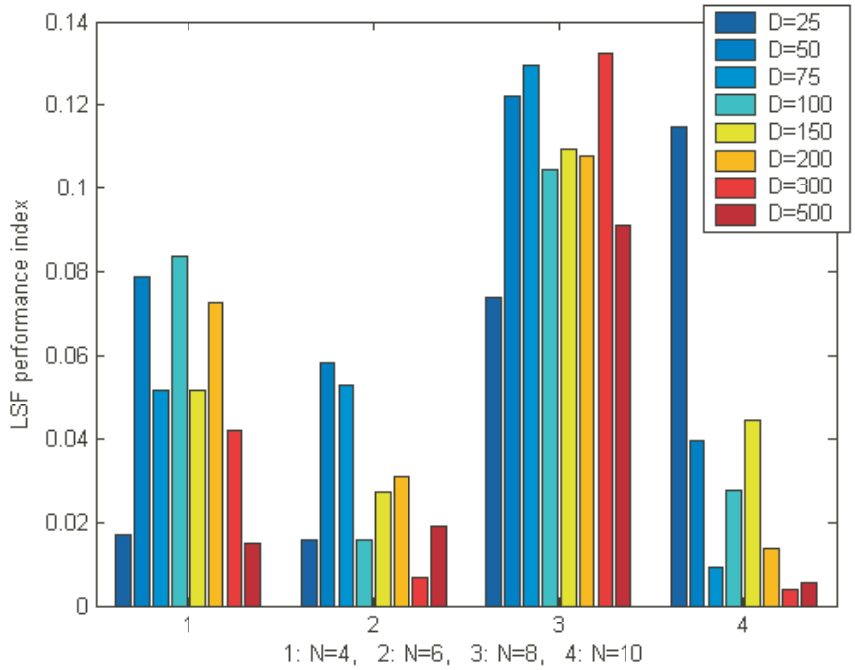


Figure 5.10: LSF performance indices, P_{LSF}, for transformation from Speaker-1 to Speaker-2 with $L = 128$ and the transformation given in Equation 5.8

Table 5.2: Mutual information in bits, obtained from the mapping histograms of the voiced frames of the two speakers.

	New System, $N = 4$	**MELP-based System**
$L = 128$	1.62	1.21
$L = 96$	1.44	1.08
$L = 64$	1.26	0.96

5.4.2 Subjective Evaluations

An **ABX** test has been used to evaluate the speaker-similarity performance of the VT system. During the **ABX** text, **A** and **B** represent the original speakers, and **X** is the converted speech either from **A** to **B** or from **B** to **A**. Subjects are asked to determine whether **X** is more similar to **A** or **B**. Only one original sentence of the speakers is provided to the subjects instead of providing the original forms of all the transformed sentences. This is to prevent the speaker-specific long-term behavior of the intonation along the sentences from effecting the decision of the subjects. The test includes 6 transformed **X** sentences and 3 transformations for two different L values ($L = 64, 96$). These are the L values which result in the highest performance indices. D is 75, which gives the highest performance indices for most of the cases. For 3 of the transformed sentences N has been equated to 4, and for the other 3 sentences N is 10. This is to observe the difference in perceptual results after dimension reduction. 2 of the transformations are from Speaker-2 to Speaker-1 and the remaining 4 transformations are from Speaker-1 to Speaker-2. All of the transformations have been done using Equation 5.8, since objective evaluations have shown that performance indices using Equation 5.8 are higher than the cases using Equation 5.9.

20 subjects have taken the test. 118 converted sentences have been detected as the target speaker out of 120 sentences in total. 2 of the incorrect decisions were for $L = 96$ and $N = 10$. Since no errors have been detected with $N = 4$ case, it is possible to say that dimension reduction increases the system performance. When compared to the MELP-based VT system, the system presented in this chapter performs slightly better in terms of human perception.

This system has a correct speaker detection rate of 98.3%, while the MELP-based system has that of 96.7 %.

CHAPTER 6

CONCLUSIONS

6.1 Summary

In this dissertation, new approaches in the design of VT techniques are considered. A triphone-balanced speech corpus and segmentation tools for Turkish have been developed, which were required to develop a VT system for Turkish.

In Chapter 3, we have presented the speech corpora and recognition tools, which we have developed as a basis for our VT research. A phonetic alphabet, METUbet, which includes not only phonemes but also allophones in Turkish [10], has been developed. Using a 2.5 million text corpus collected from online newspapers and the phonetic alphabet, triphone occurrence rates in Turkish have been determined. These rates have been used to obtain a triphone-balanced set of 2462 sentences. Audio corpus of 100 speakers, each uttering 40 sentences selected from the balanced sentence-set, has been used to train the speech recognition system of CSLR, *Sonic* [49]. During the part of the research at CSLR, *Sonic* has been ported to Turkish to develop a phonetic aligner and a phoneme recognizer. Then the 100-speaker triphone-balanced audio corpus has been labelled with word-level, phoneme-level, and HMM-state level alignments. Using a 20 speaker test-set (separate from the 100-speaker train set), both systems have been evaluated. Phoneme boundaries determined by the phonetic-aligner have been compared with the hand-aligned phoneme boundaries. 91.2% of the boundaries were inside the 20 msec. neighborhood of the hand-aligned boundaries. For the phoneme recognizer, the overall phone

error rate has been determined to be 29.3%.

In Chapter 4, we have proposed and implemented a new approach for VT, which is based on MELP speech analysis. This system is based on obtaining speaker-specific codebooks of LSFs out of MELP's MSVQ LSF codebook. Those codebooks are used to train a mapping histogram, which is used for LSF transformation from one speaker to the other. The baseline system uses the maxima of the histograms for LSF transformations. The shortcomings of this system, which are the limitation of the target LSF space and spectral discontinuities due to independent mapping of subsequent frames, have been overcome by applying a dynamic programming approach. This approach considers all target codewords corresponding to one source codeword using the probabilities obtained from the histogram matrix. Dynamic programming also considers the LSF distances between the subsequent frames of the source speaker. This approach improved the performance of the system in terms of both the speech quality and the speaker identification.

In Chapter 5, we have introduced a new LSF quantization scheme using principle component analysis for dimension reduction and k-means clustering. LSF codebooks are obtained individually for each speaker. The same dynamic programming approach as in Chapter 4 has been applied for final LSF computation. Objective evaluations have shown that this system performs better than the MELP based system in Chapter 4 in terms of voice similarity with the target speaker. It has also been shown that dimension reduction using PCA improves the system performance due to more efficient quantization.

6.2 Directions for Future Research

Further improvements in VT performance can be achieved by addressing the problems and extending solutions of the methods described in this thesis.

The proposed method uses sentences out of the triphone-balanced audio corpus. Recordings for each speaker have been done independently. This may cause the speakers to utter sentences with different durations and intonations,

especially while uttering long sentences. The recording session could be improved such that the speakers listen to a prerecorded utterance before they utter each sentence, and then they mimic the intonation in the prerecorded one. This would reduce the need for DTW before training the system. DTW may introduce errors in aligning the frames of the two speakers, which reduces the performance of the mapping. Moreover, recording phrases or words instead of long sentences may lead to better control over the intonation and durations. With long sentences, however, small pauses, glottal stops, or vowel reductions occur, which causes errors during automatic segmentation and frame matching.

The proposed method attempts to convert only the spectral behavior of speech. Spectral behavior has been reported to carry the highest amount of speaker individuality in the literature [6, 20]. However, speaker individuality also includes the prosodic characteristics specific to a speaker. The speaking rate, pitch contour, and durations also make up part of the speaker individuality [21]. Our system modifies the pitch contour linearly to match the mean pitch; however, it has been reported that time averaged pitch contours play a much smaller role in speaker individuality than dynamics of the pitch contours [21]. Long term behavior of the prosody may be modelled and transformed from one speaker to the other to increase the VT performance.

Both of the methods we have proposed for VT modify the LPC spectrum of the source speaker and changes only the pitch period of the residual. This is the state-of-the-art in many VT systems today. However, the residual signal also carries some amount of speaker information [6]. During our research, we have observed a correlation between the LSF codewords of MELP and the quantized Fourier magnitude peaks of the residual signal. This correlation may be used to obtain the best target residual relating the modified LPC spectrum, instead of modifying the source speaker's residual. This approach has been applied in [6], however, it has been reported that voice quality degrades while increasing the similarity with the target. The same approach may be applied to improve the performance of our VT system after modifications to satisfy the speech

quality.

TTS systems today use corpus-based approaches as presented in Appendix A. The reason for that is the concatenation of prerecorded speech units results in much more natural synthetic speech rather than using model-based approaches. A similar approach may be used for VT in such a way that parts of speech are selected from a pre-recorded speech corpus of the target speaker and then concatenated. The challenge in this approach is that the source speaker's speech should be clearly recognized so that the correct correspondence in the target speaker's corpus is determined. Unit selection approaches used in TTS, which increase the naturalness of synthetic speech, could be ported to VT research area.

Besides the proposed future works above, more comprehensive inspections on the phenomenon of speaker individuality may be proposed. What makes a speaker's speech individual exactly still remains to be question needing clearer answers. Today's state-of-the-art VT systems use standard feature sets such as LSFs and cepstral coefficients to obtain functions mapping the speech of different speakers. Human perception, on the other hand, may be based on different feature sets for different speakers. For example, average pitch period may be the discriminative feature to identify a male-female speaker pair. For two male speakers with similar average pitch periods, on the other hand, the discriminative feature might be the spectral behavior of speech, or the pitch contour along a sentence. Speaker-specific feature selection for VT may be a future direction. Moreover, investigation of new feature sets or new speech models other than today's state-of-the-art speech models which would characterize the individualities in the speech signal is still an interesting and an open field of work.

APPENDIX A

SPEECH SYNTHESIS TOOLS DEVELOPED FOR TURKISH SPEECH RESEARCH

In this chapter, we present our work on speech synthesis tools developed for Turkish. First a brief introduction to the text-to-speech (TTS) systems will be given. Then the tools developed for a Turkish TTS system will be presented.

A.1 Introduction and Background

The goal of TTS synthesis is to enable a machine to transmit oral information to a user in a man-machine communication context. A TTS system aims to read any text, whether it is directly introduced in the computer by an operator or scanned and submitted to an optical character recognition system. Reading should be intelligible and natural. In the context of TTS synthesis, it is impossible to record and store all words of the focus language. Therefore, TTS can be defined as production of speech by machines, by the way of automatic phonetization of the sentences to utter. Current TTS systems address three areas: text and phonetic analysis, prosody generation, and speech synthesis. Text analysis module converts the text that enters the synthesizer in some electronically coded format into a linguistic representation. Tagged and phonetically labelled text is feeded into the prosody predictor, which determines prosody parameter values. The signal processing module outputs the

synthesized speech with the prosodic properties determined by the prosody generation module.

Speech waveform generation in a TTS system can be achieved using two different approaches: Rule-based TTS and corpus-based TTS. The early TTS was constructed based on rules that researchers determined from their objective decisions and experience [69]. The researcher extracts the rules for speech production by an Analysis-by-Synthesis (AbS) method. In the AbS method, parameters characterizing a speech production model are adjusted by performing iterative feedback control so that the error between the observed value and that produced by the model is minimized. Such rule determination needs professional expertise since it is difficult to extract consistent and reasonable rules [33]. The rule based TTS has usually an unnatural speech quality because speech waveform is generated by a model, which generally needs some approximations in order to model the complex human vocal mechanism [69]. Current TTS systems are usually corpus-based synthesizers, in which a large amount of speech data is used. This approach has been developed recently through the improvements in computer speeds and performances. Output waveform speech is synthesized by concatenating the selected units from the speech corpus and then modifying their prosody. The structure of corpus-based TTS is given in Figure A.1. Corpus-based TTS systems can synthesize speech more natural than the rule-based TTS systems, because the concatenated parts are the natural speech parts themselves. If the selected units need little modification, natural speech can be synthesized by concatenating speech waveform segments directly.

In this thesis, the speech synthesis tools developed are aiming a diphone-based speech production model. A diphone-based TTS system is a special kind of a corpus-based synthesizer, which concatenates diphones previously recorded to synthesize speech. The corpus consists of one sample (or a set of samples) for each phone-pair (diphone) existing in the language. The TTS engine, *Festival Speech Synthesis System*, has been used as the development

framework [70]. *Festival* offers a general multi-lingual framework for building speech synthesis systems as well as including examples of various modules. Voices in many languages including English (UK and US), Spanish and Welsh has been developed using *Festival*.

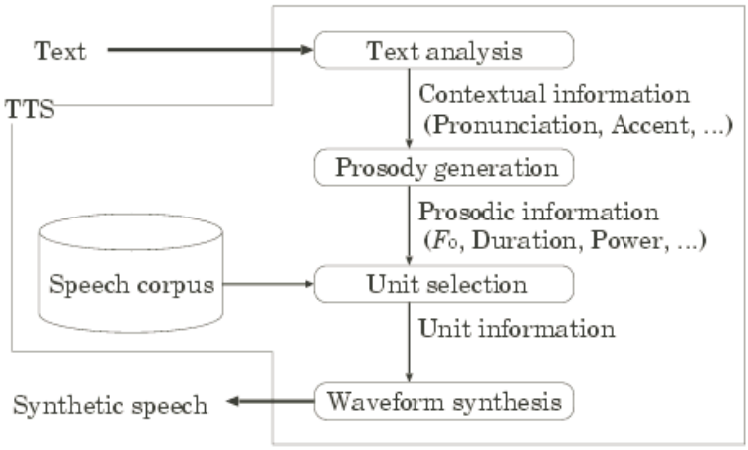


Figure A.1: Structure of corpus-based TTS.

We are going to give brief information on the modules of a corpus-based TTS system illustrated in Figure A.1, before moving on to the tools developed for a Turkish diphone-based TTS synthesizer.

A.1.1 Text Analysis

In the text analysis module, an input text is converted into contextual information, such as pronunciation, part-of-speech and so on. First elements of the document structure which may have direct implications on prosody, such as sentence breaking and paragraph segmentation, are determined. Then *text normalization*, which involves conversion from various numbers, symbols, dates, phone numbers, cardinals into a common orthographic transcription suitable for phonetic transcription. In some systems (for example Chinese), a designated symbol marks the end of the sentence, while in many languages as such as English and Turkish a period marks both the end of the sentence and the abbreviations. Tokenization into words in many Asian languages such as Mandarin Chinese is not trivial, because spaces are not used to delimit

words. A minimal requirement for this task is an online dictionary, however this is not usually enough to overcome this problem, because there are words not found in the dictionary. After the text normalization a *linguistic analysis* is applied, which recovers the syntactic and semantic features of the words, phrases, clauses and sentences important for both pronunciation and prosodic choices. Commercial TTS systems usually have some minimal parsing heuristics developed strictly for TTS [35]. During the linguistic analysis, the normalized text is divided into morphemes, which are minimum units of letter strings having linguistic meaning. These morphemes are tagged with their parts of speech, and a syntactic analysis is performed. Then, the module determines phoneme and prosodic symbols, e.g. accent nucleus, accentual phrases, boundaries of phonetic clauses, and syntactic structure [56].

A.1.2 Prosody Generation

In the prosody generation module, prosodic features such as F_0 contour, amplitude (power contour), and phoneme duration are predicted from the contextual information output from the text analysis. Prosodic information is important for the intelligibility and naturalness of the synthetic speech.

One of the most famous models that represent the F_0 contour is Fujisaki's model [56, page 166]. This model decomposes the F_0 contour into two components: a phrase component that decreases gradually toward the end of the sentence, and an accent model that increases and decreases rapidly at each accentual phrase. This model also fits the observations we have made previously on Turkish sentences [71]. Many data-driven algorithms to model F_0 have been proposed. In [72], an F_0 contour of a whole sentence is produced by concatenating segmental F_0 contours which are generated by modifying vectors that are representatives of typical F_0 contours. Using the natural F_0 contours selected from a speech corpus, instead of representative vectors, to generate an F_0 contour is proposed in [73]. In this algorithm, if there is an F_0 contour having equal contextual information to the predicted information in

the speech corpus, the F_0 contour is used without modification. Algorithms using *ToBI* labels have been proposed and widely used for F_0 contour prediction [35, page 742]. HMM-based methods to model both the F_0 contour and duration have also been proposed [74]. In this method, the F_0 contour, power contour, and phoneme durations are generated directly by HMMs. Some TTS systems do not perform the prosody generation, but contextual information is used instead of the prosody information for the unit selection stage [75].

In our TTS system, prosody modification module is a simple one. A rule-based duration module and an F_0 model similar to Fujisaki's method obtained from our observations in [71] have been used.

A.1.3 Unit Selection

In this module, optimum set of units is selected from a speech corpus by minimizing the degradation of naturalness caused by various factors, such as prosodic difference, or spectral difference. Many types of basic speech units have been used in the literature. Phonemes, diphones [76], VCV (vowel-consonant-vowel) units or CVC units [77]. In order to use the stored data effectively and flexibly, using non-uniform units with variable lengths have also been proposed [78, 79]. In this approach, an optimum set of synthesis units are selected by minimizing a cost capturing the degradation caused by spectral difference, difference in phonetic environment, and concatenation between units in a synthesis procedure.

Units in our system are diphones. The basic idea behind building diphone databases is to explicitly list all possible phone-phone transitions in a language. This makes the wrong, but practical, assumption that co-articulatory effects never go over more than two phones [70]. The exact definition of phone here is in general non-trivial because various allophonic variations may in some cases be also included, as will be discussed in our work in Section A.2.1. Unlike generalized unit selection where multiple occurrences of phones may exists with various distinguishing features, in a diphone database only one occurrence of

each diphone is recorded. This makes selection much easier but also makes for a large laborious collection task. Diphone synthesis has the advantage that the memory required for the TTS is extensively reduced compared to the multiple unit-selection synthesis case.

A.1.4 Waveform Synthesis

An output speech waveform is synthesized from the selected units in the final module of the TTS. In general two approaches have been used: Waveform concatenation without speech modification and speech synthesis with speech modification.

Waveform concatenation without speech modification uses the natural variation of the acoustic units from a large speech database to reproduce the desired prosodic characteristics in the synthesized speech [80]. Therefore, synthetic speech has no degradation caused by signal processing. Any degradation in the synthetic speech quality is caused by the possible prosodic errors [33]. In order to prevent this degradation, it is necessary to prepare a large-size speech corpus.

Speech synthesis with speech modification uses signal processing techniques to generate a speech waveform with target prosody. The Time-Domain Pitch-Synchronous OverLap-Add (TD-PSOLA) has been proposed, which changes the prosody in time domain by re-arranging the center positions of windowed speech frames to modify pitch [81]. Harmonic plus Noise Model (HNM) has been proposed as a high-quality speech modification technique [82]. In this model, speech signals are represented as a time-varying harmonic component plus a noise component. In our system, residual-excited LPC method (RELP) is used [70, 7]. RELP method modifies the inverse filtered residual signal and re-synthesizes speech using the same LPC filter driven by the modified residual.

In terms of naturalness of synthetic speech, waveform concatenation without modification outperforms, however naturalness is not always consistent. Speech synthesis with modification does not sound as natural as waveform

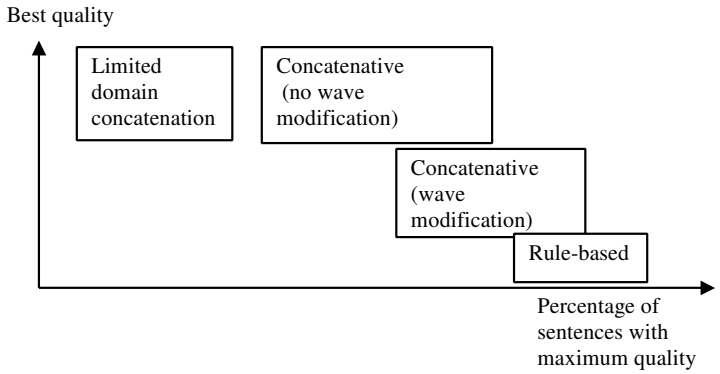


Figure A.2: Quality and task-independence in speech synthesis.

concatenation, but the sound quality is consistent. It is a common experience that a single system can sound well on one sentence and terrible on the other. For that reason, quality of the best sentences and the percentage of the sentences for which such a quality is reached are considered separately [35]. This tradeoff is illustrated in Figure A.2.

A.2 Diphone Synthesis Tools Developed for Concatenative Speech Synthesis

A new diphone corpus for concatenative speech synthesis has been designed and recorded for Turkish. The resulting corpus has been integrated into the *Festival* Speech Synthesis System [76]. The basic idea behind building a diphone corpus is to explicitly list all possible phone-to-phone transitions in a language. This is based on the practical assumption that co-articulatory affects do not go more than over two phones. All possible diphone pairs for Turkish have been determined based on the METUbet phonetic symbols [7]. Tools for constructing a list of nonsense carrier words for those diphones and collecting the audio corpus have been developed. The diphone list and simple prosodic modules have been integrated into the *Festival* Speech Synthesis System and the resulting speech synthesizer has been evaluated using a Diagnostic Rhyme Test (DRT) [7].

Table A.1: New symbols added to the METUbet alphabet for speech synthesis.

METUbet	Example
AAG	*Ağ*aç
AG	L*ağ*ım
EEG	Er*eğ*li
EG	*Eğ*len
IG	*Iğ*dir

A.2.1 Diphone Corpus Construction

Before developing the diphone list for Turkish, some new symbols have been added to the METUbet symbol list given in Figure 3.1, to consider the affect of $\breve{g}$ on Turkish vowels. $\breve{g}$ usually lengthens the vowel it precedes or acts as a weak Y when it is between front vowels [10]. It has not been considered in the aligner and the phoneme-recognizer development as explained in Section 3.4.2.1. For synthesis, on the other hand, phoneme durations are important and should be considered. Therefore, we have considered vowels followed by the $\breve{g}$ as new phonemes instead of considering the $\breve{g}$ by itself. Considering the $\breve{g}$ alone is impossible in terms of developing special carrier words, since it is actually not a phoneme, realized by itself, but it acts on vowels to change their way of articulation [10]. The new symbols added to METUbet, considering $\breve{g}$, for speech synthesis purposes are listed in Table A.1.

The new symbol set has 48 symbols to represent Turkish phonemes and allophones. The typical diphone size is the square of the number of the phone number for any language, which is $48 \times 48 = 2304$ in our case. However, there are phonotactic constraints in human languages. Some phone-phone pairs do not occur physically. All possible diphone pairs have been determined based on letter-to-sound rules obtained from the Turkish phonetic dictionary [10]. 2283 possible diphones have been determined. A tool (*Perl* script) to construct nonsense carrier words has been developed. Nonsense words help the speaker keep constant prosody during recording. The carrier words are formed such that they obey phonetic rules of Turkish such as vowel harmony and syllable-final oral stop voicing and etc. The initial vowel in each carrier word helps

Table A.2: Examples of diphones and their carrier words

Diphone	Carrier Word	Carrier Word in METUbet
P-AA	a paradagun	AA P-AA D AA RR AA GG U NN
KK-AA	a kadaragun	AA KK-AA D AA RR AA GG U NN
A-M	a lamaragun	AA L A-M AA RR AA GG U NN
A-F	a lafaragun	AA L A-F AA RR AA GG U NN
M-B	a dambat	AA D AA M-B AA T
S-F	a dasfat	AA D AA S-F AA T
P-UE	e püderegen	E P-UE D EE RR EE G EE NN
OE-T	e böteregen	E B-OE T EE RR EE G EE NN

the speaker adjust the amplitude and it is pronounced as a long vowel sound for this adjustment, while the final *agun* or *egen* part (one of them is selected depending on the vowel harmony) helps to keep the pitch constant during recording. Table A.2 shows some of the carrier words to give an insight on the carrier-word formation strategy.

A.2.2 Recording the Diphone Corpus

A user interface tool for systematic recording of the diphone corpus has been developed. The tool displays each word for 5 seconds on the computer screen both in graphemes and in METUbet symbols and lets the user record his/her voice. Then it displays the next carrier word. The tool enables direct accusation of groups of 5, 10, 15, 20, 25, or 30 words in Windows PCM *.wav files. Recording in groups saves recoding time of all 2283 words. It also lets the speaker to increase the number of words in a recording group and save more time as he/she gets used to reading the carrier words. The tool creates a log file associated with each word group file, which carries the information of location of each individual word inside the file. This is for later separation of the group files before integrating them into *Festival*. The tool also allows the user to record multiple copies of all files, so that the speaker can listen to them during the recording session and select the best one. Figure A.3 shows the user interface during recording. The beginning word of the word group to be recorded is selected on the left panel and the selected word is displayed

below both panels. The right panel shows the already recorded files. The first file *rec0000_0004.wav* includes the first 5 carrier words, while the second one, *rec004_0013.wav*, includes 10 words, from *word-0* to *word-13*.

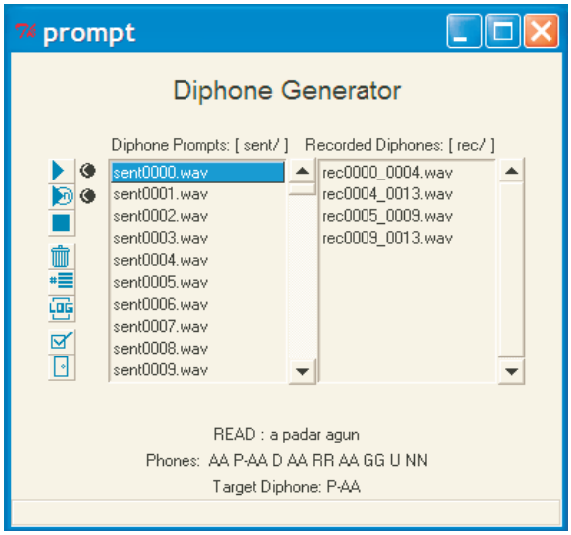


Figure A.3: Diphone recording user interface.

The audio has been collected in a sound-isolated recording studio at CSLR. Female voice at 16 kHz sampling rate has been collected. The *Festival* Speech Synthesis System can use glottal pulses for precise pitch prediction optionally. To take the advantage of this option, speech has been collected together with the glottal pulse, using a device called *electroglottograph (EGG)*. The EGG device measures the variations in the electrical impedance of the neck at the level of the vocal folds caused by the variation of vocal fold contact as the vocal folds vibrate. The method uses a pair of electrodes on the neck that apply a small current, which is safe, and measures the patterning of vocal fold contact area. The resulting waveform, when plotted simultaneously with the speech waveform, provides accurate measurements of the voiced/unvoiced regions and the pitch period. The derivative of the glottal waveform has sharp peaks at glottal openings. An example waveform with its glottal waveform is illustrated in Figure A.4. The upper panel shows the speech waveform and the lower panel shows the glottal wave. This is the diphone [EE-SH] in Turkish uttered by a female speaker. Voiced/unvoiced transition can be determined

accurately by the disappearing of the glottal pulses as observed in the figure. An EGG device of *Glottal Enterprises*, *Model EG-2 PC*, has been used for this data collection. The recording set-up is illustrated in Figure A.5. The device takes the audio and the EGG signal as input separately and combines them at the stereo output with two channels. When connected to a computer equipped with a sound card via a stereo cable, the left channel of the output signal carries the audio (microphone) signal, and the right carries channel the EGG. A *Labtec axis-502* microphone has been used to record the speech signal.

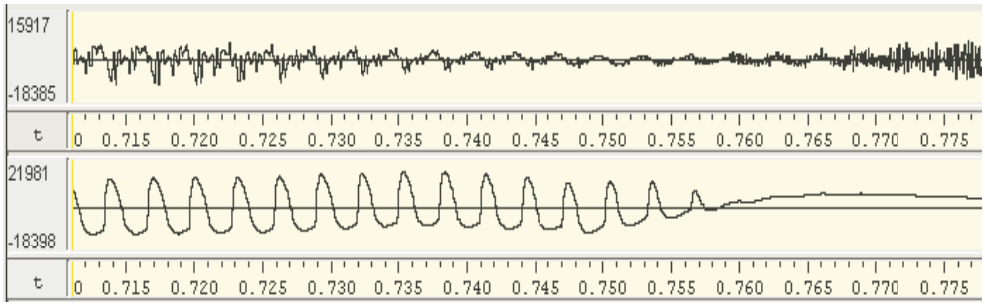


Figure A.4: Speech waveform (upper panel) and the glottal waveform (lower panel). This is female speech collected at 16kHz, presented speech part is the [eş] boundary in Turkish. Time axis is in seconds.

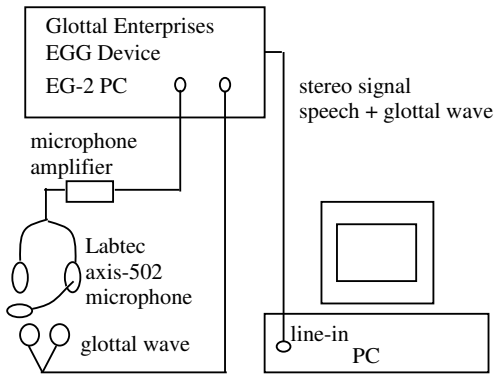


Figure A.5: Diphone recording set-up.

Once the diphone recording phase is completed, the files are needed to be integrated into *Festival*. A script to divide diphone group files into single-

Table A.3: Diphone index list (.est file) for *Festival*

EST-File index			
DataType ascii			
NumEntries 2283			
IndexName ozgul-diphone			
EST-Header-End			
P-AA rec0000	1.340	1.395	1.440
T-AA rec0001	1.195	1.255	1.300
B-AA rec0002	1.070	1.135	1.180
D-AA rec0003	1.320	1.395	1.425
M-AA rec0004	1.285	1.355	1.395
NN-AA rec0005	1.020	1.075	1.120
F-AA rec0006	0.965	1.035	1.080
S-AA rec0007	1.110	1.185	1.220
⋮	⋮	⋮	⋮

word files have been produced and used to obtain 2283 separate .wav files. *Festival* system also requires a diphone index list file (a *.est* file) [83]. This index list includes diphones, their corresponding wave file names, the start time, mid-time (phone boundary of the diphone) and the end-time in seconds. The beginning part of the index list file is shown in Table A.3. Phoneme boundaries have been obtained automatically with the *Sonic* Turkish phoneme aligner system, which has been presented in Section 3.4.2.1.

The next step for synthetic speech for a new language is integrating the natural language processing modules into *Festival*.

A.2.3 Developing Natural Language Processing Modules for Festival

Modules required by the *Festival* speech synthesizer structure were written in *Scheme* programming language, which is a dialect of the *Lisp* programming language [84]. The natural language module scripts requited by *Festival*, therefore, have been developed in *Scheme*. *Festival* requires two different types of scripts: Language specific text module scripts (phones, lexicon, tokenization) and speaker specific prosodic module scripts (duration and intonation). This

differentiation helps us define different duration and intonation styles for different speakers of the same language. Figure shows the block diagram of the natural language modules in *Festival*.

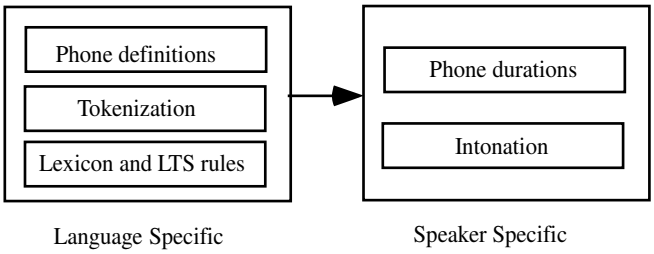


Figure A.6: Natural language processing modules in *Festival*.

A.2.3.1 Language Specific Modules

The first text analysis module is the *phone-set definition module* in which every symbol of the METUbet alphabet is classified according to phone features like vowel height, consonant voicing and etc. Phone definitions have been made based on the phone definitions given in [10]. Phone definitions for vowels are given in Table A.4.

The next module, which is the *lexicon module*, includes letter-to-sound rules. These are rules to convert from Turkish graphemes to METUbet symbols. They have been developed at METU from the Turkish phonetic dictionary [10]. Since Turkish is a phoneme-based language, the rules are able to cover almost all Turkish words, except for some words originated from other languages. There are 107 rules developed for Turkish, and they have been rewritten in *Scheme*. Some examples are given in Table A.5.

The next module is the tokenization module. This module converts numbers (decimal or integer up to 999 billion), percentages, 24-hour clock, Celsius degrees, and some common abbreviations (such as Dr. to "doktor" or, Şub. to Şubat) into a word. We have not provided a lexicon to the system, because Turkish is an agglutinative language and even a simple lexicon should include millions of words. Instead, all pronunciations are determined via letter-to-sound rules. This module also includes syllabification of words, which will be

Table A.4: Turkish vowel definitions in *Festival* text module (V/C: vowel or consonant, length: **S**hort or **L**ong, height: 1/2/3 levels, frontness: 1/2/3 levels, roundness: **R**ound or **F**lat)

PHONE	V/C	LENGTH	VOWEL HEIGHT	VOWEL FRONTNESS	VOWEL ROUNDNESS
A	V	S	3	3	F
AA	V	S	3	3	F
AG	V	L	2	2	F
AAG	V	L	2	2	F
E	V	S	3	1	F
EE	V	S	3	1	F
EG	V	L	2	1	F
EEG	V	L	2	1	F
IY	V	S	1	1	F
IYG	V	L	1	1	F
I	V	S	1	3	F
IG	V	L	1	3	F
O	V	S	3	1	R
OG	V	L	2	1	R
OE	V	S	3	3	R
OEG	V	L	2	3	R
U	V	S	1	1	R
UG	V	L	1	1	R
UE	V	S	1	3	R
UEG	V	L	1	3	R

used by the speaker specific modules to assign intonation to parts of speech.

A.2.3.2 Speaker Specific Modules

Speaker specific modules determine the major components of the prosody, which are pitch and duration of the concatenated speech segments.

The pitch contours of the sentences are determined by the intonation module. The intonation module of *Festival* predicts accents on a per syllable basis. Target fundamental frequency values are determined depending on the syllable stress values. We have used Classification and Regression Tree (CART) definitions of *Festival* to determine phase boundaries. Based on our previous work on Turkish pitch contours [71], sentence pitch contours are determined

Table A.5: Some examples of letter-to-sound rules in *Scheme*

Rule	Explanation
(u [v] a = V);	**v** after **u** and before **a** is **V** in METUbet
(o [v] u = V);	**v** after **o** and before **u** is **V** in METUbet
(ko [v] CONSONANT = V);	**v** after **ko** and before any consonant is **V** in METUbet
([v] = VV);	all other **v**'s are **VV** in METUbet
([b] = B);	all **b**'s are **B**

using CART trees. Our observations on Turkish sentence and word prosody in [71] can be summarized as follows:

- Pitch contour has a declining characteristic along the sentence for all types of sentences.
- The last syllables of all words, except for the verb, which is the final word usually, are accented (i.e. pitch contour inclines).
- Verb and the word before verb have declining pitch contours.
- The word before verb could have rising or falling pitch contour depending on the speaker.
- Negative verbs (verbs including the negation morphemes, *-me* or *-ma*), have the pitch contour rising before the negation morphemes.
- In yes/no question sentences, the question morpheme (*-mi, -mu*) is accented.

Note that these observations are far from modelling Turkish sentence or word prosody in detail, but they only provide generalizations which help the synthetic speech sound much more natural than flat intonation. Modelling Turkish prosody requires a more detailed research and it is beyond the scope of this thesis. However, the above observations provide a good framework for the speech synthesis system at the beginning.

An analysis has been made on the on the previously recorded sentences of our speaker. Mean values of the sentence-beginning and sentence-final pitch are

Table A.6: Phonemes and their mean durations for our female TTS speaker.

PHONE	DURATION (sec)	PHONE	DURATION (sec)
AA	0.054	G	0.068
AAG	0.108	GG	0.061
A	0.046	H	0.058
AG	0.092	G	0.094
E	0.050	K	0.091
EG	0.098	KK	0.090
EE	0.045	L	0.047
EEG	0.093	LL	0.050
I	0.034	M	0.068
IG	0.075	N	0.076
IY	0.035	NN	0.059
IYG	0.070	P	0.096
O	0.061	R	0.055
OG	0.123	RR	0.042
OE	0.062	RH	0.076
OEG	0.124	S	0.118
U	0.039	SH	0.114
UG	0.079	T	0.084
UE	0.035	V	0.063
UEG	0.071	VV	0.051
B	0.066	Y	0.058
C	0.078	Z	0.088
CH	0.105	ZH	0.129
D	0.056	SIL	0.607
F	0.091		

obtained. The beginning and final values have been obtained as 246 Hz and 161 Hz respectively with standard deviations 33.6 Hz and 23.1 Hz. An imaginary linear contour which connects those two points is used as the sentence pitch contour. This imaginary line is lifted up by 76 Hz (which is obtained from observations) at the first and second pitch rise, which correspond to the final syllables of the first and the second word in the sentence [71].

For the duration module, we have provided mean phoneme durations, which have been obtained from the phoneme aligned 40 triphone-balanced sentences of our female speaker. Duration list is given in Table A.6.

Table A.7: Turkish Words for Diagnostic Rhyme Test

VOICING	NASALITY	SUSTENATION	SIBILATION	GRAVENESS	COMPACTNESS
ver-fer	mit-bit	var-bar	çöp-köp	pek-tek	yer-ver
ben-pen	nar-dar	şal-çal	can-gan	bür-dür	kay-tay
cin-çin	nam-dam	ver-ber	çem-kem	mal-nal	hay-fay
den-ten	ney-dey	fay-pay	caz-gaz	pak-tak	göl-döl
zor-sor	mor-bor	şam-çam	çok-kok	bol-dol	kar-par
dün-tün	nal-dal	yer-ger	cay-kay	met-net	yurt-kurt
vol-fol	mür-dür	ser-çer	can-kan	ban-tan	gaz-baz
gel-kel	mey-bey	sar-çar	çan-kan	bön-dön	kör-pör
zen-sen	mal-dal	vak-bak	çöl-göl	ver-der	yaz-raz
dan-tan	mal-bal	sal-çal	çöz-köz	pas-tas	has-fas

A.2.4 Waveform Generation

The standard method for diphone re-synthesis, which is in the released system of *Festival*, is RELP (**R**esidual **E**xcited **L**inear **P**rediction) [85]. This method uses the residual signal of the speaker to excite the vocal tract filter obtained from LPC analysis.

A.2.5 Evaluations

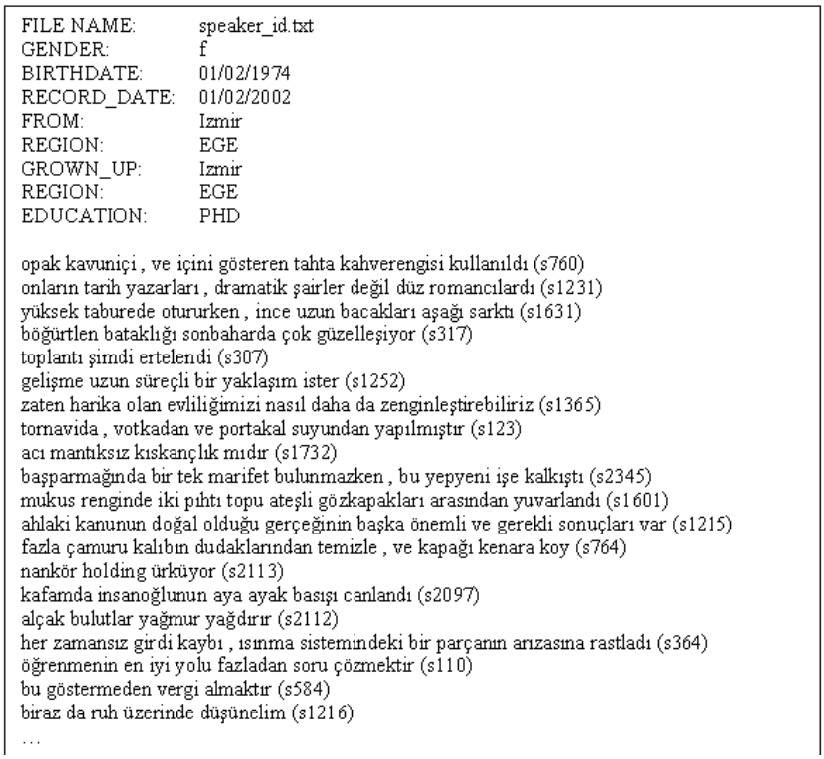
To test the intelligibility of our system, we have designed a simple **D**iagnostic **R**hyme **T**est (DRT) for Turkish. DRT is a response test with two response alternatives containing systematic, minimal phonemic contrasts in the initial consonant and it is often used for testing TTS systems [86]. The subject would be asked to indicate whether a synthetic item was intended as for example, *dune* or *tune*. 60 pairs of meaningful words with different confusability groups have been determined. Using meaningful words makes the system reliable, fast, and easy to administer and score [86]. The words have been determined considering the close correspondence with DRT standard for American English [87].

During the test, subjects can see the monosyllable similar word pairs, but they hear only one of them. The fraction of the words they identify correctly is the measure of the intelligibility of speech over the system. The six per-

ceptual attributes (voicing, nasality, sustenation, sibilation, compactness, and graveness) have been tested and 10 monosyllable pairs for each group have been determined. The words are presented in Table A.7. 12 pairs (2 words from each group) are selected randomly for each subject from the list. One word from each pair is synthesized by our TTS system. Test has been repeated for 20 subjects. The overall intelligibility of the system has been found to be 86.5%.

APPENDIX B

EXAMPLE OF A TRIPHONE-BALANCED CORPUS COLLECTION LOG FILE



FILE NAME: speaker_id.txt
GENDER: f
BIRTHDATE: 01/02/1974
RECORD_DATE: 01/02/2002
FROM: Izmir
REGION: EGE
GROWN_UP: Izmir
REGION: EGE
EDUCATION: PHD

opak kavuniçi , ve içini gösteren tahta kahverengisi kullanıldı (s760)
onların tarih yazarları , dramatik şairler değil düz romancılardı (s1231)
yüksek taburede otururken , ince uzun bacakları aşağı sarktı (s1631)
böğürtlen bataklığı sonbaharda çok güzelleşiyor (s317)
toplantı şimdi ertelendi (s307)
gelişme uzun süreçli bir yaklaşım ister (s1252)
zaten harika olan evliliğimizi nasıl daha da zenginleştirebiliriz (s1365)
tornavida , votkadan ve portakal suyundan yapılmıştır (s123)
acı mantıksız kıskançlık mıdır (s1732)
başparmağında bir tek marifet bulunmazken , bu yepyeni işe kalkıştı (s2345)
mukus renginde iki pıhtı topu ateşli gözkapakları arasından yuvarlandı (s1601)
ahlaki kanunun doğal olduğu gerçeğinin başka önemli ve gerekli sonuçları var (s1215)
fazla çamuru kalıbın dudaklarından temizle , ve kapağı kenara koy (s764)
nankör holding ürküyor (s2113)
kafamda insanoğlunun aya ayak basışı canlandı (s2097)
alçak bulutlar yağmur yağdırır (s2112)
her zamansız girdi kaybı , ısınma sistemindeki bir parçanın arızasına rastladı (s364)
öğrenmenin en iyi yolu fazladan soru çözmektir (s110)
bu göstermeden vergi almaktır (s584)
biraz da ruh üzerinde düşünelim (s1216)
...

Figure B.1: Beginning part of a triphone-balanced corpus collection log-file.

Figure B.1 illustrates the beginning part of one example log file prepared for the speaker, whose specifications are recorded at the beginning of the file. This file is prepared before the recording session and 40 sentences are selected randomly for every speaker from the triphone-balanced sentence set.

APPENDIX C

MUTUAL INFORMATION

In this section, preliminary basic information on the mutual information concept will be given.

Mutual Information:

Mutual information is a measure of the amount of information that one random variable contains about the other [66]. Before giving its definition, the concept of *entropy* is explained here very briefly. Let X be a discrete random variable with alphabet χ and the probability mass function $p(x) = Pr\{X = x\}$, $x \in \chi$. Note that, here probability mass function is denoted by $p(x)$ rather than $p_X(x)$ for convenience. Therefore, $p(x)$ and $p(y)$ will actually refer to two different probability mass functions $p_X(x)$ and $p_Y(y)$ respectively.

Definition: *Entropy*

The entropy of a random variable is a measure of the uncertainty of the random variable. In other words, entropy is a measure of the amount of information required on the average to describe the random variable. Entropy, $H(X)$, is defined by:

$$H(X) = -\sum_{x \in \chi} p(x) \log p(x). \tag{C.1}$$

When the log is to the base 2, entropy is expressed in bits.

Definition: *Joint Entropy*

If we have a pair of random variables X and Y with a joint distribution $p(x, y)$, we can define the joint entropy, $H(X, Y)$, as:

$$H(X,Y) = -\sum_{x\in\chi}\sum_{y\in\psi} p(x,y)\log p(x,y), \tag{C.2}$$

where ψ is the alphabet of random variable Y.

Definition: *Conditional Entropy*

Conditional entropy of a random variable is the uncertainty of one random variable, knowing the distribution of the other random variable. For X and Y, with a joint distribution $p(x,y)$, conditional entropy, $H(Y|X)$, is defined as:

$$\begin{aligned} H(Y|X) &= \sum_{x\in\chi} p(x)H(Y|X=x) \\ &= -\sum_{x\in\chi} p(x)\sum_{y\in\psi} p(y|x)\log p(y|x) \\ &= -\sum_{x\in\chi}\sum_{y\in\psi} p(x,y)\log p(y|x). \end{aligned} \tag{C.3}$$

Note that $H(X|Y) \neq H(Y|X)$. However, $H(X) - H(X|Y) = H(Y) - H(Y|X)$, which will be defined as the *mutual information* below later. This is the amount of decrease in the entropy of one random variable when the other random variable is known.

Definition: *Relative Entropy*

The relative entropy is a measure of the inefficiency of assuming that the distribution is $q(x)$, when the true distribution is $p(x)$. The relative entropy between two probability distributions $p(x)$ and $q(x)$, $D(p||q)$, is defined as:

$$D(p||q) = \sum_{x\in\chi} p(x)\log\frac{p(x)}{q(x)}. \tag{C.4}$$

The above definition depends on the convention that $0\log\frac{0}{q} = 0$ and $p\log\frac{p}{0} = \infty$. Relative entropy is zero, when $p = q$, and it is always non-negative. However, it is not a true distance measure, since it is not symmetric and it does not satisfy the triangle inequality.

Definition: *Mutual Information*

Mutual information is a measure of the amount of information that one random variable contains about another random variable. It is the reduction

in the uncertainty of one random variable due to the knowledge of the other. Mutual information, $I(X;Y)$, is the relative entropy between the joint distribution, $p(x,y)$, and the product distribution $p(x)p(y)$, i.e.,

$$I(X;Y) = \sum_{x\in\chi}\sum_{y\in\psi} p(x,y)\log\frac{p(x,y)}{p(x)p(y)}. \tag{C.5}$$

It can also be shown that $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$ [66].

APPENDIX D

PRINCIPLE COMPONENT ANALYSIS

In principle component analysis (PCA), a set of data are summarized as a linear combination of an orthonormal set of vectors. The data matrix is $X_{n\times d}$, whose rows are data vectors x_i, $i = 1, \cdots, n$. The data vectors, x_i, are summarized by the approximating function,

$$f(x, T) = u + (xT)T^T, \tag{D.1}$$

where$f(x, T)$ is a vector-valued function, u is the mean of the data $\{x_i\}$, and T is a $d\times m$ matrix with orthonormal columns. The principle component decomposition estimates the projection matrix T, which minimizes the risk function,

$$R(x, T) = \frac{1}{n}\sum_{i=1}^{n} \|x_i - f(x_i, T)\|^2 \tag{D.2}$$

subject to the condition that the columns of T are orthonormal [68]. The mapping $z_i = x_iT$ provides a low-dimensional projection of the vectors x_i if $m < d$. If $m = d$, then it is possible to reconstruct x_i perfectly by back-transforming z_i with T^{-1}.

Without loss of generality, the data can be assumed to be zero-mean, and u can be set to zero. The transformation matrix T, which minimizes $R(x, T)$ in Equation D.2, is determined using the *singular value decomposition* (SVD) of the data matrix X [68]. SVD is given as

$$X = U\Sigma V^T, \tag{D.3}$$

where the columns of U are the eigenvectors of XX^T, and the columns of V are the eigenvectors of X^TX. The matrix Σ is a $n \times d$ matrix which is in the form:

$$\Sigma = \begin{bmatrix} \tilde{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \ \tilde{\Sigma} = diag(\sigma_1, \sigma_2, \cdots, \sigma_w). \tag{D.4}$$

where $\mathbf{0}$ are null matrices and σ_i^2 are the non-zero eigenvalues of the Hermitian and nonnegative definite matrix X^TX. σ_i are arranged such that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_w \geq 0$ and $\sigma_{w+1}, \sigma_{w+2}, \cdots$ are all zero, with $1 \leq w \leq d$.

To produce a projection with dimension $m < d$, which has maximum variance, all the eigenvalues σ_i^2 except the first m are set to zero. Then we redefine the decomposition as

$$\tilde{X} = \tilde{U}\Sigma_m T^T, \tag{D.5}$$

where Σ_m denotes the modified diagonal $d \times d$ eigenvalue matrix where only the first m elements on the diagonal are nonzero and they are $\sigma_1, \sigma_2, \cdots, \sigma_m$. T is a $d \times m$ matrix constructed from the first m columns of V and $\tilde{U}$ is constructed from the first m columns of U and it is of size $n \times m$. $\tilde{X}$ is the best approximation to X in the sense of minimization of Equation D.2. Then the m-dimensional projection vectors are given by

$$Z = XT, \tag{D.6}$$

where Z is an $n \times m$ matrix whose rows correspond to the projection z_i for a given data sample x_i.

The principle components have the following optimal properties in the class of linear functions $f(x, T)$ [68]:

- The principle components provide a linear approximation that represents the maximum variance of the original data in a low-dimensional projection.

- Z provide the best low-dimensional linear representation in the sense that the total sum of the squared distances from data points to their projections on the principle components in the space are minimized.
- The function $f(x, T) = (xT)T^T$ minimizes the risk function in (D.2). Note that since T has orthonormal columns, the left inverse of it is T^T.

APPENDIX E

K-MEANS CLUSTERING ALGORITHM

The *k-means* algorithm is actually the *Lloyd Iteration* algorithm, which is used for clustering data [67]. The algorithm starts with a given initial codebook and ends with an improved codebook with a reduced average distortion D, which is defined as $E\{d(X, Q(X))\}$. $E\{\cdot\}$ is the expectation operation, $d(X, Q(X))$ is the distortion measure between the set of input vectors, X, and its quantized form, $Q(X)$. Assume X is a data matrix with row vectors x_i. The algorithm is given as:

- **Step 1:** Begin with an initial codebook C_1. Set $m = 1$. $C_m = \{y_i; i = 1 \cdots, N\}$.
- **Step 2:** Find the optimal partition into quantization cells, that is, use the nearest condition to form the nearest neighbor cells: $R_i = \{x : d(x, y_i) \leq d(x, y_j); all j \neq i\}$.
- **Step 3:** Generate the improved codebook $C_{m+1} = \{centroid(R_i); i = 1, \cdots, N\}$.
- **Step 4:** Compute the average distortion for C_{m+1}. If it has changed by a small amount since the last iteration, stop. Otherwise set $m + 1 \rightarrow m$ and go to Step 2.

The stopping criterion commonly tests if the fractional drop in the average distortion, $(D_m - D_{m+1})/D_m$, is below or above a suitable threshold. If the algorithm converges to a codebook in the sense that further iterations no

longer produce any changes in the set of reproduction values, then the resulting codebook is at least sub-optimal [67].

REFERENCES

[1] *TIMIT Acoustic-Phonetic Continuous Speech Corpus.* National Institute of Standards and Technology Speech Disc 1-1.1, NTIS Order No. PB91-5050651996, October 1990.

[2] T. Schultz and A. Waibel, "Language-independent and language-adaptive acoustic modeling for speech recognition," *Speech Communication, Elsevier Science B.V.*, vol. 35, pp. 31–51, August 2001.

[3] M. U. Doğan and L. M. Arslan, "Speaker-based evaluation of isolated word recognition on tübitak-turtel auido database (in turkish)," in *Proc. of the 10th Signal Processing and Communications Applications Conference, SİU'02*, 2002.

[4] I. Zitouni, J. Olive, D. Iskra, and et.al., "Orientel: Speech-based interactive communication applications for the mediterranean and the middle east," in *Proc. of the 7th International Conference on Spoken Language Processing, ICSLP'02*, 2002.

[5] T. Çiloğlu, D. Acar, and A. Tokatlı, "Orientel: Speech-based interactive communication applications for the mediterranean and the middle east," in *Proc. of the 7th International Conference on Spoken Language Processing, ICSLP'02*, 2004.

[6] A. Kain, *High Resolution Voice Conversion.* PhD thesis, OGI School of Science Engineering at Oregon Health Science University, Portland, Oregon, 2001.

[7] Ö. Salor, B. L. Pellom, and M. Demirekler, "Implementation and evaluation of a text-to-speech synthesis system for turkish," in *Proc. of the 8th European Conference on Speech Communication and Technology, EUROSPEECH'03*, 2003.

[8] A. Kain and M. Macon, "Spectral voice conversion for text-to-speech synthesis," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'98*, vol. 1, pp. 285–288, 1998.

[9] Y. Stylianou, "Assessment and correction of voice quality variabilities in large speech databases for concatenative speech synthesis," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'99*, vol. 1, 1999.

[10] İ. Ergenç, *Konuşma Dili ve Türkçe'nin Söyleniş Sözlüğü*. Simury Yayınları, 1995.

[11] *Specifications for the Analog to Digital Conversion of Voice by 2400 Bit/Second Mixed Excitation Linear Prediction*. Federal Information Processing Standards Publication, 1997.

[12] J. D. Markel and J. A. H. Gray, *Linear Prediction of Speech*. Springer-Verlag, 1976.

[13] D. H. Klatt and L. C. Klatt, "Analysis, synthesis, and perception of voice quality variations among female and male talkers," *Journal of Acoustical Society of America*, vol. 87, pp. 820–837, February 1990.

[14] A. E. Rosenberg, "Effect of glottal pulse shape on the quality of natural vowels," *Journal of Acoustical Society of America*, vol. 49, no. 2, pp. 583–590, 1971.

[15] H. Kuwaba and Y. Sagisaka, "Acoustic characteristics of speaker individuality: Control and conversion," *Speech Communication*, vol. 16, pp. 165–173, 1996.

[16] K. Itoh, "Perceptual analysis of speaker identity," in *In Speech Science and Technology, S. Saito, Ed. IOS Press, ch. 2.6*, pp. 133–145, 1992.

[17] H. Matsumoto, S. Hiki, T. Sone, and T. Nimura, "Multidimentional representation of personal quality of vowels and its acoustical correlates," *IEEE Transactions on Audio and Electroacoutics*, vol. 21, pp. 428–436, October 1973.

[18] B. Necioğlu, M. A. Clements, T. P. B. III, and A. Schmidt-Nielsen, "Perceptual relevance of objectively measured decriptors for speaker characterization," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'98*, vol. 2, pp. 869–872, 1998.

[19] T. Takagi and H. Kuwaba, "Contributions of pitch, formant frequency and bandwidth to the perception of voice-personality," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'86*, 1986.

[20] H. Kuwaba, "A perceptual experiment on voice individuality by altering pitch and formant frequencies," in *132th Meeting of Acoutical Society of America*, 1996.

[21] M. Akagi and T. Ienaga, "Speaker individuality in fundemantal frequency contours and its control," in *Proc. of European Conference on Speech Communication and Technology, EUROSPEECH'95*, 1995.

[22] D. G. Childers, "Glottal source modeling for voice conversion," *Speech Communication*, vol. 16, no. 2, pp. 127–138, 1995.

[23] M. Narendranath, H. A. Murthy, S. Rajendran, and B. Yegnanarayana, "Transformation of formants for voice conversion using artificial neural networks," *Speech Communication*, vol. 16, no. 2, pp. 207–216, 1995.

[24] M. Abe, S. Nakamura, S. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'88*, pp. 655–658, 1988.

[25] H. Valbret, E. Moulines, and J. P. Tubach, "Voice transformation using psola technique," *Speech Communication*, vol. 11, no. 2, pp. 175–187, 1992.

[26] L. M. Arslan and D. Talkin, "Speaker transformation using sentence hmm based alignments and detailed prosody modification," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'98*, pp. 289–292, 1998.

[27] L. M. Arslan, "Speaker transformation algorithm using segmental codebooks (stasc)," *Speech Communication*, vol. 28, no. 3, pp. 211–226, 1999.

[28] Y. Stylianou, O. Cappé, and E. Moulines, "Statistical methods for voice quality transformation," in *Proc. of European Conference on Speech Communication and Technology, EUROSPEECH'95*, pp. 447–450, 1995.

[29] S. Furui, "Research on individuality features in speech waves and automatic speaker recognition techniques," *Speech Communication*, vol. 5, pp. 183–197, 1986.

[30] Y. Stylianou, O. Cappé, and E. Moulines, "Continous probabilistic transform for voice conversion," *IEEE Transactions on Speech and Audio Processing*, vol. 6, pp. 451–454, March 1998.

[31] Ö. Salor, M. Demirekler, and B. L. Pellom, "A system for voice conversion based on adaptive filtering and lsf distance optimization for text-to-speech synthesis," in *Proc. of the 8th European Conference on Speech Communication and Technology, EUROSPEECH'03*, 2003.

[32] H. Mizuno and M. Abe, "Voice conversion algorithm based on piecewise linear conversion rules of formant frequency and spectrum tilt," *Speech Communication*, vol. 16, no. 2, pp. 153–164, 1995.

[33] T. Toda, *High-Quality and Flexible Speech Synthesis with Segment Selection and Voice Conversion.* PhD thesis, Nara Institude of technology, Japan, 2003.

[34] Ö. Salor and M. Demirekler, "Spectral modification for context-free voice conversion using melp speech coding framework," in *Proc. of International Symposium on Intelligent Multimedia, Video and Speech Processing, ISIMP'04*, 2004.

[35] X. Huang, A. Acero, and H. W. Hon, *Spoken Language Processing, A Guide to Theory, Algorithm, and System Development.* Prentice Hall PTR, 2001.

[36] K. Shikano, K. Lee, and R. Reddy, "Speaker adaptation through vector quantization," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'86*, 1986.

[37] S. K. Lee, D. H. Youn, and I. W. Cha, "Voice personality transformation using an orthogonal vector space conversion," in *Proc. of the European Conference on Speech Communication and Technology, EUROSPEECH'95*, 1995.

[38] N. Iwahashi and Y. Sagisaka, "Speech spectrum conversion based on speaker interpolation and multi-functional representation with weighting by radial basis function networks," *Speech Communication, Elsevier Science B.V.*, vol. 16, pp. 139–151, 1995.

[39] O. Türk and L. M. Arslan, "Subband based voice conversion," in *Proc. of the 7th International Conference on Spoken Language Processing, ICSLP'02*, 2002.

[40] B. L. Pellom and J. H. L. Hansen, "Spectral normalization employing hidden markov modelling of line spectrum pair frequencies," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'987*, 19987.

[41] B. L. Pellom, *Sonic: The University of Colorado Continuous Speech Recognizer.* Technical Report: TR-CSLR-01, CSLR, University of Colorado, March 2001.

[42] J. Kornfilt, *Turkish.* Routledge, 1997.

[43] C. A. I. P. Association, *Handbook of the International Phonetic Association, A Guide to the Use of the International Phonetic Alphabet.* Cambridge University Press, 1999.

[44] *SAMPA, Computer Readable Phonetic Alphabet.* Ed. J.C. Wells. Aug. 2004. Department of Phonetics and Linguistics, University College London. $\langle http : //www.phon.ucl.ac.uk/home/sampa/\rangle$, *Last accessed* : *Jan.*2005.

[45] *SAMPA for Turkish.* Ed. J.C. Wells. May. 2003. Department of Phonetics and Linguistics, University College London. $\langle http : //www.phon.ucl.ac.uk/home/sampa/turkish.htm\rangle$, *Last accessed* : *Jan.*2005.

[46] Ö. Salor, B. L. Pellom, T. Çiloğlu, and M. Demirekler, "On developing new text and audio corpora and speech recognition tools for the turkish

language," in *Proc. of the 7th International Conference on Spoken Language Processing, ICSLP'02*, 2002.

[47] M. K. Ravishankar, *Efficient Algorithms for Speech Recognition.* PhD thesis, Carnegie Mellon University, 1996.

[48] K. F. Lee, "Context-dependent phonetic hidden markov models for speaker-independent continuous speech recognition," *IEEE Transactions of Acoustics, Speech and Signal Processing*, vol. 38, pp. 599–609, April 1990.

[49] *SONIC: Large Vocabulary Continuous Speech Recognition System.* [http://cslr.colorado.edu/beginweb/speech_recognition/sonic.html].

[50] R. G. Leonard and G. Doddington, *TIDIGITS.* Linguistic Data Consortium, University of Pennsylvania, ISBN:1-58563-018-7, 1993.

[51] J. Zhang, W. Ward, and B. L. Pellom, "Improvements in audio processing and language modeling in the cu communicator," in *Proc. of the European Conference on Speech Communication and Technology, EUROSPEECH'01*, 2001.

[52] LDC94S13A, *CSR-II (WSJ1) Complete.* Linguistic Data Consortium, University of Pennsylvania, ISBN:1-58563-030-6, 1994.

[53] D. Graff, A. Canavan, and G. Zipperlen, *Switchboard-2 Phase 1.* Linguistic Data Consortium, University of Pennsylvania, ISBN:1-58563-138-8, 1998.

[54] B. L. Pellom and J. H. L. Hansen, "Automatic segmentation of speech recorded in unkonown noisy characteristics," *Speech Communication*, vol. 25, pp. 97–116, August 1998.

[55] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals.* Signal Processing Series. Prentice Hall, 1978.

[56] T. Dutoit, *An Introduction to Text-To-Speech Synthesis.* Kluwer Academic Publishers, 1996.

[57] P. F. Yang and Y. Stylianou, "Real time voice alteration based on linear prediction," in *Proc. of the International Conference on Spoken Language Processing, ICSLP'98*, 1998.

[58] K. K. Paliwal, "Interpolation properties of linear prediction parametric representations," in *Proc. of the European Conference on Speech Communication and Technology, EUROSPEECH'95*, 1995.

[59] F. Itakura, "Line spectrum representation of linear prediction of speech signals," *Journal of Acoustical Society of America*, vol. 57, p. S35, 1975.

[60] B. L. Pellom, *Enhancement, Segmentation, and Synthesis of Speech with Application to Robust Speaker Recognition.* PhD thesis, Duke University, 1998.

[61] K. K. Paliwal and B. S. Atal, "Efficient vector quantization of lpc parameters at 24 bits/frame," *IEEE Transactions on Speech and Audio Processing*, vol. 1, pp. 3–14, January 1993.

[62] A. V. McCree and T. P. B. III, "A mixed excitation lpc vocoder model for low bit rate speech coding," *IEEE Transactions of Acoustics, Speech and Signal Processing*, vol. 3, pp. 242–250, July 1995.

[63] W. P. LeBlanc, B. Bhattacharya, S. A. Mahmoud, and V. Cuperman, "Efficient search and design procedures for robust multi-stage vq of lpc parameters for 4kb/s speech coding," *IEEE Transactions on Speech and Audio Processing*, vol. 1, pp. 373–385, October 1993.

[64] K. K. Paliwal, "A perception-based lsp distance measure for speech recognition," *Journal of Acoustical Society of America*, vol. 84, pp. 14–15, November 1988.

[65] R. P. Cohn and J. S. Collura, "Incorporating perception into lsf quantization - some experiments," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'97*, 1997.

[66] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* John Wiley and Sons, Inc., 1991.

[67] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression.* Kluwer Academic Publishers, 1996.

[68] V. Cherkassky and F. Mulier, *Learning from Data.* John Wiley and Sons, 1998.

[69] D. H. Klatt, "Review of text-to-speech conversion for english," *Journal of Acoustical Society of America*, vol. 82, pp. 737–793, 1987.

[70] A. W. Black, P. Taylor, and R. Caley, *The Festival Speech Synthesis System, System Documentation.* Language Technologies Institute, Carnegie Mellon University and Cepstral, LLC, December 2002.

[71] B. Oskay, Ö. Salor, Ö. Özkan, M. Demirekler, and T. Çiloğlu, "Determining the prosody from written turkish sentences and applications," in *Proc. of the 9th Signal Processing and Communications Applications Conference, SİU'01*, 2001.

[72] T. Kagoshima and M. Akamine, "An f0 contour control codel for totally speaker driven text to speech system," in *Proc. of the International Conference on Spoken Language Processing, ICSLP'98*, 1998.

[73] M. Isogai and H. Mizuno, "A new f0 contour control method based on vector representation of f0 contour," in *Proc. of the European Conference on Speech Communication and Technology, EUROSPEECH'99*, pp. 727–730, 1999.

[74] T. Yoshimura, K. Tohuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling od spectrum, pitch and duration in hmm-based speech synthesis," in *Proc. of the European Conference on Speech Communication and Technology, EUROSPEECH'99*, pp. 2347–2350, 1999.

[75] M. Chu, H. Peng, H. Yang, and E. Chang, "Selecting non-uniform units from a very large corpus for concatenative speech synthesizer," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'01*, pp. 785–788, 2001.

[76] P. A. Taylor, A. Black, and R. Caley, "The architecture of the festival speech synthesis system," in *Proc. of The Third ESCA Workshop in Speech Synthesis*, pp. 147–151, 1998.

[77] Ö. Salor, *Signal Processing Acpects of Text-to-Speech Synthesizer in Turkish*. Master Thesis, Middle East Technical University, Turkey, 1999.

[78] Y. Sagisaka, "Speech synthesis by rule using an optimal selection of non-uniform synthesis units," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'88*, pp. 679–682, 1988.

[79] Y. Sagisaka, N. Kaiki, N. Iwahashi, and K. Mimura, "Atr v-talk speech synthesis system," in *Proc. of the International Conference on Spoken Language Processing, ICSLP'92*, pp. 483–486, 1992.

[80] W. N. Campbell, "Chatr: A high-definition speech re-sequencing system," in *Proc. of Joint Meeting of ASA and ASJ*, pp. 1223–1228, 1996.

[81] E. Moulines and F. Charpentier, "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones," *Speech Communication*, vol. 9, pp. 453–467, 1990.

[82] Y. Stylianou, "Applying the harmonic plus noise model in concatenative speech synthesis," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 1, pp. 21–29, 2001.

[83] A. W. Black and K. A. Lenzo, *Building Synthetic Voices*. Language Technologies Institute, Carnegie Mellon University and Cepstral, LLC, January 2003.

[84] R. K. Dybvig, *The Scheme Programming Language, Ansi Scheme*. Prentice Hall, PRT, 1996.

[85] M. Hunt, D. Zwierynski, and R. Carr, "Issues in high quality lpc analysis and synthesis," in *Proc. of European Conference on Speech Communication and Technology, EUROSPEECH'89*, vol. 2, pp. 348–351, 1989.

[86] W. B. Kleijn and K. K.Paliwal, *Speech Coding and Synthesis*. Elsevier Science B. V., 1995.

[87] A. S3.2-1989(R1995), *American National Standard Method for Measuring the Intelligibility of Speech over Communication Systems*. American National Standards of the Acoustical Society of America, 1989.

VITA

Özgül Salor was born in Ankara, Turkey in 1975. She received her B.Sc. degree with High Honors and M.Sc. degree from Middle East Technical University (METU), Department of Electrical and Electronics Engineering in 1997 and 1999, respectively. She is working towards Ph.D. degree at METU, Department of Electrical and Electronics Engineering since 1999.

She worked as a software engineer at ASELSAN Inc. from June 1996 to August 1998. Then she has joined METU, Department of Electrical and Electronics Engineering, where she is currently employed as a research assistant since August 1998. She has worked as a professional researcher at the Center for Spoken Language Research (CSLR), University of Colorado at Boulder, USA, from September 2001 to February 2003. Her work at CSLR has been supported by TÜBİTAK, the Scientific and Technical Research Council of Turkey, through a combined doctoral scholarship for one year and by CSLR for the rest. Her research was supported by TÜBİTAK, through the combined doctoral scholarship program, from September 2000 to September 2003. Her areas of interest are signal processing, speech synthesis, text-to-speech, and voice conversion.

Her publications are:

International

- **Salor Ö** and Demirekler M, "Spectral Modification for Context-free Voice Conversion Using MELP Speech Coding Framework", IEEE International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004.
- **Salor Ö**, Pellom B and Demirekler M, "Implementation and Evaluation of a Text-to-Speech Synthesis System for Turkish", Proceedings of the 8th European Conference on Speech Communication and Technology, EUROSPEECH, 2003

- **Salor Ö**, Demirekler M and Pellom B, "A System for Voice Conversion Based on Adaptive Filtering and LSF Distance Optimization for Text-to-Speech Synthesis", Proceedings of the 8th European Conference on Speech Communication and Technology, EUROSPEECH, 2003

- Öztürk Ö, **Salor Ö**, Çiloğlu T and Demirekler M, "Duration Modelling for Turkish Text-to-Speech System", 4th International Conference on Language Resources and Evaluation, 2004.

- **Salor Ö**, Pellom B, Çiloğlu T and Demirekler M, "On Developing New Text and Audio Corpora and Speech Recognition Tools for the Turkish Language", Proceedings of IEEE International Conference on Spoken Language Processing, ICSLP, 2002.

National

- **Salor Ö** and Demirekler M, "MELP Konuşma Kodlama Algoritması Kullanarak Konuşma Dönüştürme", Sinyal İşleme ve Uygulamaları Kurultayı, SİU, 2004.

- **Salor Ö**, Demirekler M and Pellom B, "Turkish Text-to-Speech Using the festival Speech Synthesis Engine", Sinyal İşleme ve Uygulamaları Kurultayı, SİU, 2003.

- **Salor Ö**, Demirekler M and Pellom B, "Spectral Training Using Adaptive Filtering for Voice Conversion", Sinyal İşleme ve Uygulamaları Kurultayı, SİU, 2003.

- **Salor Ö**, Pellom B, Çiloğlu T and Demirekler M, "New Corpora and Tools for Turkish Speech Research", Sinyal İşleme ve Uygulamaları Kurultayı, SİU, 2002.

- Özbek Y, Orguner O, **Salor Ö** and Demirekler M, "Ayrıştırılmış Türkçe Kelimelerin Otomatik Bölülenmesi", Sinyal İşleme ve Uygulamaları Kurultayı, SİU, 2002.

- Oskay B, **Salor Ö**, Özkan Ö, Demirekler M and Çiloğlu T, "Türkçe Metinden Ezgi Belirme ve Konuşma Sentezine Uygulaması", Sinyal İşleme ve Uygulamaları Kurultayı, SİU, 2001.

- **Salor Ö**, and Demirekler M, "Sinyal şleme Yöntemleri Kullanarak Türkçe Kelime ve Cümlelerde Ezgi Değişimi", Sinyal İşleme ve Uygulamaları Kurultayı, SİU, 1999.

Master Thesis

- **Salor Ö**, "Signal Processing Aspects of Text-to-Speech Synthesizer in Turkish", Thesis Supervisor: Mübeccel Demirekler, Middle East Technical University, Sept. 1999