A CORPUS-BASED CONCATENATIVE SPEECH SYNTHESIS SYSTEM FOR TURKISH

by

Haşim Sak

B.S., in Computer Engineering and Information Science, Bilkent University, 2000

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering
Boğaziçi University
2004

A CORPUS-BASED CONCATENATIVE SPEECH SYNTHESIS SYSTEM FOR TURKISH

APPROVED BY:

Assist. Prof. Tunga Güngör . . . .
(Thesis Supervisor)

Assoc. Prof. Levent Arslan . . . .
(Thesis Co-supervisor)

Prof. Lale Akarun . . . .

Prof. Fikret Gürgen . . . .

Dr. Yaşar Safkan . . . .

DATE OF APPROVAL: 16.6.2004

ACKNOWLEDGEMENTS

I would like to express my gratitude to my thesis supervisors Assist. Prof. Dr. Tunga Güngör and Assoc. Prof. Dr. Levent Arslan for their guidance and support in this work. I appreciate Assist. Prof. Tunga Güngör very much for being so helpful, understanding and patient to me. I am very grateful to Assoc. Prof. Levent Arslan for introducing me to the field of speech processing and as my teacher in this field.

I would like to thank all of my colleagues at work for their helps and understandings to me. It is a pleasure for me to work with them.

It is very nice to have such brothers like that I have. Without the support of my brothers, Ergün, Halis and Ersin, I could not certainly accomplish this work.

I would also like to thank all of my friends namely Ali, Cüneyt, Garo, Nage, Onur, Umut and Rana, who were kind enough to help me carry out the listening tests.

Prof. Lale Akarun and Prof. Fikret Gürgen also deserve my thanks for kindly accepting being in my thesis committee.

Special thanks go to my dear friend Dr. Yaşar Safkan for teaching me and for his invaluable contributions in this thesis. I am very much grateful to him.

ABSTRACT

A CORPUS-BASED CONCATENATIVE SPEECH SYNTHESIS SYSTEM FOR TURKISH

Speech synthesis (text-to-speech) is the process of converting the written text into machine generated synthetic speech. Concatenative speech synthesis systems render speech by concatenating pre-recorded speech units. Corpus-based methods (unit selection) use a large inventory to select the units and concatenate. This thesis is part of an effort to design and develop an intelligible and natural sounding corpus-based concatenative speech synthesis system for Turkish. The implemented system contains a relatively simple front-end comprised of text analysis, phonetic analysis, and optional use of transplanted prosody. The unit selection algorithm is based on commonly used Viterbi decoding algorithm of the best path in the network of the units. The back-end is the speech waveform generation based on the harmonic coding of speech and overlap-and-add mechanism. In this work, the different unit sizes such as syllables, phones and half-phones have been experimented with. Speech corpus design and recording script preparation methods have been explained. A speech model based on harmonic coding of speech has been developed for speech representation and waveform generation. The harmonic coding has enabled us to compress the unit inventory size by a factor of three. A Viterbi decoding algorithm using spectral discontinuity cost and prosodic mismatch objective cost measures has been implemented. A Turkish phoneme set has been designed. Text-to-phoneme conversion for Turkish has been worked on, and a root words pronunciation lexicon has been constructed. A simple text normalization module has been implemented. The importance of prosody in unit selection has been studied by using transplanted prosody vs no synthetic prosody modeling in unit selection. Subjective tests have been carried out for evaluating the synthesized speech quality. The final Turkish speech synthesis system got 4.2 MOS like score in the listening tests.

ÖZET

TÜRKÇE İÇİN KORPUS TABANLI BİRLEŞTİRMELİ KONUŞMA SENTEZLEME SİSTEMİ

Konuşma sentezi yazılı metnin makine tarafından üretilmiş sentetik konuşmaya çevrilmesi işlemidir. Birleştirmeli konuşma sentezleme sistemleri sentezlemeyi daha önceden kaydedilmiş ses parçalarını birleştirerek yapar. Korpus tabanlı metotlar (parça seçme) birleştirilecek ses parçalarını seçmek için geniş bir ses parçası veritabanı kullanırlar. Bu tez kulağa doğal insan sesi gibi gelen, anlaşılabilir korpus tabanlı birleştirmeli bir konuşma sentezleme sistemi geliştirmek için harcadığımız emeğin bir sonucudur. Tasarlanan sistem metin normalizasyonu, metin analizi ve isteğe bağlı kullanılan nakledilen vurgu ön birimlerini içerir. Parça seçme algoritması veritabanındaki parçaların oluşturduğu ağda Viterbi algoritması ile en iyi patikanın bulunmasına dayanır. Arka uç harmonik kodlama ses modeli ve üst üste getirip ekleme yöntemini kullanarak ses dalga formunu oluşturur. Bu çalışmada farklı parça büyüklükleri, örneğin heceler, fonemler ve yarım fonemler denenmiştir. Konuşma korpusu tasarımı ve kayıt metinlerinin seçilmesinde kullanılan metotlar açıklanmıştır. Sesi modellemek ve ses dalgası oluşturmak için harmonik kodlama yöntemine dayanan bir ses modeli geliştirilmiştir. Harmonik kodlama, ses veritabanını 3 kat sıkıştırmayı sağlamıştır. Parça seçmede spektral süreksizlik ve vurgusal uyumsuzluk objektif maliyet ölçekleri kullanan Viterbi algoritması yazılmıştır. Türkçe fonem seti oluşturulmuştur. Türkçe için metinden foneme çevrim üzerinde çalışılmış ve de kök kelimelerin okunuşlarını içeren bir sözlük hazırlanmıştır. Basit bir metin normalizasyon modülü yazılmıştır. Parça seçmede vurgunun önemini araştırmak için nakledilen vurgu kullanan ve vurgu modeli kullanmayan sistemler karşılaştırılmıştır. Sentetik konuşma kalitesini değerlendirmek için öznel dinleme testleri yapılmıştır. Sonuç olarak MOS benzeri bir derecelendirmede 4.2 puan alan bir Türkçe konuşma sentezleme sistemi geliştirilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AbS	Analysis-by-Synthesis
AR	Auto Regressive
ATN	Augmented Transition Network
C	Consonant
CPU	Central Processing Unit
DFT	Discrete Fourier Transformation
DRT	Diagnostic Rhyme Test
F_0	Fundamental Frequency
FFT	Fast Fourier Transformation
FST	Finite State Transducer
HMM	Hidden Markov Model
HNM	Harmonic-plus-Noise Modeling
IPA	International Phonetic Alphabet
IVR	Interactive Voice Response
LPC	Linear Predictive Coding
MFCC	Mel Frequency Cepstral Coefficients
MOS	Mean Opinion Score
POS	Part-of-Speech
RELP	Residual Excited Linear Prediction
TTS	Text-to-Speech
V	Vowel
/x/	Phoneme x

1. INTRODUCTION

Speech synthesis or text-to-speech (TTS) is the computer automated conversion of raw or tagged input text into audible, intelligible speech. The use of synthetic speech is applicable and desirable for some real world situations and applications such as in human-machine interaction, hands and eyes free access of information, interactive voice response systems (IVR), screen reader software for the visually handicapped and in other applications where the text in digital form is available and speech correspondence is required. TTS can also be considered as a speech coding system that achieves superior compression ratios than what is possible with waveform coders and vocoders [1]. TTS has also been used to assist in language learning. The animated agents have benefited from TTS since the correspondence between the text and the synthesized utterance enables the visual characters to mimic and gesture accordingly. Where the input text changes frequently and the domain is not limited, the use of TTS provides great flexibility instead of using pre-recorded speech, since recording all messages is not feasible in these situations.

The generation of synthetic voice that imitates human speech from plain text is not a trivial task, since this generally requires great knowledge about the real world, the language, the context where the text comes from, a deep understanding of the semantics of the text content and the relations that underlie all these information. However, many research and commercial speech synthesis systems developed have contributed to our understanding of all these phenomena, and have been successful in various respective ways for many applications.

There have been three major approaches to speech synthesis: articulatory, formant and concatenative [1, 2, 3, 4]. Articulatory synthesis tries to model the human articulatory system, i.e. vocal cords, vocal tract, etc. Formant synthesis employs some set of rules to synthesize speech using the formants that are the resonance frequencies of the vocal tract. Since the formants constitute the main frequencies that make sounds distinct, speech is synthesized using these estimated frequencies. Concatenative speech synthesis is based on the idea of concatenating pre-recorded speech units to construct the utterance. Concatenative systems tend to be more natural than other two since original speech recordings are used instead of models and parameters. In concatenative systems, speech units can be fixed-size diphones or variable length units such as syllables and phones. The later systems are known as unit selection, since a large speech corpus containing more than one instance of a unit is recorded and variable length units are selected based on some estimated objective measure to optimize the synthetic speech quality.

The diphone-based synthesis produces unnatural robot-sounding speech due to the use of fixed number of speech units namely diphones that are sound-to-sound transitions. The unit selection methods have been developed to overcome this problem by storing more than one instance for each unit and provide better speech quality, since using more units provides prosodic and acoustic variability found in natural speech [3, 4, 5, 6]. Speech synthesis using more than one instance for each unit requires a unit selection algorithm to choose the appropriate units to concatenate. The unit selection algorithms have been generally based on dynamic programming. The input text to be synthesized is first converted to a target specification that includes generally phonemes and related information such as energy, duration and pitch for each phoneme. To choose the units from the unit inventory that best fit in this specification, the units from the unit inventory can be considered as constructing a network of nodes. Each node has the target cost that is the cost of using that specific unit for the realization of the target unit in the specification. The links between the units have the concatenation cost that is an estimate of the cost of joining two units. The join cost is zero for two units that naturally occurs adjacent in the speech database. The Viterbi decoding algorithm is used to find the best path that is the path with the least total of all target and concatenation costs over the path. The speech units on the path with the minimal total cost is used to synthesize the speech waveform. Concatenating the speech waveforms results in some glitches at the concatenation points in the synthesized utterance. Therefore, to ensure smooth concatenation of speech waveforms and to enable prosodic modifications on the speech units a speech model is generally used for speech representation and waveform generation [7].

In this thesis, we describe our efforts to develop a corpus-based concatenative speech synthesis system for Turkish. The next chapter gives an overview of the research on speech synthesis based on unit selection. System architecture chapter describes our system design and shows and explains components and the interactions between components. Speech corpus chapter explains the methods used to prepare the speech corpus and the recording scripts. Text-to-Speech front-end chapter explains the system components: text analysis, phonetic analysis and prosody generation that constitute the front-end of the synthesis system. The chapter named unit selection describes the methods and algorithms employed to solve the issues on selecting appropriate units to concatenate. Waveform generation part describes our implementation of the harmonic sinusoidal coding to be used for speech representation and waveform generation model. In the experiments we talk about the methods that we tried in developing the final system. The quality assessment describes the subjective tests that we conducted to evaluate the system performance and discusses the test results. In conclusion, we give a summary of the work done and the results obtained.

1.1. Problem Statement

Corpus-based concatenative speech synthesis, also called unit selection, has emerged as a promising methodology to solve the problems with the fixed-size unit inventory synthesis, e.g., diphone synthesis [3, 4, 8]. Using a fixed-size unit inventory requires making unit concatenations at each unit join; as a result the output speech quality is degraded. The prosodic modification of the units is also needed since limited number of units exist in the inventory. These signal modifications further degrade speech quality and result in unnatural synthetic speech.

In corpus-based systems, the acoustic units of varying sizes are selected and concatenated from a large speech corpus. The speech corpus contains more than one instance of each unit to capture prosodic and spectral variability found in natural speech; hence the signal modifications needed on the selected units are minimized if an appropriate unit is found in the unit inventory. The use of more than one instance of each unit requires a unit selection algorithm to choose the units from the inventory that matches the best the target specification of the input sequence of units. The unit selection algorithm favors choosing consecutive speech segments from the speech unit inventory to minimize the number of joins.

The output speech quality of unit selection in terms of naturalness is much better than fixed-size unit inventory synthesis. However the unit selection presents some challenges to speech synthesis. The speech quality is good most of the time, but the quality is not consistent. If the unit selection algorithm fails to find a good match for a target unit from the inventory, the selected unit is needed to undergo some prosodic modifications which degrade the speech quality at this segment join. Some systems even choose not to do any signal modifications on the selected units [5]. To ensure a consistent quality, a good speech corpus design that covers all the prosodic and acoustic variations of the units that can be found in an utterance has to be addressed. It is not feasible to record larger and larger databases given the complexity and combinatorics of the language, instead we need to find a way for optimal coverage of the language [9].

The unit selection algorithm should also do a good job of selecting the appropriate units, to ensure high quality speech synthesis. But this is not an easy task since the unit selection algorithm has to use some objective measures to decide which units to concatenate given the target specification of the unit sequence formed from the input utterance to be synthesized.

The construction of the correct target specification for the unit sequence from the input text is also very important since unit selection will be based on this specification. The specification generally contains the identity of the units used in the system such as phonemes, the phonetic context of the units and prosody prediction for the units. The prosody prediction requires the intonation modeling (pitch and accent (stress) prediction), phrase boundary detection and duration modeling (timing), which are not easy to correctly predict from raw text.

The waveform generation should also be done in a way to smooth the discontinuities around the unit concatenations. The discontinuities are due to spectral, phase, pitch period, energy and formant frequency mismatches. A speech representation model has to be used to ensure a smooth concatenation at unit boundaries.

Evaluating the speech synthesis quality is also a difficult task for speech synthesis systems. The synthetic speech quality is dependent on many factors and it is hard to evaluate the performances of system components in isolation. The complex interaction between different system parts makes it difficult to tune the system performance using quality assessment methods.

1.2. Motivation and Objective

The emergence of corpus-based methodologies enabled the development of high quality speech synthesis systems for languages such as English [8, 10, 11]. In this study, our aim is designing and developing a natural sounding Turkish text-to-speech synthesis system and framework using corpus-based concatenative speech synthesis methodology. We will develop an open system architecture that will be easy to improve the functionalities of the modules. The interaction between modules will be well defined for easy operation and integration. The unit selection techniques will be investigated for Turkish. The agglutinative nature of Turkish will be dealt with. The speech corpus design issues will be explored. A unit selection algorithm based on Viterbi decoding will be implemented. Acoustic and prosodic features used in cost calculation in unit selection will be extracted. The effect of using transplanted prosody on the voice quality will be explored. Harmonic sinusoidal coding for representing and concatenating speech waveforms will be studied for Turkish. Subjective tests will be carried out to assess the output speech quality.

2. LITERATURE SURVEY

In recent years, the research in speech synthesis has focused on corpus-based concatenative methods. The advent of faster CPU's and mass storage spaces with decreasing costs have contributed to the applicability of the use of large speech corpus and unit selection algorithms. With the concatenation of longer speech segments that naturally occur adjacent in the speech database, the synthesized voice quality has improved in terms of intelligibility, pleasantness and naturalness compared to the other methods, e.g., formant synthesis and diphone synthesis.

ATR v-Talk speech synthesis system developed at ATR labs introduced the unit selection approach from a large speech database [3]. The selection of units was based on minimizing an accoustic distance measure between the selected units and target spectrum. The prosodic features like duration and intonation have been added to the target specification to choose more appropriate units in terms of prosody in CHATR speech synthesis system [4]. A. Hunt and A. Black have contributed to the area the idea of applying Viterbi decoding of best path algorithm for unit selection [6]. The unit selection process as they realized has many similarities with a best path decoding algorithm commonly applied in HMM-based speech recognition systems. In their system, the units in the speech database are considered as a network of nodes and transitions. The speech synthesis or unit selection in this system corresponds to selecting the units on a path in this network that matches best the sequence of target specifications of units derived from input text. In the network of units the cost of selecting a unit corresponding to a target specification is called the target cost and is calculated as an estimate of distance between a database unit and the target. The concatenation cost on the other hand is an estimate of cost to concatenate two database units. By constructing a state transition network with these costs, a Viterbi search is carried out by using pruning techniques based on phonetic context similarity to the target, the target cost and the concatenation cost. They have also proposed two training algorithms for determining the weights used in target and concatenation cost estimation. One is based on selecting a set of weights that give an overall minimal cepstral difference between a set of utterances synthesized and the actual spoken correspondents. The other one uses linear regression to estimate the weights by predicting the distance between units and their n-best matches in the database in terms of acoustic difference by a linear weighting of the costs.

Finite-state transducer (FST) based approach to unit selection has also been studied [12]. In that work, the unit selection architecture is represented as a composition of FST components each tailored to a specific purpose such as lexicon, pronunciations and waveform generation.

For synthesis back-end, various speech representation models and waveform generation methods have been devised [1, 7]. Using a speech model to represent speech waveform is required since it enables us to concatenate units smoothly and it may facilitate compressing the large speech databases. Linear predictive coding (LPC) as an analysis-by-synthesis (AbS) speech model tries to model the speech by a set of linear coefficients [1]. LPC is actually an auto-regressive (AR) filter. The filter coefficients can be in a number of ways including auto-correlation method. Some variations of LPC such as residual excited linear prediction (RELP) have also been employed for better quality speech modeling. The harmonic sinusoidal coding tries to represent the speech by sum of the sinusoids that are harmonically related [13].

The Next-Gen speech synthesis system developed at the AT&T labs is one of the commercial systems that use unit selection [8]. The system unifies the best chosen approaches from AT&T Flextalk, the Festival and CHATR system. The front-end, i.e. the text and linguistic analysis and prosody generation is from Flextalk, the unit selection is a modified version of CHATR and the framework for all these is borrowed from the Festival. As an improvement to the CHATR unit selection, the system uses half phones compared to phonemes as the basic speech units [14]. This allows phoneme or diphone concatenation at a unit boundary. For the back-end, a Harmonic plus Noise Model (HNM) representation of the speech has been developed [7]. HNM considers the speech is composed of a sum of harmonic sinusoids plus a noise component. These two components are separated in the frequency domain as a low band harmonic part and a high band noise part. The calculation of the amplitudes and phases of the harmonics is reduced to minimizing a least-squares criterion between the original signal and the harmonic sinusoidal approximation to the signal. The noise part is modeled as an AR filter excited by Gaussian noise. The phase mismatches at the unit boundaries are eliminated by an offline process that updates the unit boundaries using the center-of-gravity approach [15]. Also for smooth concatenation at unit boundaries, a linear interpolation of HNM parameters for the joining units is done. The output speech waveform is synthesized using the overlap-and-add process.

To reduce the runtime complexity of the unit selection, some algorithms for preselection of units have been proposed [16]. The first method proposed is preselection filtering which uses preselection of n best units for each synthesis context of five phones in the unit database. Then in the selection of the candidate units for a particular phone in a triphone context, all contexts of five phones that cover the triphone sequence are taken. The second method uses a synthesis-based preselection. In the analysis part, a huge sentence database is synthesized storing to an inventory file which units are chosen for each triphone sequence. Then for a synthesis request, only the units for the triphone sequences that have been previously selected and recorded in the inventory are used. A speed up factor of 10 without the loss of speech quality in the unit selection is reported.

Unit selection based concatenative speech synthesis approach has also been used in the IBM Trainable Speech Synthesis System [11, 17]. The system uses the Hidden Markov Models (HMMs) to phonetically label the recorded speech corpus and aligns HMM states to the data. The units used in the unit selection process are HMM state sized speech segments. The unit selection is a dynamic programming based search, which uses decision trees to facilitate the choice of appropriate units, with a cost function to optimize. The segments in the speech database is coded into mel frequency cepstrum coefficients (MFCCs). A unit pre-selection algorithm that discards some of the units is used to reduce the database size and to improve the runtime performance of the system. Prosody generation is carried out by a rule-based front-end.

Weighted finite-state transducers have been successfully applied to the multilingual text analysis at Bell Labs [18]. The lexicon, morphological rules for morphological analysis, language model rules for phrasal accentuation and prosodic phrasing, rules for expansion of numbers in digits and abbreviations, and phonological rules for determining the pronunciation of words are all represented by weighted FSTs.

Duration analysis and modeling of Turkish has been recently studied [19]. In that work, first duration analysis of Turkish phonemes are given and the factors that are found to be effective in governing timing in Turkish utterances are extracted. Then four duration models namely mean durations of the phonemes, mean durations of the triphones, tree-based modeling of triphone durations and a linear model all trained from phonetically labeled speech corpus are implemented. The linear additive and triphone tree models are found to be superior than others.

Intonation and stress characteristics in Turkish sentences have also been investigated [20]. The fundamental frequency contours for sentences in a single speaker speech database have been analyzed and a fundamental frequency contour generation system has been presented. The system is based on a template sentence pitch contour database that has been constructed using sentence type, syntactic structures of sentences using part-of-speech (POS) tagging and word stress. The input sentence is matched with one of the templates and the pitch contour of the selected template is used for predicting the pitch contour of the input sentence.

In a study to determine what is more important in a concatenative speech synthesis system, the effect of pitch, duration and acoustics is experimentally evaluated using Microsoft Whistler TTS engine [21]. It is found that synthetic pitch degrades the speech quality most, while a simple look-up table for phoneme durations does a good job.

Turkish is an agglutinative language and uses affixes that are mostly suffixes with a few exceptional prefixes. A TTS system for Turkish requires use of a morphological analysis component for pronunciation lexicons and linguistic analysis. For morphological analysis of Turkish, augmented transition networks (ATNs) [22] and finite-state techniques [23] have been applied. A pronunciation lexicon for Turkish has been recently developed using finite-state techniques [24].

Speaker selection is also a critical decision in terms of synthesis quality. According to [8], the speaker selection has up to 0.3 MOS score effect on the output synthesis quality. There have been some work on evaluating the speaker voice appropriateness to TTS [8].

3. SYSTEM ARCHITECTURE

The corpus-based concatenative Turkish speech synthesis system architecture is shown in Figure 3.1. The system components shown are common in most of the speech synthesis systems using unit selection. The system can be mainly divided into three parts: the front-end, the unit selection and the back-end. The front-end is responsible for producing an internal linguistic and prosodic description from the input text to be used for unit selection. This description is fed into the unit selection as the target specification. The unit selection uses this specification to choose the units from the speech database that minimize a cost function between the specification and the units from the speech unit inventory. The waveforms for the selected units are concatenated in the back-end. The smoothing of concatenation points are also handled in the back-end.

The outputs from system components can also be seen in Figure 3.1. The system uses a well-defined internal data structure to store the information for the text to be synthesized. This structure is communicated between components and each component appends extracted information using the already existing information in the structure and task specific algorithms. This enables each system component to be developed independently and makes it flexible to improve the functionalities of each component separately if required.

The system components and the responsibilities are briefly described below. The following sections explain the system components in detail.

Turkish Pronunciation Lexicon: The lexicon (dictionary)is a collection of words and their pronunciations and any other word level information. The text analysis component uses the lexicon for determining the pronunciation of the words. The Turkish

Figure 3.1. Corpus-based concatenative Turkish speech synthesis system architecture

lexicon that we prepared is a root word pronunciation lexicon. The lexicon may be expanded to contain any information about a word if the other components need it.

Speech Unit Inventory: The speech recordings from a speaker are processed to construct an inventory of speech units (speech corpus). The construction of the speech inventory is an offline process. The unit inventory stores the waveforms for the units, phone identities of the units, phonetic context and prosodic annotations for the units. The waveforms in the speech corpus have been compressed for efficient storage of units using a harmonic sinusoidal coding method.

Text Analysis: Text analysis component is responsible for converting the input text into an internal linguistic description. Text normalization and sentence breaking is the sub-processes carried out in this component. The input to this component is raw text and the output is an internal data structure that is a linked list of sentences which are also a linked list of words that have the normalized forms. This text structure contains room for the information that other components will gather and is communicated through components.

Phonetic Analysis: Phonetic analysis refers to the conversion of the linguistic description in orthographic form to phonemes. This component is responsible for grapheme-to-phoneme conversion for Turkish. The input is the text structure from the previous component and the pronunciations for the words normalized are appended to this structure.

Prosodic Analysis: Prosody analysis annotates the internal linguistic description with prosodic features i.e. the pitch, energy and duration for each phoneme to produce a target specification of the input text to be used in unit selection. The current implementation of the prosody analysis component does not include any synthetic prosody generation and uses only transplanted prosody that is the real prosody from the original speech recordings.

Unit Selection: The selection of the appropriate units from the speech corpus that contains thousands of units is dealt with here. The unit selection algorithm that we used is based on commonly used dynamic programming. The input to this component is the internal text structure annotated with phonetic and prosodic features that is called target specification and the output is the sequence of the units selected from the speech inventory.

Waveform Generation: The unit speech waveforms need to be concatenated smoothly to generate the synthesized speech waveform. Since the waveforms of two joining units may come from different utterances, the concatenation points may sound some glitches caused by acoustic and prosodic mismatches in the units. The parametric encoding of speech waveforms for more natural concatenation and easy speech modifications are the methods employed here.

4. SPEECH CORPUS DESIGN

The speech corpus used for testing the algorithms in this thesis is a female speaker speech database. It is commercially owned by the company named GVZ that develops Turkish speech related technologies. The speech corpus contains about 20 hours of speech recorded by a professional speaker covering about 30000 Turkish phrases. The speech corpus has been divided into two sets namely training and test set. The test set contains about 1000 phrases used for the purpose of evaluating the synthesis quality. The remaining recordings have been used to construct two speech unit inventories of different sizes. One of the speech inventories uses all the recordings from the training set and contains about 19 hours of speech recording. The other speech unit inventory has been confined to contain 5000 phrases corresponding to about 3 hours of speech. Use of two training sets of different sizes enables us to evaluate the relative performance of algorithms on different sized speech corpora. We also want to see the effect of speech database size on the output speech quality. The use of smaller speech database during system development is also beneficial and justified by being able to see the effects of our works on speech quality more easily and faster and easier modifications on small databases. We have in final system compared the MOS scores from the listening tests for the two speech inventory in the section named quality assessment.

The unit inventory design has a major effect on speech quality. In unit selection finding units that match best the target specification is more probable with a large number of units in a carefully designed database since we can capture much more prosodic and acoustic variability for the phones. The speech quality is severely degraded when an appropriate unit can not be found. The prosodic modifications of the selected units also degrade the speech quality so it should be done minimally, such a thing is possible only when we find a close match.

The recording scripts have been constructed by a Greedy algorithm that tries to choose sentences based on their phonetic context. The recording script contains phrases or word groups rather than full sentences to prevent common repeating of some words that makes the database size bigger while adding to overall synthesis quality little. The phrases have been collected from online Turkish text material and have been preprocessed to break into phrases by using punctuation marks. They have been checked manually to ensure the phrases are complete and well formed otherwise eliminated. Then a greedy algorithm has been used to select the phrase with greatest score calculated as the total frequency of the triphone contexts found in the phrase normalized by the number of triphones to ensure the short ones are promoted. The algorithm updates the frequencies of selected phrase triphones to zero and runs again on the remaining phrases. The algorithm produced 30000 phrases. These phrases have been recorded by a professional female speaker in a sound isolated room in multiple sessions. The recordings then have been phonetically auto-labeled by using GVZ's speech recognition engine. The phoneme boundaries have been manually corrected by hand. There has been no prosodic labeling on speech data.

To choose the subset of speech corpus that we used to construct 5000 phrase sized speech unit inventory we have used a Greedy algorithm similar to the one used in constructing recording scripts. It tries to choose the syllables that have not been yet covered in the selected phrases. In Turkish, syllables have a prosodic integrity in themselves. We can categorize syllables in Turkish as having the patterns V, VC, VCC, CV, CVC, and CVCC where C designates consonant and V vowel phonemes. We have also considered syllable boundaries, sentence start and end, and word boundaries. Using the greedy algorithm we choose the subset of the speech corpus that covers all variations of these patterns. The algorithm chooses 5000 phrases from the speech corpus having 30000 phrases.

The phonetically labeled speech corpus is converted to a binary indexed speech unit inventory file to be used in the unit selection process. The speech inventory contains the unit identities, the phonetic context information, the prosodic features namely energy, duration and pitch of units which are automatically extracted from the speech waveform and the speech waveforms of the units. Since in waveform generation we use a harmonic coding based speech model, the speech waveforms are stored with the parameters of this model. Using parametric coding of speech enables us to compress the speech unit inventory about three times with a slight effect on the output speech quality. Since for unit selection we use units of size half-phone, the phone sized units in the labeled speech are converted automatically to half-phone sized units. The half-phone unit boundaries are aligned to the frame boundaries of the parametric harmonic speech model.

5. FRONT-END

The front-end of our speech synthesis system is responsible for producing an internal linguistic and prosodic description from the input text. This description is input to the unit selection as the target specification for the utterance to be synthesized. In this thesis, the functionalities of the front-end have been kept at minimal and we concentrated our efforts on unit selection and waveform generation. The following subsections describe the Turkish phoneme set used and the subsystem components in the front-end.

5.1. Turkish Phoneme Set

Phonemes are the smallest units of speech sound in a language that can serve to distinguish one word from another [1]. Turkish alphabet has 29 letters classified as 8 vowels (a, e, ı, i, o, ö, u, ü) and 21 consonants (b, c, ç, d, f, g, ğ, h, j, k, l, m, n, p, r, s, ş, t, v, y, z). However, Turkish orthography can not represent all the sounds in Turkish. In our system for phonetic transcriptions we adopted a new phoneme set based on the SAMPA standard. The SAMPA identifies 8 vowels and 24 consonants (excluding two consonantal allophones, /w, N/)for representing Turkish sounds and designates a length mark /:/ to represent the lengthening of some vowels in loanwords in Turkish. Based on the SAMPA phoneme set for Turkish, we adopted a new phoneme set as shown in Tables 5.1 and 5.2 with example words and corresponding SAMPA phonemes. The new phoneme set designates new symbols for some of the SAMPA phonemes and introduces three more phonemes, /öo, üu, ea/ corresponding to allophones of the phonemes /o, u, a/, respectively.

5.2. Lexicon

For phonetic analysis and text analysis, a Turkish lexicon has been constructed. The lexicon is used for phonetic pronunciations, abbreviation and acronym expansion. The lexicon contains approximately 3500 entries for the words in root form and their

Table 5.1. Turkish phoneme set

Phonemes	**Example Words**	**SAMPA**
a	**a***şk*	a
b	**b***ugün*	b
c	**c***uma*	dZ
ç	**ç***amur*	tS
d	**d***ünya*	d
e	**e***vet*	e
f	**f***utbol*	f
g	**g***ece*	gj
ğ	*do***ğ***a*	G
h	**h***ayat*	h
ı	**ı***şık*	1
i	**i***nsan*	i
j	**j***üri*	Z
k	**k***ader*	c
l	**l***ider*	l
m	**m***avi*	m
n	**n***isan*	n
o	**o***yun*	o
ö	**ö***zgürlük*	2
p	**p***ara*	p
r	**r***enk*	r
s	**s***es*	s
ş	**ş***ans*	S
t	**t***at*	t
u	**u***yku*	u
ü	**ü***lke*	y
v	**v***eda*	v

Table 5.2. Turkish phoneme set(cont.)

Phonemes	**Example Words**	**SAMPA**
y	**y***eni*	j
z	**z***aman*	z
aa	***a****lim*	a:
öo	*alk****o****l*	
üu	*sük****u****net*	
uu	*kan****u****nen*	u:
ii	*mill***i**	i:
ea	*dikk****a****t*	
ee	*m****e****mur*	e:
gg	**g***aga*	g
kk	*a****k****ıl*	k
ll	*a****l****kış*	5

corresponding pronunciations. The lexicon currently contains only pronunciations for the words. The small size of the lexicon is the result of the Turkish's relatively simple pronunciation schema compared to English. In Turkish, pronunciations of many words can be derived easily by one-to-one mapping of letters to phonemes. The exception is the words that have been borrowed from other languages such as Arabic and Persian. The entries in the lexicon are mostly these kinds of words. Such a word from the lexicon is *fedakarlık* which has the pronunciation */f e d aa k ea r l ı k/* using our conventional phoneme set. The fourth letter /a/ has a standard pronunciation of phoneme /a/, however in this word this letter is pronounced much longer than the standard /a/ phoneme, so our phoneme set has introduced the /aa/ phoneme that sounds similar to /a/ phoneme with a longer duration. This sound has contributed to Turkish from Arabic.

5.3. Text Analysis

The input text to the speech synthesis system needs to be processed and converted to a linguistic representation which should be in a suitable form for the subsequent operations to work on. The text analysis component in a TTS system generally includes text normalization, document structure detection and linguistic analysis subcomponents. In this work a relatively simple text analysis is done.

The input text is first parsed into sentences and words using white space characters and punctuation marks and stored in an internal text data structure that is a linked list of sentence structure that is also a linked list of word structure. The sentence structure has been designed to store sentence level information such as sentence type and word structure has been designed to store word level information such as POS tagging and word pronunciation. No linguistic analysis on syntax and semantics is done. However a simple text normalization has been included in this study. Text normalization component converts nonorthographic symbols into orthographic symbols. The abbreviations and acronyms are expanded and digit sequences are converted to word forms. The characters that can not be converted to speech are discarded. The punctuation marks are preserved.

5.4. Phonetic Analysis

Turkish is a phonetic language meaning that a simple grapheme-to-phoneme conversion is possible for most of the words due to close relationship with Turkish orthography and phonology. Phonetic analysis component converts text in orthographic (written) form to phonemes. However there are some exceptional words that are mostly loanwords. The Turkish pronunciation lexicon is mainly used for determining the pronunciations of these words. There are also cases where one-to-one mapping is not possible such as vowel lengthening and palatalization in pronunciations of some suffixes depending on vowel harmony as described in [24]. A complete phonetic analysis for Turkish requires use of morphological analysis and finite-state techniques and such a system is described in [24].

For grapheme-to-phoneme (letter-to-sound) conversion, we developed a simpler system that gives satisfactory results for most of the words. Turkish is an agglutinative language, therefore for text to phoneme conversion, a morphological parser implementation for Turkish based on augmented transition networks has been used[22]. Morphological parser is used to separate the root word and suffixes. The root word pronunciations are looked up in the pronunciation lexicon. The pronunciations of suffixes are found by a direct mapping of letters to the phonemes in the phoneme set. The root word pronunciation lexicon contains about 3400 word-pronunciation pairs. The pronunciations of other root words that are not in the lexicon are found by a direct mapping as in the case of suffixes.

5.5. Prosodic Analysis

The system has been designed to use a prosodic analysis component. However, the current work does not contain a prosody generation module implementation. We are planning to add pitch contour synthesis and duration modeling. To evaluate the efficiency of using prosodic analysis we tailored the system to optionally use the transplanted prosody from the original speech utterances. Prosody generation module can provide pitch, duration, and energy information for an original speech utterance which can be used as the target specification in the unit selection process to synthesize the input text. This method has been used in quality assessments to see the effect of real prosody on the output speech quality. Test results can be seen in quality assessment section.

6. UNIT SELECTION

The output of the front-end of our system is a target unit sequence corresponding to the input text to be synthesized. The target specification is a sequence of phonemes each having target energy, pitch and duration values. The speech corpus has also been processed to construct a unit inventory storing the phonemes with the same prosodic features as target sequence and the phoneme context information. Since we use a large speech database, there is more than one instance for each phoneme, each possibly having different phonetic context and prosodic and acoustic realizations. Therefore for the target specification we have a large number of choices from the unit inventory. In concatenative speech synthesis, choosing the right units is very important for the quality of the synthesized voice. An appropriate selection of units may also allow to get rid of prosodic modifications of the selected units which generally degrade the output speech quality. The unit selection module tries to choose the optimal set of units from the unit inventory that best matches the target specification. The algorithm that we employed is based on the unit selection algorithm first applied in CHATR speech synthesis system [6]. The following sections describe the unit size that we used in unit selection and the unit selection algorithm.

6.1. Unit Size

In unit selection, units of differing sizes have been used in literature. The syllable, phoneme, diphone, halfphone, HMM state-sized units have all been used in various systems [6, 8, 11]. The unit selection in CHATR uses phonemes as the speech units. The diphones that are the speech units from the second half of a phone to the first half of the following phone, are used in diphone-based concatenative synthesis. The diphone synthesis produces highly intelligible speech due to the fact that diphones capture some of the coarticulation effect at the phone boundaries. The AT&T Next-Gen speech synthesis system uses halfphones for the basic speech units. Since the halfphones provide the flexibility of using phonemes or diphones or a mixture of both for the speech segments, we also adopted the halfphones as basic speech units. We have done some experiments to empirically decide the relative costs of using halfhones and diphones.

6.2. Decoding Using Viterbi Search

Optimal unit selection given the target specification from the unit inventory resembles the best-path decoding algorithm commonly used in HMM-based speech recognizers [6]. The speech unit inventory is analogous to the grammar network in HMM-based recognizers and can be considered as a state-transition network. The best-path decoding of the words in the grammar is very similar to determining optimal unit sequence in the network of units. The transition cost in speech recognizers corresponds to the concatenation cost in unit selection. The target cost used in unit selection corresponds to HMM state observation cost. This analogy guides us to the use of dynamic programming to find the optimal unit sequence. The pruned Viterbi search algorithm commonly used in HMM-based speech recognizers can be easily adopted to the problem of the unit selection. The algorithm that we used in unit selection is a Viterbi best path decoding algorithm that is very similar to the one used in CHATR speech synthesis system, and is described below using the notation from [6].

The unit selection algorithm can be stated as given the target specification $t_1^n = (t_1, ..., t_n)$ finding the unit sequence $u_1^n = (u_1, ..., u_n)$ that optimizes a cost function of the distance between the target specification and the unit sequence. In unit selection, there are two kinds of cost function, namely target cost and concatenation cost as shown in Figure 6.1. Target cost, also called unit cost, is an estimate of the cost of using a selected unit in place of the target specification of that unit. This cost is a measure of how well the unit from the unit inventory suits the corresponding target unit in the specification. This cost can be calculated as a weighted sum of the target sub-costs as follows:

$$C^t(t_i, u_i) = \sum_{j=1}^{P} w_j^t C_j^t(t_i, u_i)$$

Figure 6.1. Target and concatenation cost in unit selection

where P is the number of target sub-costs and w_j^t is the corresponding weights.

The concatenation or join cost is an estimate of the cost of concatenating two consecutive units. This cost is a measure of how well two units join together in terms of spectral and prosodic characteristics. The concatenation cost for the two units that are naturally adjacent in the unit inventory is zero. Therefore choosing adjacent units in unit selection is promoted resulting in better speech quality. This cost can be calculated as a weighted sum of the concatenation sub-costs as follows:

$$C^c(u_i, u_{i+1}) = \sum_{j=1}^{Q} w_j^c C_j^c(u_i, u_{i+1})$$

where Q is the number of concatenation sub-costs and w_j^c is the corresponding weights.

The total cost of selecting a unit sequence u_1^n given the target specification t_1^n is the sum of the target and concatenation costs:

$$C(t_1^n, u_1^n) = \sum_{i=1}^{n} C^t(t_i, u_i) + \sum_{i=1}^{n-1} C^c(u_i, u_{i+1})$$

The unit selection algorithm tries to find the unit sequence u_1^n from the unit inventory that minimizes the total cost:

$$\min_{u_1^n} C(t_1^n, u_1^n)$$

We have implemented a Viterbi decoding algorithm to find the optimal unit sequence. A state-transition network for the units from the speech inventory can be seen in Figure 6.2. The Viterbi algorithm tries to find the optimal path through the network of the nodes. Since the number of units in unit inventory is very large, we have implemented some pruning methods to limit the number of units considered in unit selection. For the

Figure 6.2. Unit selection using Viterbi algorithm can be seen as finding the optimal path through the network of speech units

run-time efficiency of the Viterbi search, we first prune the units from the unit selection based on the length of the matching phonetic context. We start with units that matches 3 phonemes to the left or right with the target specification. If we can not find some minimum number of units matching, we consider the phonemes with matching context length of 2 phonemes. If the number of units is less than a minimal number, we consider all the units that phonetically match the unit in target specification.

For the target sub-costs $C^t_j(t_i, u_i)$, we use the context match length, the energy, duration and pitch difference between the target and selected unit, the place of the unit in the syllable, word and sentence.

For the concatenation sub-costs $C^c_j(u_i, u_{i+1})$, we use the cepstral distance and energy, duration and pitch difference between the consecutive units. The cepstral distance cost calculation is described in the following section.

6.2.1. Cepstral Distance Cost

The cepstral distance at the concatenation points of two consecutive units $(u_i u_{i+1})$ is used in concatenation cost calculation. The cepstral distance is an objective measure of the spectral mismatch between two joining units. For cepstral distance calculation, we used Mel-Frequency Cepstrum Coefficients (MFCC's). We extract the MFCC of the last frame of the first unit and the first frame of the second unit and use the distance between two MFCC vectors for cepstral distance cost.

For calculation of the MFCC's, we window the signal with a hamming window

Figure 6.3. Hamming window for N=30

Figure 6.4. Windowed speech signal

as shown in Figure 6.3 which is calculated using the following equation.

$$h[n] = \begin{Bmatrix} 0.54 - 0.46\cos(2\pi n/N) & 0 \leq n < N \\ 0 & otherwise \end{Bmatrix}$$

The effect of applying a hamming window on a speech waveform is shown in Figure 6.4.

MFCC's of a speech signal can be calculated by taking the discrete cosine transform (DCT) of the filtered magnitude of the Fourier transform of that signal by nonlinear triangular filters [1]. The use of nonlinear triangular filters is motivated by the workings principle of the human hearing system.

The DFT of a signal can be calculated as follows.

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi nk/N}, \qquad 0 \leq k \leq N$$

The boundary points of the triangular filters can be found using the following equation.

$$f[m] = (\frac{N}{F_s})Mel^{-1}(Mel(f_l) + m\frac{B(f_h) - Mel(f_l)}{M+1})$$

where F_s is the sampling frequency, N is the FFT size, M is the number of filters in the filterbank, f_l and f_h are the lowest and highest frequencies of the filterbank. Mel is defined as follows:

$$Mel(f) = 1125\ln(1 + \frac{f}{700})$$

Mel^{-1} is is the inverse of the Mel:

$$Mel^{-1}(f) = 700(\exp(\frac{f}{1125}) - 1)$$

The triangular filter m in the filter bank of M filters is given by:

$$H_m[k] = \left\{ \begin{array}{cc} 0 & k < f[m-1] \\ \frac{(k-f[m-1])}{(f[m]-f[m-1])} & f[m-1] \leq k < f[m] \\ \frac{(f[m+1]-k)}{(f[m+1]-f[m])} & f[m] \leq k \leq f[m+1] \\ 0 & k > f[m+1] \end{array} \right\}$$

The log-energy of the filtered FFT of the signal at output of each filter is calculated as:

$$S[m] = \ln \left[\sum_{k=0}^{N-1} |X[k]|^2 H_m[k]) \right], \qquad 0 < m \leq M$$

The discrete cosine transform of the M filter outputs give the mel-frequency cepstrum coefficients of the input speech signal:

$$c[n] = \sum_{m=0}^{M-1} S[m] \cos(\pi n(m - 1/2)/M), \qquad 0 \leq n < M$$

The spectral discontinuity cost between the last frame of the unit u_i and the first frame of u_{i+1} is calculated as the squared magnitude of the difference between the MFCC's of the frames as follows:

$$C^{c}_{spectral}(u_i, u_{i+1}) = |\mathbf{c}_i(N-1) - \mathbf{c}_{i+1}(0)|^2$$

where N denotes the number of frames of the unit u_i and $\mathbf{c}_i(n)$ the MFCC of the unit u_i at frame n.

7. WAVEFORM GENERATION

The unit selection outputs a sequence of units from the speech inventory to be used for the generation of waveform for the input text. The waveform generation module concatenates the speech waveforms of the selected units. We used a speech representation and waveform generation method based on harmonic sinusoidal coding of speech. Analysis-by-synthesis technique has been used for sinusoidal modeling.

The sinusoidal coding encodes the signal with a sum of sinusoids whose frequency, amplitude and phase are adequate to describe each sinusoid. The harmonic coding is a special case of the sinusoidal coding where the frequencies of the sinusoids are constrained to be the multiples of the fundamental frequency. The harmonic coding takes the advantage of the periodic structure of the speech and is very effective in coding the voiced and unvoiced signals. The harmonic structure of the speech signal can be seen in Figure 7.1. The method that we used for speech representation and waveform concatenation is based on harmonic coding of voiced/unvoiced speech.

The harmonic coding is a parametric coding method. Unlike waveform coders which try to construct the original waveform, parametric coders (vocoders) try to encode the speech into a parametric representation that capture its perceptually important characteristics. For the parameter estimation of harmonic coding, an analysis-by-synthesis framework is used. Harmonic coders represent the speech signal using the magnitudes and phases of its spectrum at multiples of the fundamental frequency. Low bit rate harmonic coders even use the synthetic phase rather than original phase to lower the bit rate. However a high quality speech synthesis requires that the speech representation should be transparent to the listener. Therefore, we used the original phase in the harmonic coding of speech. The coded speech quality heavily depends

Figure 7.1. Power spectral density of a speech signal. The peaks at the pitch harmonics can be clearly seen in the figure

on the correct parameter estimation. For robust parameter estimation we used an analysis-by-synthesis methodology.

A perfectly periodic signal can be represented as a sum of sinusoids:

$$x[n] = \sum_{k=0}^{T_0-1} A_k \cos(nk\omega_0 + \phi_k)$$

where T_0 is the fundamental frequency of the signal, $\omega_0 = 2\pi/T_0$, ϕ_k is the phase of the kth harmonics, and A_k is the amplitude of the kth harmonics. For the quasiperiodic speech signals the same equation can be used to approximate the signal. This approximation can even be used to model the unvoiced sounds. In this case, the fundamental frequency is set to 100 Hz. The error in representing the speech by a harmonic model is estimated as:

$$\epsilon = \sum_{k=-T_0}^{T_0} w^2[k](x[k] - \widetilde{x}[k])^2$$

where w is a windowing function, x is the real speech signal and $\widetilde{x}$ is the harmonic model for the speech signal. For parameter estimation of the harmonic coding, we use this function for error minimization criterion. The values for A_k and ϕ_k that minimize the error is calculated by solving the linear set of equations obtained by integrating the error function. Finding model parameters is a least squares problem. We used QR factorization method for solving the set of linear equations to obtain the model parameters. The correct pitch period estimation is an important part of harmonic coding. The following section describes the method that we used for fundamental frequency estimation.

The model parameters are calculated in a pitch-synchronous manner using overlapping windows of two pitch periods. The scalar quantization of model parameters is done. The unit speech inventory has been compressed about 3 times using quantized model parameters.

Figure 7.2. The original spectrogram for the utterance “Birdenbire karşıdan iri bir köpek geçti”

Figure 7.3. The spectrogram for the reconstructed utterance “Birdenbire karşıdan iri bir köpek geçti” using harmonic coding

The waveform generation using the model parameters for speech waveforms of units are done by taking inverse FFT of the parameters and then overlap-and-add mechanism is used for smooth concatenation of the units. The effect of harmonic coding on a sample utterance is shown in Figures 7.2, 7.3 and 7.4.

7.1. Pitch Estimation

Pitch or fundamental frequency is the rate at which the vocal folds in the human speech production system vibrate, that is the opening and closing of the glottis. Voiced sounds like /a/ cause the vocal folds vibrate, however the unvoiced sounds like /s/ does not vibrate the folds. For the harmonic coding of speech, correct pitch estimation is very important. The quality of the coded speech is severely degraded at the wrong pitch marks. Pitch estimation is also used in unit selection process since the pitch contributes to the perceived prosody most [21]. The algorithm that we used for pitch estimation is based on the normalized autocorrelation method. The autocorrelation of a signal using N samples can be calculated as follows:

$$R[k] = \sum_{n=0}^{N-1-k} x[n]w[n]x[n+k]w[n+k]$$

Figure 7.4. The spectrogram for the reconstructed utterance “Birdenbire karşıdan iri bir köpek geçti” using harmonic coding with parameter quantization

Figure 7.5. Speech waveform for the utterance “karşıdan”

Figure 7.6. Pitch track with autocorrelation for the utterance “karşıdan”

where $x[n]$ is the input signal and $w[n]$ is a window of length N, possibly a rectangular window.

The autocorrelation function has been commonly used for pitch estimation. The value of k that gives the highest value of the autocorrelation function $R[k]$ excluding $k = 0$, is determined to be the pitch period of the signal. The search for the pitch is constrained to a region that the pitch period can be, i.e. 50 Hz-500Hz. The window size is set to be at least two expected maximum pitch period. The speech waveform for the utterance *karşıdan* and its pitch track based on autocorrelation method can be seen in Figures 7.5 and 7.6. The unvoiced regions such as for /k/, /ş/ sounds are shown to have zero pitch. At the end of the utterance pitch doubling has occurred.

The normalized autocorrelation is calculated as:

$$R_n(k) = \frac{\sum_{n=0}^{N-1} x[n]x[n+k]}{\sqrt{\sum_{n=0}^{N-1} x^2[n] \sum_{n=0}^{N-1} x^2[n+k]}}$$

The normalized autocorrelation method is more reliable than autocorrelation method, since the number of samples used in calculation is constant. The unsmoothed pitch track with normalized autocorrelation method can be seen in Figure 7.7. The normalized autocorrelation function gives the same value at multiples of the pitch period for perfectly periodic signals. This may cause the detection of pitch period as two times pitch period, that is called pitch halving. We used a decaying factor in the calculation of the normalized autocorrelation which decreases the value of the function as k increases. As can be seen in Figure 7.8, this helps to correct most of the pitch halving errors.

Figure 7.7. Unsmoothed pitch track for the utterance “karşıdan”

Figure 7.8. Unsmoothed pitch track with a decaying term for the utterance “karşıdan”

We also performed some post-processing to smooth the pitch track, since the normalized autocorrelation method is error prone. The smoothing process takes into the consideration that the pitch does not change drastically from frame to frame. We applied median smoothing that keeps a history of the pitch values, sorts it and takes the one in the middle. Smoothed pitch track is shown in Figure 7.9.

Figure 7.9. Smoothed pitch track for the utterance “karşıdan”

8. EXPERIMENTS

In the development of the Turkish text-to-speech system many experiments have been done to evaluate the system performance with different configurations.

In unit selection, different unit sizes have been experimented with. The use of halfphones in unit selection enables us to use relative weighting for diphone and phone concatenation. In the informal listening tests we can conclude that using a mixture of two concatenation gives best speech quality.

Since the current implementation does not include a prosody generation module, we wanted to see the effect of transplanted prosody on the output speech quality. For this purpose, the prosody of original recordings have been extracted and used in the unit selection as the target specification. The original recording, the synthesized utterance with no prosody and the synthesized utterance with transplanted prosody are evaluated in the quality assessment section.

During the development of the harmonic coder, the autocorrelation and crosscorrelation based pitch detection algorithms have been experimented. The crosscorrelation method has been found to give better performance on the coded speech quality. The different scalar quantization methods have been tried to lower the bit rate of the coder.

The weights used in cost calculation in the unit selection have been empirically decided by changing the parameters and listening the output speech quality. The cepstral distance between units have been experimented.

Two speech inventory of different sizes have been used in quality assessment to measure the affect of using large speech databases in unit selection.

For waveform concatenation, time-domain waveform joining and harmonic coding based overlap-and-add mechanism was tried. Time-domain waveform joining was found to cause glitches in the synthesized speech. The use of harmonic coding with overlap-and-add mechanism results in smoother concatenations.

The effect of using the cost of mismatch in pitch periods of two joining units has been experimented. The units that are longer than average has been given extra cost.

9. QUALITY ASSESSMENT

For the evaluation of the synthetic voice quality, we carried out formal listening tests. The tests were of two type. One of the test requires the listeners to rank the voice quality in terms of naturalness, intelligibility and pleasantness using a MOS like scoring. The other test is a diagnostic rhyme test.

Mean opinion score (MOS) tests are commonly used for evaluating the quality of speech coding algorithms. The MOS have also been used to assess the synthesized speech quality. The MOS scores for speech synthesis are generally have been given in three categories, namely the intelligibility, the naturalness and the pleasantness. We carried out formal subjective MOS like tests to assess the quality of the synthesized speech produced from our system. The test was carried out by snythesizing a set of 50 sentences that have been selected from the speech corpus randomly that has been kept separate from the training set used for constructing synthesis database. The reason for choosing the sentences for which we have also the original speech waveforms spoken by our speaker is that we also use the original recordings in our tests to ensure the reliability of our test results. In the MOS test, 10 subjects (2 females) and 50 sentences were used. The subjects listened the sentences using headphones. The sentences were at 16kHz and 16 bits. The subjects were instructed to rate the sentences on a scale of 1-5 where 1 is very poor and 5 is excellent. Some speech samples of speech coders having different MOS scores were presented to the subjects to ensure consistency in evaluating the speech quality. The subjects were also familiarized with the speech synthesis by listening some example utterances of varying quality. In the MOS test we evaluated the quality of the five systems. The first system uses the original recordings, that have been coded by our harmonic coder and reconstructed, from the test speech corpus that have been kept separate from the training set. The second system uses our unit selection synthesizer with a speech unit inventory containing about 19 hours of speech recording. The third system uses a speech inventory containing about 3 hours of recording. The fourth system is the same system with the second one except that the original prosody from the original recordings are used in the unit selection process. The final system

Table 9.1. Systems evaluated in MOS test with average scores

System	**Description**	**MOS**
X	The original recordings with harmonic coding	4.91
T	Speech synthesis using 19 hours of speech	4.2
W	Speech synthesis using 3 hours of speech with original prosody	4.11
G	Speech synthesis using 19 hours of speech with original prosody	4.01
R	Speech synthesis using 3 hours of speech	4

Figure 9.1. MOS scores for the evaluated systems

is the same system with the third system except the original prosody is used. Using five systems 50 sentences that can be found in the appendix A are synthesized. 250 sentences from all of the systems are divided into five sets of 50 sentences. Each set contains 10 sentence from each system. In each set, all of the sentences are used and no repeating of the same sentence from different systems is allowed. Each set is listened by two subjects. The subjects give ratings in terms of intelligibility, naturalness and pleasantness to each sentence. The evaluated systems and their average MOS scores can be seen in Table 9.1. Each system's MOS scores for each category is shown in Figures 9.1 and 9.2. The detailed scores for each system is given in the appendix B. The differences in system ratings were found to be significant using ANOVA analysis. The analysis yielded an F-value of about 21 whereas the critical F-values are about 3.3 and 5.0 for P=0.01 and P=0.001, respectively.

We also conducted an intelligibility test. Diagnostic rhyme test (DRT) uses monosyllabic words that have the consonant-vowel-consonat pattern. This test measures the capability of discrimination of the initial consonants for the system evaluated. The DRT word list of ANSI standard for English contains one hundred and ninety two words arranged in ninety six rhyming pairs which differ only in their initial consonant sounds.

Figure 9.2. MOS scores for the evaluated systems grouped by test category

Table 9.2. DRT word list for Turkish

Voicing		**Nasality**		**Sustenation**		**Sibilation**		**Graveness**		**Compactness**	
var	far	mal	bal	van	ban	çent	kent	biz	diz	türk	kürk
ben	ten	mat	bat	ve	be	saç	taç	pas	tas	fan	han
gez	kez	naz	daz	var	bar	sez	tez	boz	doz	ver	yer
bul	pul	mil	bil	şap	çap	jön	yön	pek	tek	faz	haz
din	tin	mit	bit	vur	bur	jel	gel	pers	ters	dün	gün
diz	tiz	mor	bor	şam	çam	sin	tin	fon	ton	tap	kap
zor	sor	mut	but	şan	çan	zan	tan	post	tost	tuş	kuş
zevk	sevk	mir	bir	fes	pes	say	tay	put	tut	toz	koz
zar	sar	muz	buz	şark	çark	zam	tam	pak	tak	tas	kas
zen	sen	nam	dam	fil	pil	zat	tat	poz	toz	taş	kaş
zil	sil	nar	dar	şal	çal	zerk	terk	pür	tür	tat	kat
bay	pay	nem	dem	şık	çık	çal	kal	bağ	dağ	tel	kel
ders	ters	nur	dur	şok	çok	sak	tak	bul	dul	düz	güz
gör	kör	nal	dal	fas	pas	çil	kil	bel	del	tül	kül
vay	fay	nil	dil	fark	park	çim	kim	but	dut	ton	kon
göl	çöl	men	ben	fiş	piş	san	tan	fer	ter	tork	kork

The list has been divided into six categories depending on the distinctive features of speech. The categories has been constructed in terms of voicing, nasality, sustenation, sibilation, graveness, and compactness characteristics of the sounds. For assessing the intelligibility of the synthesized speech in Turkish, we constructed a DRT word list for Turkish based on the categories of the DRT word list of English as shown in Table 9.2. The DRT list has been designed to exploit the distinctive features of Turkish speech at maximum. The list contains total of 96 words in pairs.

Using the DRT word list for Turkish, we carried out an intelligibility test for our system. The randomly selected words from each pair of the DRT word list was synthesized using our TTS system. The output speech waveforms were played to 10 native Turkish listeners who were then asked to choose which one of the words given in pairs from the DRT list they heard. The listeners were assured to have a good hearing and discrimination of sounds. The test results are shown in Table 9.3 as the percentage of number of correct selections for the two systems evaluated.

Table 9.3. Systems evaluated in DRT test with DRT scores

System	**Description**	**DRT**
T	Speech synthesis using 19 hours of speech	0.95
R	Speech synthesis using 3 hours of speech	0.94

10. CONCLUSIONS

In this work, a corpus-based concatenative speech synthesis system architecture for Turkish has been designed and implemented. We have done a literature survey on corpus-based concatenative speech synthesis research. Different techniques have been investigated and applied for optimal speech corpus design. Turkish phonetics have been studied to create a phoneme set that is suitable and adequate for representing all the sounds in Turkish to be used in a speech synthesis system. A pronunciation lexicon for the root words in Turkish has been prepared. A simple text normalization module for Turkish has been implemented. A grapheme-to-phoneme conversion module based on morphological analysis of Turkish has been implemented. Transplanted prosody have been experimented for evaluating the importance of intonation and duration modeling in the system. A unit selection algorithm based on dynamic programming has been implemented. Target and concatenation costs to be used in unit selection have been extracted. A cepstral distance measure used in concatenation cost calculation was implemented. A speech representation based on harmonic coding has been implemented. Speech corpus has been compressed by a factor of three with slight degradation on the voice quality using the speech model. The smooth concatenation of speech units have been succeeded using the harmonic coding parameters. As a result a unit selection-based concatenative speech synthesis system capable of generating highly intelligible and natural synthetic speech for Turkish has been developed. Subjective tests have been carried out to assess the speech quality generated by the system. A DRT word list for Turkish has been constructed to carry out the intelligibility tests. The final system got 4.2 MOS like score and 0.95 DRT correct word discrimination percentage.

APPENDIX A: SENTENCES USED FOR MOS TESTS

1. bugün ata adlı özel uçakla çorluya gitti.
2. sonuç olarak herkes konuyu kendine göre yorumladı.
3. ve beni ölüm cezasına çarptırdılar.
4. musaya apaçık dokuz mucize verdik.
5. kafesleri tüm aramalarına karşın bulamamışlar.
6. herkesin birden cesareti kırıldı.
7. yelken bezinden bir çuval vardı.
8. nasıl yararlı kullanılabileceği anlatılıyor.
9. müzikle pek ilgisi kalmamıştı.
10. kalbinin en soluksuz atışıydı.
11. o gün kralın söylediği en akıllıca söz bu olmuştu.
12. ünlü sanatçı zülfü livaneli.
13. cumhurbaşkanı turgut özal.
14. felsefe antik yunanda çıktıysa da...
15. gabriel evli bir kadın.
16. bu dilencinin babası ha!
17. neyle itham ediliyorsun?
18. ister suçlu ister suçsuz olsun.
19. bu tutarsızlıktan kurtulmanın bir yolu var mıdır?
20. derslerden yeter derecede bilgi edinebildim mi?
21. herkeste büyük şaşkınlık uyandırdı.

22. üç boyutlu cisimlerin iki boyutlu izdüşümleridir.

23. o bölümü yazıyorum şimdi.

24. kısa vadeli dış borçların ödenmesinde zorluklar olmuştur.

25. üzerini yapraklarla kapladı.

26. mari için büyük bir darbe oldu.

27. her kesimden insanımızın namus borcudur.

28. ben o kadar büyük bir günahkarım ki.

29. allah onların ne gizlediklerini bilir.

30. sivrisineklere yem oldu.

31. zaman en kötü günü de sona erdirir.

32. iki tane aktif yanardağı vardı.

33. kendisinin de vardır sigortası kardeşinin de.

34. dördüncü bölüm siyasi haklar ve ödevler.

35. yavaş yavaş tatlı düşlere kaptırdı kendini.

36. aslında evsizlik beni hiç rahatsız etmiyor.

37. güzel mi güzel bir yer burası.

38. üçü de yolun ilk dönemecinde çarçabuk gözden yittiler.

39. bilimsel yayınları acıyla izledim.

40. fuar altı eylüle kadar açık kalacak.

41. ben ordu başkumandanına çıktım.

42. yoksa okulu kapatmak zorunda kalacağım.

43. bu durum çok doğal karşılanır.

44. ekonomide radikal önlemlere ihtiyaç olduğunu yazdı.

45. elle işlenmiş çok hoş bir porselen fincan.

46. diplomasi ermeni ve rum sorununu tartışıyor.

47. yoksa hiç öğrenmesem mi?

48. yaşamak yalnız yemek ve uyumak olmamalı.

49. güvenilir bir ortam yaratmaktır.

50. kendini eğitmesi zaman aldı.

APPENDIX B: MOS TEST RESULTS

The detailed MOS test scores for each system explained in Table 9.1 can be seen in the following figures.

Figure B.1. Speech synthesis system G

Figure B.2. Speech synthesis system R

Figure B.3. Speech synthesis system T

Figure B.4. Speech synthesis system W

Figure B.5. Speech synthesis system X

REFERENCES

1. Huang, X., A. Acero and H. W. Hon, *Spoken Language Processing*, Prentice Hall PTR, New Jersey, 2001.

2. Greenwood, A. R., “Articulatory Speech Synthesis Using Diphone Units”, *IEEE international Conference on Acoustics, Speech and Signal Processing*, pp. 1635–1638, April 1997.

3. Sagisaka, Y., N. Iwahashi and K. Mimura, “ATR v-TALK Speech Synthesis System”, *Proceedings of the ICSLP*, Vol. 1, pp. 483–486, 1992.

4. Black, A. W. and P. Taylor, “CHATR: A Generic Speech Synthesis System”, *Proceedings of the International Conference on Computational Linguistics*, Vol. 2, pp. 983–986, 1994.

5. Black, A. W. and P. Taylor, “Automatically Clustering Similar Units for Unit Selection in Speech Synthesis”, *Proceedings of the European Conference on Speech Communication and Technology*, Vol. 2, pp. 601–604, Rhodos, Greece, 1997.

6. Hunt, A. and A. W. Black, “Unit Selection in a Concatenative Speech Synthesis System Using a Large Speech Database”, *Proceedings of the IEEE on Acoustics and Speech Signal Processing*, Vol. 1, pp. 373–376, Munchen, Germany, 1996.

7. Stylianou, Y., “Applying the Harmonic Plus Noise Model in Concatenative Speech Synthesis”, *IEEE Trans. on Speech and Audio Processing*, Vol. 9, No. 1, pp. 21-29, January 2001.

8. Beutnagel, M., A. Conkie, J. Schroeter, Y. Stylianou and A. Syrdal, “The AT&T Next-Gen TTS system”, *In The Proceedings of the Joint Meeting of ASA, EAA, and DAGA*, pp. 18-24, Berlin, Germany, 1999.

9. Bernd, M., “Corpus-based Speech Synthesis: Methods and Challenges”, *Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (Univ. Stuttgart)*, AIMS 6 (4), pp. 87-116, 2000.

10. Donovan, R. E., “A New Distance Measure for Costing Spectral Discontinuities in Concatenative Speech Synthesisers”, *Proceedings of the 4th ISCA Tutorial and Research Workshop on Speech Synthesis*, Atholl Palace Hotel, Scotland, 2001.

11. Donovan, R. E. and E. M. Eide, “The IBM Trainable Speech Synthesis System”, *Proceedings of the international Conference on Spoken Language Processing*, Vol. 5, pp. 1703–1706, Syndney, Australia, 1998.

12. Bulyko, I., *Flexible Speech Synthesis Using Weighted Finitestate Transducers*, University of Washington, Ph.D. Dissertation, Electrical Engineering, 2002.

13. Rowe, D. G., *Techniques for Harmonic Sinusoidal Coding*, Ph.D. Thesis, University of South Australia, 1997.

14. Conkie, A., “Robust Unit Selection System for Speech Synthesis”, *Proc. Joint Meeting of ASA, EAA and DEGA*, Berlin, Germany, March 1999.

15. Stylianou, Y., “Removing Phase Mismatches in Concatenative Speech Synthesis”, *The 3 rd ESCA/COCOSDA Workshop on Speech Synthesis*, Jenolan Caves, NSW, Australia, November 1998.

16. Conkie, A., M. C. Beutnagel, A. K. Syrdal and P. E. Brown, “Preselection of Candidate Units in a Unit Selection-based Text-to-Speech Synthesis System”, *In ICSLP-2000*, Vol. 3, pp. 314-317, Beijing, China, October 2000.

17. Donovan, R. E., “Current Status of the IBM Trainable Speech Synthesis System”, *Proceedings of the 4th ISCA Tutorial and Research on Speech Synthesis*, Edinburgh, 2001.

18. Sproat, R., “Multilingual Text Analysis for Text-to-Speech Synthesis”, *The 12 th European Conference on Artificial Intelligence*, 1996.

19. Şayli, Ö., *Duration Analysis and Modelling for Turkish Text-to-Speech Synthesis*, M.S. Thesis, Bogazici University, 2002.

20. Abdullahbeşe, E., *Fundamental Frequency Contour Synthesis for Turkish Text-to-Speech*, M.S. Thesis, Bogazici University, 2001.

21. Plumpe, M. and S. Meredith, “Which is More Important in a Concatenative Text-to-Speech System - pitch”, *In the 3 ESCA/COCOSDAInternational Workshop on Speech Synthesis*, pp. 231-236.

22. Güngör, T., *Computer Processing of Turkish: Morphological and Lexical Investigation*, Ph.D. Thesis, Bogazici University, 1995.

23. Oflazer, K., *Two-level Description of Turkish Morphology*, Literary and Linguistic Computing, Vol. 9, No:2, 1994.

24. Oflazer, K. and S. Inkelas, “A Finite State Pronunciation Lexicon for Turkish”, *in Proceedings of the EACL Workshop on Finite State Methods in NLP*, Budapest, Hungary, April 13-14, 2003.

REFERENCES NOT CITED

Beutnagel, M., A. Conkie and A. K. Syrdal, “Diphone Synthesis Using Unit Selection”, *In SSW3*, pp.185-190, 1998.

Beutnagel, M., M. Mohri and M. Riley, “Rapid Unit Selection from a Large Speech Corpus for Concatenative Speech Synthesis”, *In Proceedings of the European Conference on Speech Communication and Technology*, Vol. 2, pp. 607-610, Budapest, Hungary, 1999.

Beutnagel, M. and A. Conkie, “Interaction of Units in a Unit Selection Database”, *Proc. European Conf. Speech Communication & Technology*, Vol. 3, pp. 1063-1066, Budapest, Hungary, Sept. 1999.

Bulyko, I. and M. Ostendorf, “Unit Selection for Speech Synthesis Using Splicing Costs with Weighted Finite State Transducers”, *Proc. of Eurospeech*, pp. 987-990, 2001.

Bulyko, I. and M. Ostendorf, “Joint Prosody Prediction and Unit Selection for Concatenative Speech Synthesis”, *in Proc. ICASSP*, Vol. 2, pp. 781-784, 2001.

Campione, E. and J. Véronis, “A Multilingual Prosodic Database”, *5th International Conference on Spoken Language Processing*, pp. 3163-3166, Sidney, 1998.

Conkie, A., G. Riccardi and R. C. Rose, “Prosody Recognition from Speech Utterances Using Acoustic and Linguistic Based Models of Prosodic Events”, *Proc. European Conf. Speech Communication & Technology, In EUROSPEECH’99*, pp. 523-526, Budapest, Hungary, Sept. 1999.

Donovan, R. E., “Segment Pre-selection in DecisionTree Based Speech Synthesis Systems”, *Proc. ICASSP 2000*,Vol. 2, pp. 937-940, Istanbul, 2000.

Gersho, A., “Advances in Speech and Audio Compression”, *Proceedings of The IEEE*, Vol. 82, pp. 900-918, June 1994.

Hon, H., A. Acero, X. Huang, J. Liu and M. Plumpe, "Automatic Generation of Synthesis Units for Trainable text-to-speech Systems", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, WA, Vol. 1, pp. 293-296, Seattle, May 1998.

Huang, X., A. Acero, J. Adcock, H.-W. Hon, J. Goldsmith, J. Liu and M. Plumpe, "Whistler: A Trainable Text-to-Speech System", *Proc. 4th Int'l. Conf. on Spoken Language Processing*, pp. 2387-2390, Piscataway, NJ, 1996.

Jefremov, A., *Modeling the Phase of the Pitch Cycle in Harmonic Coding of Speech*, Bachelor thesis, Stockholm (Sweden) - Tallinn (Estonia), 1999.

Kain, A. and Y. Stylianou, "Stochastic Modeling of Spectral Adjustment for High Quality Pitch Modification", *IEEE International Conference on Acoustics, Speech and Signal Processing 2000*, Vol. 2, pp. 949-952, Istanbul, Turkey, June 2000.

Kapilow, D., Y. Stylianou and J. Schroeter, "Detection of Non-stationarity in Speech Signals and its Application to Time-scaling", *6th European Conference on Speech Communication and Technology*, Vol. 5, pp. 2307-2310, Budapest, Hungary, Sept. 5-9, 1999.

Li, Q. and L. Atlas, "Time-Variant Least Squares Harmonic Modeling", *in Proceedings of the 2003 IEEE ICASSP*, Vol. 2, pp. II- 41-4, 2003.

Liu, M. and A. Lacroix, "Pole-zero Modeling of Vocal Tract for Fricative Sounds", *IEEE International Conference on Acoustics*, Speech and Signal Processing, ICASSP-97, Vol. 3, pp. 1659-1662, 1997.

Macon, M. W. and M. A. Clements, "Sinusoidal Modeling and Modification of Unvoiced Speech", *in IEEE Transactions on Speech and Audio Processing*, Vol. 5, No. 6, pp. 557-560, Nov, 1997.

Makashay, M. J., C. W. Wightman, A. K. Syrdal and A. Conkie, “Perceptual Evaluation of Automatic Segmentation in text-to-speech Synthesis”, *ISCLP 2000*, Vol. 2, pp. 431-434, Beijing, China, 16-20 Oct. 2000.

Prevost, S. and M. Steedman, “Specifying Intonation from Context for Speech Synthesis”, *Speech Communication*, Vol. 15, No. 1-2, pp. 139-153, 1994.

Salami, R., C. Laflamme, B. Bessette and J.-P. Adoul, “ITU-T Recommendation G.729 Annex A: Reduced Complexity 8 kb/s CS-ACELP Codec for Digital Simultaneous Voice and Data”, *IEEE Communications Magazine*, Vol. 35, No. 9, pp. 56-63, Sep. 1997.

Serra, X., “Musical Sound Modeling with Sinusoids Plus Noise”, *in Musical Signal Processing*, pp. 91-122, Swets & Zeitlinger, 1997.

Stylianou, Y., “Analysis of Voiced Speech Using Harmonic Models”, *137th Mtg. Acoustical Society America*, Berlin, 14-19 March 1999.

Stylianou, Y., “Assessment and Correction of Voice Quality Variabilities in Large Speech Databases for Concatenative Speech Synthesis”, *IEEE International Conference on Acoustics, Speech and Signal Processing*, Phoenix, Arizona, 1999.

Stylianou, Y., “Concatenative Speech Synthesis Using a Harmonic Plus Noise Model”, *The 3 rd ESCA/COCOSDA Workshop on Speech Synthesis*, Jenolan Caves, NSW, Australia, November 1998.

Stylianou, Y., T. Dutoit and J. Schroeter, “Diphone Concatenation Using a Harmonic Plus Noise Model of Speech”, *Proc. Eurospeech*, pp. 613-616, Rhodes, Greece, 1997.

Stylianou, Y., “Synchronization of Speech Frames Based on Phase Data with Application to Concatenative Speech Synthesis”, *6th European Conf. Speech Communication and Technology*, pp. 2343-2346, Budapest, Hungary, 5-9 Sept, 1999.

Stylianou, Y., “A Simple and Fast Way for Generating a Harmonic Signal”, *IEEE Signal Processing Letters*, Vol. 7, No. 5, pp. 111-113, May 2000.

Sun, X., “Predicting Underlying Pitch Targets for Intonation Modeling”, *Proc. of the 4th 1SCA Tutorial and Research Workshop on Speech Synthesis*, pp. 143-148, Perthshire, Scotland, 2001.

Sun, X. and T. H. Applebaum, “Intonational Phrase Break Prediction Using Decision Tree and N-Gram Model”, *Proc. of 7th European Conference on Speech Communication and Technology (Eurospeech)*, Vol. 1, pp. 537-540, Aalborg, Denmark, 2001.

Syrdal, A. K., C. Wightman, A. Conkie, Y. Stylianou, M. Beutnagel, J. Schroeter, V. Strom, K. S. Lee and M. J. Makashay, “Corpus-based Techniques in the AT&T NextGen Synthesis System”, *ICSLP*, Vol. III, pp. 410-415, Beijing, China, 16-20 Oct. 2000.

Syrdal, A. K., Y. Stylianou, L. Garrison, A. Conkie and J. Schroeter, “TD-PSOLA versus Harmonic plus Noise Model in Diphone Based Speech Synthesis”, *IEEE ICASSP*, pp. 273-276, Seattle, WA, 1998.

Syrdal, A. K., J. Hirschgerg, J. McGory and M. Beckman, “Automatic ToBI Prediction and Alignment to Speed Manual Labeling of Prosody”, *Speech Communication*, Vol. 33 (1-2), pp. 135-151, Jan. 2001.

Syrdal, A. K. and J. McGory, “Inter-transcriber Reliability of ToBI Prodosic Labeling”, *ICSLP 2000*, Beijing, China, October 2000.

Syrdal A. K., “Phonetic Effects on Listener Detection of Vowel Concatenation”, *EUROSPEECH 2001*, Aalborg, Denmark, September 2001.

Stylianou, Y., “On the Implementation of the Harmonic Plus Noise Model for Concatenative Speech Synthesis”, *IEEE International Conference on Acoustics, Speech* *and Signal Processing*, Istanbul, Turkey, June 2000.

Stylianou, Y. and A. K. Syrdal, "Perceptual and Objective Detection of Discontinuities in Concatenative Speech Synthesis", *Proc. ICASSP*, Vol. 2, pp. 837-840, 2001.

Taylor, P. "Concept-to-Speech synthesis by phonological structure matching", *Philosophical Transactions of the Royal Society, Series A. 356(1769)*, pp. 1403-1416, 2000.

Stylianou, Y., "On the Implementation of the Harmonic Plus Noise Model for Concatenative Speech Synthesis", IEEE International Conference on Acoustics, *Speech and Signal Processing*, Istanbul, Turkey, June 2000.

Stylianou, Y. and A. K. Syrdal, "Perceptual and Objective Detection of Discontinuities in Concatenative Speech Synthesis", *Proc. ICASSP*, Vol. 2, pp. 837-840, 2001.

Virtanen, T., *Audio Signal Modeling with Sinusoids Plus Noise*, MSC Thesis, Tampere University of Technology, 2001.

Wightman, C. W., A. K. Syrdal, G. Stemmer, A. Conkie and M. Beutnagel, "Perceptually Based Automatic Prosody Labeling and Prosodically Enriched Unit Selection Improve Concatenative text-to-speech Synthesis", *In ICSLP*, Vol. 2, pp. 71-74, 2000.

Xue, S. and D. Deliyski, "Effects of Aging on Selected Acoustic Voice Parameters: Preliminary Normative Data and Educational Implications", *Educational Gerontology, 27, Taylor & Francis*, London, New York, pp. 159-168, 2001.

Yi, J. R. W. and J. R. Glass, "Natural-Sounding Speech Synthesis using Variable-Length Units", *Proc. ICSLP-98*, Sydney, Australia, Vol. 4, pp. 1167-1170, 1998.

Yi, J. R. W., J. R. Glass and I. L. Hetherington, "A Flexible, Scalable Finite-state Transducer Architecture for Corpus-based Concatenative Speech Synthesis", *ICSLP*, Vol. 3, pp. 322-325, 2000.