A HYBRID STATISTICAL/UNIT-SELECTION TEXT-TO-SPEECH SYNTHESIS SYSTEM FOR MORPHOLOGICALLY RICH LANGUAGES

A Thesis

by

Ekrem Güner

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Electrical and Electronics Engineering

Özyeğin University
June 2013

A HYBRID STATISTICAL/UNIT-SELECTION TEXT-TO-SPEECH SYNTHESIS SYSTEM FOR MORPHOLOGICALLY RICH LANGUAGES

Approved by:

Assistant Professor Cenk Demiroğlu,
Advisor
Department of Electrical and Electronics
Engineering
Özyeğin University

Professor Tanju Erdem
Department of Electrical and Electronics
Engineering
Özyeğin University

Associate Professor Barış Bozkurt
Department of Electrical and Electronics
Engineering
Bahçeşehir University

Date Approved: June 2013

To my parents

ABSTRACT

Two most prominent examples of Text-to-Speech (TTS) systems are Unit Selection based TTS (UTTS) and the Hidden Markov Model (HMM) based TTS (HTTS). UTTS has been the dominant approach of the last decade while HTTS has been increasingly getting more attention from the TTS research community. Both systems have distinct pros and cons. Despite its success, UTTS has some disadvantages such as the sudden discontinuities in speech which cause distraction whereas HTTS lacks of those artifacts. However, UTTS systems offer high quality speech given a huge unit database where the storage is not a problem. On the other hand, the small memory footprint requirement of HTTS systems makes them attractive for embedded devices. Here, a novel hybrid statistical/unit selection TTS system for morphologically rich languages is proposed. The proposed hybrid system aims at improving the quality of the baseline HTTS system while keeping the memory footprint small. First, the motivation of the proposed hybrid system is given after the comparison of both systems. Then the proposed hybrid system is presented along with the details of the baseline HTTS system. In order to assess the performances of proposed and baseline systems, the subjective and objective tests are conducted. Intelligibility and quality scores of the baseline system are comparable to the MOS scores of English reported in the Blizzard Challenge tests. Results of the AB preference tests revealed the listeners' preference for the hybrid system over the baseline system.

ÖZETÇE

Metinden Konuşma Sentezleme (MKS) alanında en yaygın kullanılan iki teknik, Birim Seçmeli MKS (BMKS) ve Saklı Markov Modeli tabanlı MKS (SMKS) teknikleridir. MKS sistemleri son dönemlerde kullanılan en dominant teknik olarak ortaya çıkarken, SMKS sistemleri de gün geçtikçe artan popülaritesi ile öne çıkmaktadır. Her iki sistemin de kendine özgü avantaj ve dezavantajları bulunmaktadır. BMKS sistemleri çok başarılı olmalarına rağmen, dinleyicileri rahatsız eden ani süreksizlikler içermektedirler. SMKS sistemleri ise birim seçme algoritmasının ortaya çıkardığı bu hatalardan yoksundurlar. BMKS sistemleri, kullanılan ses veritabanının büyüklüğüyle orantılı olarak yüksek kalitede ses üretebilmektedir. SMKS sistemleri ise çok küçük bir saklama alanı kullandıklarından, daha yaygın olarak gömülü uygulamalarda tercih edilmektedir. Bu tez çalışmasında, morfolojik olarak zengin diller için, SMKS sistemini temel alan ve veri kullanımını yine makul seviyede tutarak kalitesini arttırmayı hedefleyen bir melez istatistiksel/birim seçmeli MKS sistemi önerilmiştir. Öncelikle, iki sistemin karşılaştırması yapıldıktan sonra, önerilen melez sistemin ana fikri verilmiştir. Daha sonra melez sistem, geliştirilen temel SMKS sistemi ile birlikte ayrıntılı olarak anlatılmıştır. Temel ve melez sistemin performanslarının ölçülmesi için de, subjektif ve objektif testler gerçekleştirilmiştir. Temel sistemin anlaşılabilirlik ve kalite puanlarının, literatürde İngilizce dili için yapılan çalışmalarda rapor edilen değerlerle benzer olduğu görülmüştür. AB tercih testlerinde ise, dinleyicilerin önerilen melez sistemi temel sisteme tercih ettikleri görülmüştür.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor Professor Cenk Demiroğlu for mentoring me through the learning process of this master thesis. His invaluable guidance and intimate attitude during this period has made me much more solid researcher and made this thesis come true. I also would like to thank my thesis committee members Professor Tanju Erdem and Assoc. Professor Barış Bozkurt for spending their precious times.

During the period of this thesis, many friends have been helpful to color my life. My past and present friends in the Speech Processing Lab. made this journey more fun. I have been blessed with a friendly and cheerful group of fellow students in the graduate school. I also want to to acknowledge my flat mates for sharing with me a wonderful year.

Special thanks to our brand new university for giving us an opportunity to work with wonderful people in a warm environment.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Text-to-Speech is the general name of a language processing application which refers to converting written language signal, text, into spoken language signal, speech. The earliest example of such systems is *Voder* which is invented by Homer W. Dudley for Bell Labs. in 1930s. It was using the vocoder technology first in the time. Pitch was controlled by foot pedal. It was creating highly robotic speech in a small portion of speech spectrogram. Although synthetic speech is still distinguishable from natural speech, quality and naturalness of the produced speech increased significantly and are much more close to human generated speech now. Today, TTS technology is highly accepted by people and used in personal or service based products.

There are number of techniques to create synthetic speech which is known as **synthetic speech synthesis**. Two prominent approaches of today are concatenative and statistical speech synthesis. Concatenative speech synthesis is based on a very intuitive idea which is to create speech by sequencing the optimal harmonious speech units for a given sentence. Generally, these units are selected from a database of speech segments which are cropped from real records. Concatenative systems are considered to be able to produce high quality speech. Although the main idea is simple and successful, it is known that this is not the way humans create speech. One insightful model of human speech production is source/filter model which is used in signal processing applications. In this model air pressure and the vibration of the vocal fold are considered as the sources whereas vocal tract is considered as a filter. Then speech is decoded for each frame by source and filter coefficients with a reasonable error. Given these parameters, an encoder (vocoder) can reproduce the speech. Statistical speech synthesis systems aim at modeling these parameters with statistical distributions associated with the context information of the language. Then given a context of sentence to be synthesized, most likely speech parameters are estimated by the system using statistical decision techniques. A vocoder system creates the speech using estimated parameters.

Both techniques come with their advantages and disadvantages. Concatenative speech synthesis systems can create high quality speech but requires a huge speech database during the synthesis time. Although the high quality is achieved, these systems may also create some artifacts in synthesized speech. On the other hand, statistical speech synthesis systems are able to synthesize smooth speech at the expense of degrading some quality. These systems also use relatively less resources. Since they learn the statistics of speech parameters and use a statistical estimation technique during synthesis, they don't need the database during synthesis. Considering all these facts, researcher also have been investigated the use of different examples of both techniques together in a hybrid approach to gain the advantages of both sides.

UTTS systems are the most used type of concatenative approach. On the other side, HTTS systems are the best representative of statistical approach. In this thesis, a novel hybrid TTS system which utilizes a baseline HTTS system and a morpheme based UTTS for morphologically rich languages is proposed. The goal of the proposed systems is to improve the quality of the baseline HTTS systems while keeping the resource usage in reasonable level. In order to do that, a morpheme database consists of source and filter parameters is created from the speech database. Morpheme database comprises most used morphemes in the language with minimal context information. In the baseline HTTS system, models are trained as usual ways. In synthesis stage morphemes of the sentence to be synthesized is identified. The best morpheme candidates for given sentence are selected from the database via proposed algorithms. Then, these morphemes are used to guide the parameter generation process of HTTS scheme via proposed method. These real morpheme segments are thought to be the reference points for parameter trajectory generation. In order to assess the performance of the hybrid system, listening tests are conducted. Test results showed that listeners preferred the hybrid system over the baseline system.

The thesis is organized as follows: In section 2, background of general HTTS system is given and parts of a typical HTTS system is explained. In Section 3, first UTTS approach is briefly explained and compared with HTTS. Then, a survey of hybrid systems in the literature is given with classifications. After comparing the both techniques and introducing the different hybrid approaches, the proposed hybrid system is presented with details in Section 4. In Section 5, results of subjective and objective test are given and discussed. Finally, conclusion and comments about the work is given in Section 6.

CHAPTER II

HMM BASED TTS

In pure UTTS, the essential is to use recorded speech units to build synthetic speech. The key point is the selection of the best harmonious group of real units for given sentence. There is no attempt to learn how that units are created or speech is produced. Hence, if there is no proper unit in the database for a given context, the system is not designed to produce a suitable unit. Premise of the UTTS causes a lack of control over the synthesized speech. In order to better understand and gain control over speech synthesis, researchers have been investigated use of statistical parametric representations of speech in TTS. Source and filter model of speech is used in the statistical speech synthesis (SSS) techniques. Source and filters parameter are modeled with statistical generative models and model parameters are estimated during training session. In synthesis stage, by selecting the proper statistics for given context, best parameter sequence is generated from the models via a probabilistic approach.

SSS is not limited to a particular modeling. However, most of the works has been proposed use HMMs [1]. HTTS systems became the prominent example of SSS. Although, there are ongoing researches and developments on HTTS systems from different research groups, most referred and used system is an open-source project HMM-based speech synthesis system (HTS) [2][3] which is developed by HTS working group[1]. In order to make a consistent flow, HTS is considered as the reference while describing HTTS systems in general.

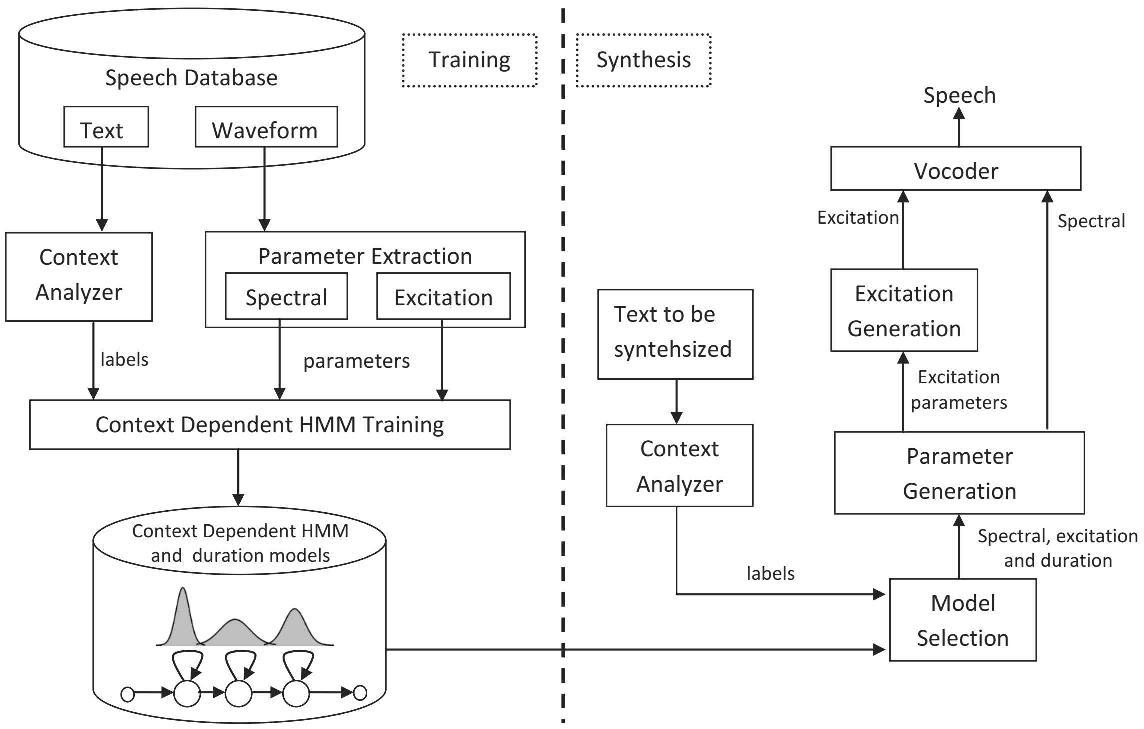

Figure 1: Overview of a HMM based Text to Speech System.

2.1 HTTS General Framework

HMMs were successfully used to model spectral parameters of speech in Automatic Speech Recognition (ASR) systems. Given that successes, using HMMs to create synthetic speeches seemed to be reasonable approach. Although, there had been studies on how to generate the speech from HMMs [4][5][6][7], Tokuda et all [8] first developed the trainable speech synthesis system based on continuous mixture HMMs in 1995. From then, HTTS has been gradually improved and became powerful technique.

General structure of an HTTS system is depicted in fig 2.1. In the training section of an HTTS system, spectrum and excitation parameters are extracted for each speech frame and associated with a descriptive label. Labels are extracted using a language processing tool, *Context Analyzer*. A typical label involves intonation and part of speech (POS) informations of different levels. An example is given in table 1.

[1] http://hts.sp.nitech.ac.jp/

Table 1: An example context information used in HTTS

Phoneme level	Preceding and succeeding phonemes
Syllable level	phoneme counts, accents and stresses of preceding, current, succeeding syllables. Position in the current word
Word level	Number of syllables in the preceding, current,succeeding words. Position in the current phrase
Phrase level	Number of syllables, words in the preceding, current, succeeding phrases. Position of the current phrase in the utterance.
Utterance level	Number of syllables, words, phrases in the utterance.
Intonation	stress, accent in syllables TOBI end tones

Using the labels, similar sounds are clustered into the same state by decision trees [9], using acoustic and phonetic questions. Then context depended HMMs are trained in a way similar to HMM based speech recognition. In synthesis stage for given text to be synthesized, decision trees obtained in the training are used to form corresponding sequence of HMMs, from the trained models. Using this sentence HMM, speech parameters are generated using an HMM based parameter generation algorithm, such as [10]. Synthetic speech is generated by a vocoder using generated parameters. HTS uses simple impulse/noise model for excitation which generates buzzy speech. To solve that problem, many systems employ a mixed-excitation approach where impulse and noise are mixed together in different bands [11]. In that case, mixing weights can be estimated and trained in the acoustic model training phase. Some other high quality vocoders have been also used to improve the quality, such as multi-band excitation [12] and STRAIGHT[13] with HTS.

2.2 Speech Parameter Modeling and Training

There are various alternatives for modeling the speech spectrum, such as mel-cepstrum and generalized mel-cepstrum parameters (MGC). Excitation can be modeled with an impulse train for voiced speech and random noise for the unvoiced speech. For the voiced speech, logarithm of the fundamental frequency (LF0) is extracted from each frame. One important problem with LF0 is that, although it has continuous values for voiced speech, it is not defined for the unvoiced speech. Therefore, a symbol indicating unvoiced speech is used for unvoiced frames. That makes the LF0 features a sequence of continuous-valued numbers and discrete symbols. In order to handle this in HMM framework, HTS uses a multi-space probability distribution (MSD) approach [14] to model state output probabilities for LF0. Changes in the pitch contour do not necessarily occur in synchrony with the spectral features. State-level alignments of those two sets of features can be very different. To avoid the mismatch, the spectrum and pitch features are fused together to create one feature vector o_t for frame t, in the training. Those two stream of features can be trained in a multi-stream training (MST)[2] framework in HTS.

Phonemes are modeled with N-state HMMs in the HTTS approach. As opposed to the HMM approach used in most current speech recognition systems, state durations can be modeled by a Gaussian distribution in the HTTS approach [15]. This allows the flexibility to set and change the phoneme durations explicitly. Spectral and pitch features[3] are typically modeled with a multivariate Gaussian distributions. Acoustic model parameters λ are trained with a maximum likelihood approach

$$\widehat{\lambda} = \arg\max_{\lambda} p(O|L, \lambda). \tag{1}$$

where O is the set of observation vectors, L is the corresponding set of labels and λ is the set of the model parameters. L is derived from the text of training data by *ContextAnalyzer*. Although those labels help accurately model the phoneme parameters, it is impractical to collect enough training for each possible combination of different contexts. Therefore, decision trees are used to cluster phoneme states

[2] S. Young et all, The Hidden Markov Model Toolkit (HTK), http://htk.eng.cam.ac.uk/

[3] pitch is modeled with MSD approach [14]. In MSD, space size is considered to be one for voiced regions and zero for unvoiced regions

that have different labels but that are automatically found to be similar [9].

Because there is no closed-form solution of Eq (1), expectation-maximization (EM) algorithm is used to find the ML estimation of the equation. The process is very similar to HMM-based speech recognition with slight differences. One of the difference is that both LF0 and spectrum is modeled and trained using MST. Other one is the context information used in the HTTS training. For speech recognition tasks, generally just phonetic level coarse-grain informations are enough since there is no re-generation of speech. In order to re-generate a speech (synthesis) as close to natural as possible, much fine-grained context informations should be taken into account. A typical context information used for a phoneme label is given in table 1.

2.3 Parameter Generation and Synthesis

Once the acoustic models are trained, they can be used to generate the spectrum and pitch parameters for a given text to be synthesized. *Context Analyzer* analyzes the sentence and finds the label for each phoneme and creates a label sequence. Then using this label sequence, corresponding sequence of HMMs is obtained. States of the HMMs are found using the labels and decision trees formed in the training. This sequence of continuous mixture of HMMs form the sentence HMM, Λ. The parameter sequence, $O = \{o_1, o_2, ..., o_T\}$, for pitch and spectrum can be generated by maximizing below posterior probability

$$\begin{aligned}
\widehat{O} &= \arg\max_{O} p(O|\Lambda) && (2)\\
&= \arg\max_{O} \sum_{n=1}^{Q} p(O, q^{(n)}|\Lambda) && (3)\\
&= \arg\max_{O} \sum_{n=1}^{Q} p(O|q^{(n)}, \Lambda)p(q^{(n)}|\Lambda) && (4)
\end{aligned}$$

where $q^{(n)}$ is any pdf sequence can be obtained from the mixtures of the ordered states of the Λ. Since maximizing O over all possible paths is a difficult problem, it is approximated as the maximizing over the one best path $\widehat{q}$. As aforementioned, HTS adopts the explicit modeling of state durations [15]. Hence, $\widehat{q}$ can be determined to maximize state duration distribution of the sequence.

$$\widehat{q} = \arg\max_{q^{(n)}} p(q^{(n)}|\Lambda) \tag{5}$$

Now, the sequence of (gaussian) pdf's, $\widehat{q} = \{q_1, q_2, ..., q_T\}$, is determined. Therefore, optimization problem turns into

$$\widehat{O} = \arg\max_{O} p(O|\widehat{q}, \Lambda) \tag{6}$$

$$= \arg\max_{O} \left(\prod_{t=1}^{T} P(O; \mu_{q_t}, \Sigma_{q_t}) \right) \tag{7}$$

Then the solution will be

$$\widehat{O} = \{\mu_{q_1}, \mu_{q_2}, ..., \mu_{q_T}\} \tag{8}$$

This is a staircase like trajectory. In order to generate smooth trajectories, optimization must be constrained with conditions which reflects the dynamics between frames. Tokuda et all[16] give couple of recipes to solve parameter generation from different steps, including solving Eq (2) with EM algorithm under dynamic constrains. Here, their recipe to solve Eq (6) under dynamic constrains is described. It is assumed that output vector at time t, o_t, is $3M$x1 vector consists of Mx1 static, speed and acceleration vectors.

$$o_t = [c_t^T, \Delta c_t^T, \Delta\Delta c_t^T]^T \tag{9}$$

A $3MT$x1 $\mathbf{O}$ vector which reflects the speed and acceleration features can be arranged as

$$\mathbf{O} = W\mathbf{C} \tag{10}$$

where $\mathbf{C} = [c_1^T, c_2^T, ..., c_T^T]$ is MTx1 vector and W is a $3MT$xMT block diagonal matrix which derives static, Δ and $\Delta\Delta$ from $\mathbf{C}$. Since the W is just a linear transformation, optimizing with respect to $\mathbf{O}$ and $\mathbf{C}$ are the same. Eq (6) can be rearranged as

$$\widehat{\mathbf{C}} = \arg\max_{\mathbf{C}} p(W\mathbf{C}|\widehat{q}, \Lambda) \tag{11}$$

In fact, it is better fits the aims that just static features are concerned at the end for vocoding. $\widehat{\mathbf{C}}$ can be found by solving the below equation

$$\frac{\partial \log(p(W\mathbf{C}|\widehat{q},\Lambda))}{\partial \mathbf{C}}|_{\mathbf{C}=\widehat{\mathbf{C}}} = 0 \tag{12}$$

and the solution is

$$W^T\mathbf{U}^{-1}W\widehat{\mathbf{C}} = W^T\mathbf{U}^{-1}\mathbf{M} \tag{13}$$

where $\mathbf{M} = [\mu_{q_1}^T, \mu_{q_2}^T, ..., \mu_{q_T}^T]$ and $\mathbf{U}^{-1} = diag[\Sigma_{q_1}^{-1}, \Sigma_{q_2}^{-1}, ..., \Sigma_{q_T}^{-1}]$. This can be solved efficiently by matrix decomposition, because of the band symmetric structure of the $W^T\mathbf{U}^{-1}W$. Eq (13) can also be solved [10],[8] in time recursive manner[17]. Solution of this will generate smooth trajectories imposed by the dynamic constrains. Moreover, variance characteristics also will be reflected in the resultant trajectory. It is an important property and it is utilized by the hybrid system proposed in this work. Generating smooth parameter trajectories, at state transition points, is good in terms of the lack of discontinuity. On the other hand, in naturalness aspect, these smooth trajectories does not have enough variation to reflect the natural phenomenons related to variations. From now on, this algorithm is referred as Maximum Likelihood based Parameter Generation (MLPG).

Once spectrum and pitch parameters are estimated, it can be used to synthesize the speech. In HTS, spectrum is modeled with mel-cepstral coefficients and speech is created by MLSA[18] synthesis filter. Excitation signal is created using the simple impulse/noise model which typically produces buzzy quality. However there are high quality vocoders which are used with HTS. Mixed-excitation approach where impulse and noise are mixed together in different bands is one of the techniques which is used with HTS [11]. In [11], they compute voicing strengths of the 6 bandpass filters in 0-8000 Hz band using normalized correlation. These 6 voicing coefficients of filters are also trained for each frame in the acoustic training phase. In synthesis stage, voicing coefficient are also generated from HMMs. Pulse train is the filtered using the coefficients and filters in voiced bands and white noise is filtered using the coefficients and filters in the unvoiced bands. Final excitation signal for the frame is the sum of filtered pulse and noise signals. Similarly, in [12], they used multi-band excitation technique for excitation modeling with HTS. There are many others used better models with HTS such as, in [13] STRAIGHT is used with HTS.

Spectrum modeling can also be enhanced. Researchers also use the LSP parameters to model the spectrum which is known to be robust to quantization errors. Subjective results from the researchers show that, the same quality, by subjective tests, can be achieved using less number of parameters when LSPs are used instead of mel-cepstral coefficients. One possible issue with LSPs is the stability. In order to avoid any possible problem, it should be ensured for all frames that HMM generated parameters should not cause an unstable filter.

CHAPTER III

HYBRID SYSTEMS

HTTS and UTTS come with their own Pros and Cons. Therefore, TTS researchers also investigated to use of both models together in different schemes to obtain the advantages of both techniques. Lots of different hybrid schemes and algorithms have been proposed by researchers. Here a comprehensive overview of the hybrid systems proposed in the literature is given. It is important to understand the motivation of these hybrid approaches to better understand the motivation of the proposed hybrid system.

HTTS systems is described in chapter 2. Here, in the first section, the UTTS systems is presented. Then, comparison between these two systems is given in the following section. Finally in the last section, an overview of the hybrid approaches in the literature is presented.

3.1 Unit Selection Based TTS (UTTS)

UTTS has been the dominant approach both in the industry and literature for decades. Despite the growing popularity of the SSS, it is still a major figure in TTS technology. UTTS based on a very intuitive, yet very successful idea: concatenating the recorded speech units to create synthetic speech. There were several types of UTTS systems in the literature. However, Hunt and Black [19] proposed a general framework and formulation of UTTS systems which became the standard over the years. They defined two costs, *target cost* $C^T(t_t, u_t)$ and *concatenation cost* $C^C(u_{t+1}, u_t)$. Then the total cost of unit sequence $U = [u_1, u_2, ..., u_T]$ for a given sentence $S = [t_1, t_2, ..., t_T]$ is given by

$$C(U,S) = \sum_{t=1}^{T} C^T(t_t, u_t) + \sum_{t=1}^{T-1} C^C(u_{t+1}, u_t) \tag{14}$$

$C^T(t_t, u_t)$ is used to measure how suitable the unit u_t is for target t_t. $C^C(u_{t+1}, u_t)$ measures how well the adjacent units u_{t+1} and u_t can be joined. Then a search is applied over all possible sequence to find the optimal sequence $\widehat{U}$ which minimizes the total cost $C(U,S)$.

$$\widehat{U} = \arg\min_{U} C(U,S) \tag{15}$$

Feature vectors used in the *target cost* calculations generally consist of phonetic and prosodic contexts. Spectral and acoustic features may be used in *concatenation cost* calculations to be used with acoustic distance measures. Researchers still look at what features to be used and how to weigh them in cost calculations.

Optimal size of the unit is not a resolved issue in UTTS. There is wide variety of unit sizes used in the works. Frame-sized [20], HMM state-sized [21], half-phones [22], diphones [23] and varying sized [24] are the some examples of different unit sizes that are used in UTTS systems. A general observation is that the longer the unit size, bigger the database is required to cover given domain [25].

3.2 Comparison of UTTS and HTTS

Although, UTTS and HTTS systems are both data driven approaches, they exhibit different characteristics due to their fundamental difference in foundations. Here they are compared in three prominent headlines: dependency on database and memory, flexibility of techniques and quality of produced synthetic speech.

3.2.1 Database and Memory Usage

As described earlier, UTTS systems are selection based systems. They don't aim at producing a unit if there is no proper one in the database. Hence, in order to adequately cover the units for given domain, considerable amount of recorded speech must be available to the system. On the other hand, for statistically model a context, small amount of data is enough. Besides, although it is expected to increase the accuracy, HTTS system does not necessarily need to counter observations for all the contexts. Clustering and statistical estimation causes a much robust system compared to UTTS in such situations. Reported HTTS systems in the literature uses less data in training compared to UTTS systems.

When memory is considered, main difference between UTTS and HTTS is that HTTS systems do not require to keep a database during synthesis. After the statistics of the models estimated in the training, just models are stored for synthesis in HTTS systems. On the other hand for UTTS systems, all data must be available to system during synthesis. Consequently, orders of magnitude less memory is needed to synthesize with HTTS systems. Such property makes HTTS systems attractive for embedded devices.

3.2.2 Flexibility

Most important advantage of HTTS is its flexibility. Parametric structure of HTTS creates a very suitable infrastructure for changing the voice characteristics, emotion and speaking style. Although UTTS systems has also ability to do conversion via voice conversion (VC) techniques [26], it is not high quality as expected from a UTTS system. Adaptation to a new voice is one of the prominent application of HTTS. Maximum a posteriori (MAP) [27] and Maximum likelihood linear regression (MLLR) [28] are two prominent techniques proposed in HTTS. Yamagishi [29] used MLLR-based adaptation techniques and presented average voice speech synthesis (AVSS) based on HMM-based speech synthesis. It can synthesize high quality speech with few minutes of target speaker's data.

3.2.3 Quality

UTTS systems use recorded speech waveforms in their units. However HTTS systems, because of its nature, uses parametric representation of the speech and produce vocoded speech. Speech production in HTTS are not yet optimal while there is no such concern in UTTS. Therefore, UTTS generates higher quality speech than HTTS. However, quality is a subjective term and highly dependent on human perception. The major problem of UTTS systems is sudden discontinuities that may occur in connection points where two adjacent units meet. These perceptual discontinuities annoys the listener. On the other hand, HTTS generates smooth parameter trajectories which eliminates the annoying glitches. Human perceptions is more sensitive to discontinuity. Therefore, in MOS tests, listener can prefer an good HTTS system over an UTTS system in which discontinuity frequency is noticeable. However the drawback of this smoothing effect of HTTS is that it suppresses the higher level information in speech. Over smoothing can cause a degradation in the naturalness of produced speech. This is a critical trade off which also motivates the hybrid TTS researchers. They aim at generating speeches as natural as in UTTS and as continuous as in HTTS.

In short, the problem of UTTS generated speech is possible discontinuities at connection points. Drawback of HTTS is the low quality of speech due to the speech production algorithm.

3.3 Hybrid Approaches

As compared in sec. 3.2, both techniques have strong and weak sides. Researchers have been proposed many hybrid schemes and algorithms to combine the strong parts of both. They generally focused on improving the quality while not giving too much importance to memory consumption. It is mainly because they already use an UTTS system as their baseline. However in this work, the aim is to improve the quality of baseline HTTS systems while keeping the memory usage in reasonable level. Depend on the motivation of hybrid scheme, it is possible to cluster these algorithms under couple of classes. Here, mainly the classification of [25] is followed.

3.3.1 HMM Guided Unit Selection

A Widely used approach in hybrid TTS literature is to use of HMMs as a guide to selection of units from database. This can be accomplished either using the HMM generated parameters directly as targets [30],[20],[31] or using the HMM likelihoods in the cost calculations [32], [33], [34]. In order to better introduce the idea and techniques used to realize the idea, the details of some systems are given.

Rouibia and Rosec [35] used HMMs to create acoustic targets for unit selection. Their feature vector consists of 12 mfcc coefficients, pitch and energy with their first and second degree derivatives. Using three state HMMs, they trained the models and applied standard ML based tree growing procedure for clustering. In synthesis stage, sequence of states are determined for each acoustic model from the given context. They predicted the duration of the each HMM state is adjusted proportional to $\frac{1}{1-p_s}$, where p_s is the probability of staying on state s. Given the duration and state sequences, they generated acoustic targets from the models using MLPG. These senone vector sequences are segmented into diphone-sized units for target cost calculations. They run a Dynamic Time Warping algorithm under the type I constraints and determine best N candidates for each diphone units. Optimal sequence is determined by dynamic search algorithm which tries to minimize the pitch distortion between last and first frames of adjacent units. They reported that their hybrid system received almost the same scores with their reference unit selection system in MOS tests.

Yang et all [36] used context dependent HMMs to be used in cost calculations. They trained HMM for spectrum and logarithm of F0, using the static, delta and acceleration components of the features. In order to overcome data sparseness problem, MDL based model clustering is performed and decision trees are created. Phone boundaries are aligned by Viterbi algorithm using the trained acoustic HMMs. Based on phone segmentation, duration model is also trained using HMMs. In synthesis stage, trained models are used in target cost formulation. Target cost is defined as a joint cost which aims at minimizing the KLD (Kullback-Leibler divergence) between target and candidate models and maximizing the likelihood of candidate unit sequence against the target models. Instead of conventional likelihood, they adopted Log Likelihood Ratio (LLR) approach. LLR of candidate unit u_n is given by

$$LLR(u_n) = \sum_{r \in \{s,f,d\}} w_{(r)} [\, \log P_{\Lambda^{(r)}}(\boldsymbol{O}_n^{(r)} | C^{(t)}) - \log P_{\Lambda^{(r)}}(\boldsymbol{O}_n^{(r)} | C_{u_n}^{(o)}) \,] \tag{16}$$

where $\{s, f, d\}$ represents the spectrum, F0 and duration respectively. w is the weight of each model. $\boldsymbol{O}_n$ is the extracted features of candidate unit u_n. $C^{(t)}$ is the context of the target and $C_{u_n}^{(o)}$ is the original context of candidate unit. Minus term in the formulation penalizes the candidates which are close to the center of the PDF which is assumed to be unnatural for given target model. They assigned the weights of each models empirically. Optimal unit sequence is determined by dynamic search algorithm and selected units are concatenated using cross-fade technique to create finale speech waveform. They reported that their system performed well in the similarity and naturalness tests of Blizzard Challenge 2011, while the intelligibility scores were not different from the others.

Pan et all [37] developed hybrid system for Mandarin which also uses HMMs in cost calculations. It is better to use more stable units in the modeling of prosody, especially for tonal languages like Mandarin. Therefore, they selected the syllables as their basic unit. For target cost calculation, they generated parameters from the trained HMM models of spectrum, F0, power and duration and used as reference. Euclidean distances between references and candidate unit are computed and target cost for the unit is given as weighted sum of four distances. They preferred the parameter generation approach over the likelihood approach, since it gives more flexibility for controlling the shape of prosody. Naturalness of synthetic speech is highly related to pitch contours. Hence, for better modeling of transitions of pitch contour between adjacent units, they trained a F0 dependency model using CART. F0 ending value and F0 ending derivative of previous syllable, F0 staring value and F0 staring derivative of current syllable are the four parameters of transition which are used in CART. During concatenation cost calculations, predicted values for these four parameters are used as reference and the distances to candidate units are calculated to give four distance scores for F0. Concatenation cost of spectrum is also calculated as distance based. Then the total concatenation cost is given as weighted sum of four F0 and one spectrum distance scores. They reported that all the weights are adjusted manually. Optimal unit sequence is obtained by viterbi algorithm and concatenated to create final speech.

Qian et all [38] used HMMs to assist to selection of the best speech segments (tiles) from unit database during synthesis. They trained HMMs for LSP, F0 and gain with traditional HMM training procedure. Then they refined the models using MGE training. During synthesis first, distances between HMM generated targets and original units are computed to determine suitable candidates for lattice creation. Therefore they defined frame based distance measures for LSP, gain and F0 and computed for each frame of candidate units and target. Computed distances then normalized by number of frames for that unit to get mean distances $\bar{d}_{F0}$, $\bar{d}_{gain}$, $\bar{d}_{LSP}$. In order to eliminate the weighting adjustment problem, they normalized the distances of all features to zero mean, unit variance standard distribution. Then, the distance between target and candidate unit is given as

$$d(u_t, u_c) = N(\bar{d}_{F0}) + N(\bar{d}_{gain}) + N(\bar{d}_{LSP}) \tag{17}$$

Only units that have the same labels with targets are considered for election. From those units the ones that are within a predefined distance from target are selected for lattice creation. They also limited the number of elected units to a maximum number. After determining the members of the lattice, Normalized Cross-Correlation (NCC) is used as matching score for each of adjacent waveform pairs. For each adjacent waveform pairs maximum NCC value and associated offsets are determined. Than the optimal unit sequence with the highest total cross correlation score is determined by viterbi algorithm. At the end, selected waveform pairs are shifted by determined offsets and concatenated with triangular cross-fading.

One common point of the described systems is that they all use ML estimations in their HMM training. However, Ling and Wang [39] introduced the Minimum Unit Selection Error (MUSE) training to improve their HMM-based unit selection system. In their previous systems they selected the phone-sized units from database by maximizing the joint probability of different HMMs. Since the HMMs are trained under ML criterion independently, setting the weight of different HMMs were done by manual operations. They also questioned about the relevance of using ML criterion for unit selection tasks. In order to address these two problems, they defined the Unit Selection Error which is the number of different units between selected and natural unit sequence for training data. They restructured the objective function of unit selection and deployed the Unit Selection error in the formula. Using the generalized probabilistic descent (GPD) algorithm [40], they iteratively optimized the models weights and HMM parameters by minimizing the Unit Selection error. Although their system is phoneme based, for this initial intend they used sentence level string error in their Unit Selection Error formulation. However, they showed by experiments that proposed systems was successful to decrease the unit selection error rate. Nevertheless, as stated by Zen et all [25], in speech recognition discriminative training systems based on fine-grain error measures generally gives better results than systems based on coarse-grain error measures. Hence, further improvement of MUSE training seems to be achievable.

3.3.2 Smoothing The Units With HMM

As mentioned in section 3.2, one of the important properties of HMM based systems is ability to create smooth trajectories. Using the premise of the HMM framework, Plumpe et all [41] proposed an technique to smooth out the units in concatenative speech synthesis. In order to fit in HMM framework, parametric speech model is adopted and extracted gain, source and filter coefficients are used to represent units. They applied their smoothing technique for spectral parameters of selected units while keeping the source parameters unchanged. LSP coefficients are proffered to model the spectrum. As discussed in Section 1, when HMMs states are used to model each frame with Gaussian random vector x = $[x \; \Delta x]$ with diagonal covariance matrices, under the i.i.d assumption, ML solution for a given sentence HMM $X = \{$x(1), x(2), ..., x(T)$\}$ can be reduced to minimization of equation (18)

$$E = \sum_{p=1}^{P} \sum_{t=1}^{T} \frac{(x_p(t) - \mu_p(t))^2}{\sigma_p(t)^2} + D\frac{(x_p(t) - x_p(t-1) - \Delta\mu_p(t))^2}{\Delta\sigma_p(t)^2} \tag{18}$$

D[1] is introduced here as weighting parameter for dynamic constraint. p represents the pth coefficient of the vector and t represents the vector in time instant t where P is the order of LSP and T is the total frame number Their idea is to use LSP vectors of real speech segment for μ instead of HMM means in order to retain naturalness inherit in concatenative speech synthesis [41]. Hence, minimization of equation will give smoothed LSP parameters of the units. As discussed in Section 2.3, this minimization problem is can be solved efficiently. Level of smoothing can be determined by adjusting D parameter. The final speech is produced using residuals and smoothed

[1] which equals to 1 in normal solution

LSP coefficients. In experiments they globally preset the D parameter. They determined state boundaries of the selected unit by HMM segmentation in the cluster of candidate units and set variance to unity. They reported that their approach was successful to reduce to discontinuities at unit boundaries.

Wouters and Macon [42] used similar smoothing idea and proposed $unitfusion$. In order to better smooth the boundaries, they used two different type of units, $fusion$ and *concatenation*. *concatenation* units are diphone sized units and gathered from nonsense words. $fusion$ units are phone sized units obtained from normal utterances. For a given sentence to be synthesized, first *concatenation* and $fusion$ units are selected. Then, each $fusion$ unit is centered at the joint point where two adjacent *concatenation* units meet. The dynamic of these two units are 'fused' with linear interpolation to create desired dynamic constraints in time, Δ^{time}, such that at the concatenation point only the dynamics of $fusion$ unit is applied whereas at the boundaries of the $fusion$ unit, the $fusion$ unit has no contribution to Δ^{time}. They also wanted to control the distance of LSP features with respect to each other while smoothing. Using the same linear fusion procedure, they also created desired dynamic constraints in distance Δ^{dist}. Then, they defined the error function as

$$\begin{aligned} E = \sum_{p=1}^{P}\sum_{t=1}^{T}(x_p(t) - f_p(t))^2 \\ +D1[(x_p(t) - x_p(t-1)) - \Delta^{time}(p,t)]^2 \\ +D2[(x_{p+1}(t) - x_p(t)) - \Delta^{dist}(p,t)]^2 \end{aligned} \tag{19}$$

$\Delta^{time}(p,t)$ and $\Delta^{dist}(p,t)$ are desired time and distance dynamics (derivatives) at time t for p^{th} coefficient. $f_p(t)$ is the original p^{th} LSF coefficient of *concatenation* unit at time t. $x_p(t)$ is the smoothed LSF coefficients. This error minimization can be solved similar way with HMM based parameter generation which is discussed in Section 2.3. However, in this formulation LSP coefficient are dependent on each other via third term. Hence, set of equation should be solved simultaneously for all LSP coefficients. They empirically set $D1$ to 20 and $D2$ is set to proportional to inverse square of $\Delta^{dist}(p, t)$. They reported that unit fusion system received better objective and subjective results than linear smoothing and default concatenation system.

One drawback of these systems is that they smooth the spectrum parameters while keeping the source parameters unchanged. This can cause degradation in the quality when there is a mismatch between smoothed spectrum parameters and source parameters.

3.3.3 HMM-driven systems

Unlike the tradition, these type of systems use natural segments or natural parameters of speech to improve their HMM based TTS system.

Excitation signal is important in generating natural sounding speech. Raitio et all [43] used the closest pre-stored excitation signal to the synthetic excitation signal to improve the quality in their HTTS system. For this purpose, extracted six speech parameters and glottal source pulse from each frame of speech by intensive analysis. List of these features given in table 2. They trained HMMs for these six parameters using standard HTS tools[2]. They also created a database of glottal source pulses and associated each pulse with the six parameters, pulse descriptor, extracted from the same frame. During synthesis, best glottal source pulse for each HMM in sentence HMM is selected by minimizing the joint cost of target and concatenation costs. The target cost is defined as the error between pulse descriptor and HMM generated parameters. The concatenation cost is defined as RMS error between the down-sampled candidate pulses. The synthetic speech is obtained by lpc vocoder using selected source pulses and HMM predicted vocal tract filter coefficients. They reported that quality and similarity scores of the hybrid system was better than default system.

[2] http://hts.sp.nitech.ac.jp/

Table 2: Feature used in HMM training

Feature Name
F0
Energy
Harmonic to noise ratio
Harmonic magnitudes
Voice source spectrum
Voice tract spectrum

Gonzalvo et all [44] also used natural segments to improve their HMM based system. They extracted 39 order mel-cepstral coefficients and 5 excitation parameters of sub-bands using STRAIGHT [45]. Along with logarithm of F0 and duration They trained SI models of these four parameters then adapted to target speaker. In the training they also constructed a pre-selection module which uses the decision trees created by HMM training. Given the context of the phoneme to be synthesized, this module gives the corresponding cluster of candidate units. In synthesis stage, a HMM-based unit selection module which is a simplified version of [46] determines the optimal unit sequence for given text using the pre-selection module. Meanwhile, another module generates the vocal tract, F0 and excitation parameters using HMM based parameter generation. They introduced Local Minimum Generation Error (LMGE) criterion which is derived from MGE [47] to minimize the error between parameter generated from HMM and optimal unit sequence. In order to align the HMM generated vocal tract parameters and optimal unit sequence, DTW is used. After the alignment, mean and the variance of HMM model is updated using the below formulas

$$\mu_{i,j,k} = \hat{\mu}_{i,j,k} - \frac{1}{N_{i,j}} \sum_{f=1}^{F} (w(f)).D_{f,k} \tag{20}$$

$$\sigma^2_{i,j,k} = \hat{\sigma}^2_{i,j,k} - \frac{1}{N_{i,j}.\hat{\sigma}^2_{i,j,k}} \sum_{f=1}^{F} (w(f)).D_{f,k}.(\hat{c}_{f,k} - \widehat{\mu_{i,j,k}}) \tag{21}$$

where $D_{f,k}$ is $\hat{c}_{f,k} - c_{f,k}$, $N_{i,j}$ is the total number of samples in j^{th} distribution of i^{th} state, F is the number of frames of the unit, k is the order of the coefficient of the PDF. $w(.)$ is the frame based weight function reaches near to 0 for boundary frames and near to 1 for middle frames. After the update, parameter generation procedure is repeated with new model and enhanced mel-cepstral parameters are produced. Final speech is generated by MLSA filter.

3.3.4 Mixing The Segments

In their Cereproc's system, Aylett and Yamagish [48] offered use of an auxiliary unit database which is created by HMM based speech synthesis to help the natural unit database. They use this synthetic database additionally to the natural one when data is sparse and concatenation costs are high. Then these mixed units are concatenated seamlessly in their system.

Pollet and Breen [49] gives an general overview of their Multiform Segment (MFS) synthesis systems. A MFS sequence is a mixed sequence of template and model segments which can be considered as original and HMM generated units respectively. Idea is to use of model segments for the parts of speech where in human perception the diminishing of quality is not an issue. In order to automatize the selection of segments, a probabilistic framework is constructed in which probability of a segment being a template or a model is assessed by expertise in Speech Perception. As an example for a segment of a stressed vowel, probability of template and for a segment of a nasal, probability of model is expected to be higher. For synthesis first, two sequences are generated, optimal template sequence and optimal model sequence. Then the optimal multiform segment sequence which maximizes the total probability is selected by viterbi algorithm from these two sets. Model segments in the optimal multiform sequences are converted to waveforms (template segments). Then template segments are concatenated to create final speech.

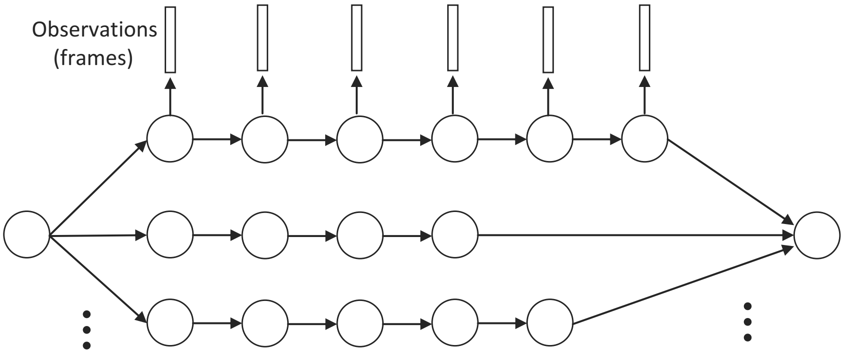

Figure 2: A unit-HMM network with N paths where only first three of them showed. Each path represents a unit in the database. A state in a path corresponds to one frame of the unit. Each state modeled with Gaussian and the observation vector -only for the first path showed here- is the corresponding frame of the unit.

These kind of systems offers to solve sparsity problems. However, Although proposed algorithms try to address, it is still a problem that switching between the natural and synthetic in produced speech which annoys people.

3.3.5 Unification

Taylor [50] investigated the unification of two approaches. He described a very general HMM framework to establish a common base for HMM based and unit selection based synthesizers. To be able to represent an unit selection system in a HMM framework, spectrum and pitch representation is used for the units rather than 'perfect' speech waveforms. It causes degradation in the generated synthetic speech, but it seems to be a fair price to pay for the purpose of unification. He defined a unit-HMM network for the units which has the same context in the database. Therefore whole database is clustered by all context such that for each context there is a one unit-HMM network. Topology of a unit-HMM network with N units is given in figure 2. Each path in this network represent one unit in the database and each state in the path represents exactly one frame of the unit. First and last state are placed for convenience to show the concatenation with previous and next unit-networks. All the transition probabilities are set to 1, except the probabilities of the branches of the first state which are 1/N. Mean of the each state is the exactly value of the state and diagonal covariances matrix has very small values. Therefore it can be interpreted that each unit-HMM network just memorizes the data without any information loss. Hence, synthesizing the speech using these unit-HMM networks will produce a speech almost as the same as the standard unit selection. Taylor also showed that by merging the pair of states one by one any kind of HMM topology can be created to be used in standard HMM based synthesis.

CHAPTER IV

PROPOSED HYBRID SYSTEM

In chapter 3.3, an overview of the hybrid systems in the literature is given. Looking at those systems, it can be seen that the most of them are focused on improving the quality of a baseline Unit Selection systems with the assistance of SSS techniques. As oppose to this tradition, in this work, a hybrid system to boost the performance of a baseline SSS system is proposed. In the existing hybrid systems, small memory footprint advantage of the SSS system is lost since both a Unit Selection and SSS system are used together. A key novelty in this work is a hybrid system that keeps the voice database size small while improving the quality of the HTTS system by utilizing the opportunities of morphological languages. Although, the idea is applied to Turkish TTS in this work, it can also be used for other morphologically rich languages such as Finnish, Estonian and Czech.

Turkish is a morphologically rich language and many different words can be generated from the same root word by using a limited set of morphemes. Given a typical Turkish utterance, a significant number of the words contain one or more morphemes. Moreover, ignoring silences, around one fourth of the speech is composed of morphemes. In the proposed system, a database of the most frequently occurring morphemes is created in training. In synthesis, best fitting morphemes are selected using the proposed morpheme selection algorithms. Then, the selected morphemes are used in HTTS within the proposed parameter generation algorithms.

This chapter is organized as follows. First, the development of the baseline HTTS system for Turkish is described. Second, proposed algorithm is introduced. Third, details of two different morpheme selection algorithms are presented. Fourth, details of parameters generation algorithm for the proposed hybrid system are given.

4.1 Baseline System Description

In this work, an HTTS system is trained for Turkish voice with HTS to be used as a baseline system. In order to train a voice, the labels of each phoneme in the utterances should be created. Labels are directly derived from the transcription of the speech. Therefore, it is important to convert the non-standard words in the transcriptions to written forms. Several consecutive text processing algorithms are used within a *Normalizer* module. This module addresses the issue of numbers, dates, punctuation etc. After the text is normalized, *ContextLabeler* creates a label file for each utterance in which each line is the label of one phoneme in the utterance.

4.1.1 Label Structure

Transcription sentences are split into phrases, words, syllables and letters to be used by *ContextLabeler* to create labels. Following context information is extracted and used.

- Phoneme level: Two preceding, current, two succeeding phonemes, position in the current syllable (forward, backward), affix information.
- Syllable level: Number of phonemes in the preceding, current, succeeding syllables. Position in the current word (forward, backward). Stress-flag. Distance to the (previous, succeeding) stressed syllable.
- Word level: Number of syllables in the (preceding, current, succeeding) words. Position in the current phrase.
- Phrase level: Number of (syllables, words) in the (preceding, current, succeeding) phrases. Position of the current phrase in the utterance.

- Utterance level: Number of syllables, words, phrases in the utterance. A flag for the sentence type (question, exclamation, other).

4.1.2 Pronunciation and Stress Modeling

Turkish has one-to-one relationship between its graphemes and phonemes for most cases. However, there are exceptions. To model (grapheme-to-phoneme) G2P mappings of Turkish, a CART model using a pronunciation lexicon [51] is trained. Then, G2P conversion is done using the CART trees.

First, a list of alternate phonemes for each grapheme are created. For example, the grapheme ”a” may be pronounced as /a/ or /aa/. These alternate mappings are then used to align words and their pronunciations in the lexicon. After alignment, a CART tree is generated for each grapheme using the context of four preceding, current and four succeeding graphemes. Those trees are then used for predicting the correct phonemes during synthesis.

Turkish stress markers typically follow a limited number of rules. However, there are exceptions. For example, the word ”ordu” can mean ”army” or the city of Ordu in Turkey. Stress is placed in different syllables depending on which one of the meanings is used in a context. Similarly, part-of-speech (POS) tagging is required to correctly place the stress markers in a word. Although one of the most advanced tools for locating stress markers in Turkish is used, the tool sometimes simply returns multiple alternatives and the algorithm has no way of picking which one to use[52]. In developing the system, stress markers that follow the linguistic rules [51] are used.

4.2 Overview of the Proposed System

An overview of the training and synthesis phases of the proposed system is shown in Fig. 3. In the training phase, HMM models and a decision tree are generated for the target speaker using speaker dependent training with HTS. Then, a morphological analyzer is used to analyze the words in the speech database. To create a morpheme

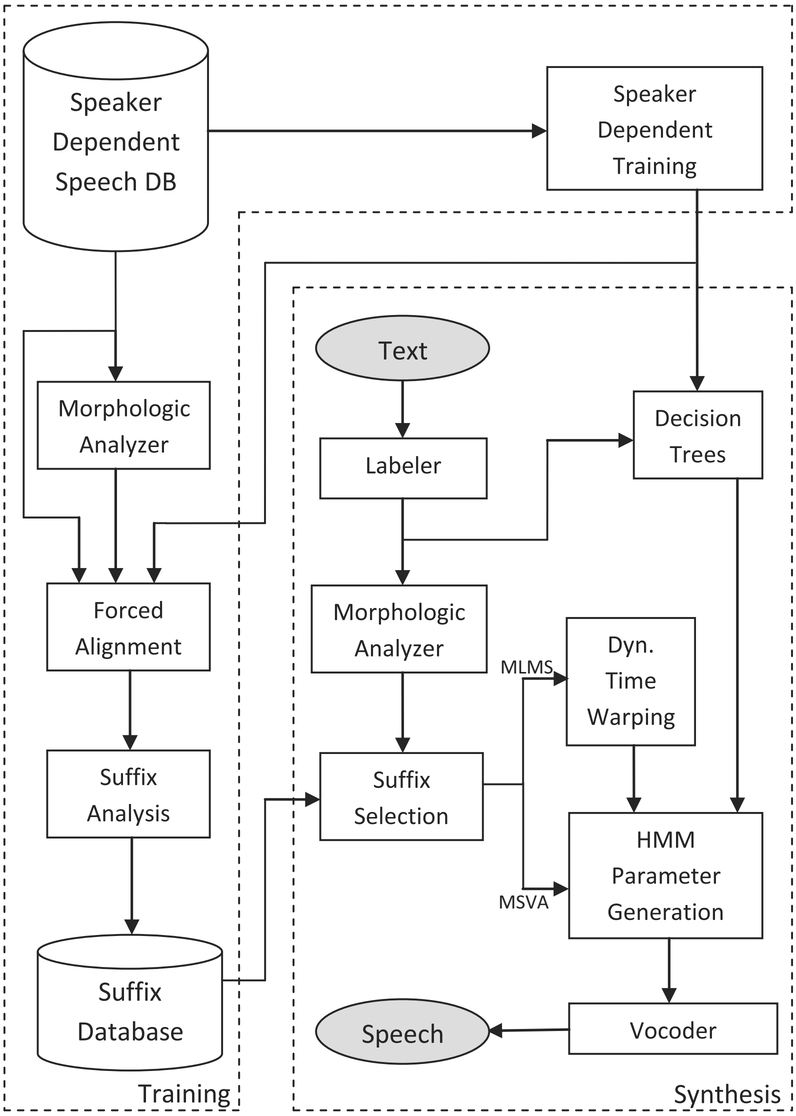

Figure 3: Overview of the proposed hybrid system.

database, waveforms that correspond to the morphemes labeled by the morphological analyzer should be extracted from speech. Forced alignment is used with the speaker-dependent HMM models to align text and speech data. Morphemes are then extracted from the speech signal using the alignment information.

Morphemes are parametrized using LPC analysis and only the LSF and pitch parameters are stored. Besides those parameters, each entry in the morpheme database contains a flag that indicates contextual features such as the presence of silence at the right context of the morpheme (phrase ending) and another flag that indicates the presence of stress on the morpheme. More detail about the context features are given in Section 4.3.1. Moreover, beginning and end times of the state-level segments are also stored in the database.

In the synthesis phase, HMM models that correspond to the input text is determined using the decision tree. Input text is analyzed using the morphological analyzer; and, for each morpheme in the text, the best fitting morpheme is selected using the algorithms described in Section 4.3.2 and Section 4.3.3. The statistics predicted by the decision tree and the parameters of the unit that is selected from the morpheme database are combined together and fed into the parameter generation algorithms described in Section 4.4.2. Finally, the parameter sequences generated are used in an LSF vocoder to synthesize speech.

The morphological analyzer described in [52] is used here. The analyzer generates the root word and the morphemes of a given word. Both inflectional and derivational features of the morphemes are produced. Nominal features (case, person/number agreement, possessive agreement) and verbal features (tense, aspect, modality, and voice) are indicated with special tags. An example is the word

```
kazanabilecegini (k a z a n)kazan+Verb+
Pos(a b i l)^DB+Verb+Able(e dZ e G)^DB+
Noun+FutPart+A3sg(i)+P3sg("n i)+Acc
```

In the word ”kazanabilecegini”, ”kazan” is the root word and the rest of the word are the morphemes. Derivational phonemes are indicated by the DB tag. Note that after every derivation, the new type of the word is also shown. For example, the root word in this example is a verb, and it is still a verb after adding the derivational morpheme ”abil” which indicates positive polarity. Stress in the morphemes are shown with the ” sign. A3sg is an inflectional marker that indicates the person/number agreement (third person singular) here.

The analyzer sometimes returns multiple alternatives. A morphological disambiguation tool can be used to resolve such cases [53].

4.3 Morpheme Selection Algorithms

In unit selection based TTS systems, typically, both target cost and concatenation cost are used in selection. Target cost is used for selecting units that are good fits for the target positions in the utterance. Concatenation cost is used for selecting units that flow naturally without abrupt changes when concatenated. In this work, firstly, the behavior of the system when maximum-likelihood based target cost is used without any concatenation cost for selection is investigated. It is found that likelihood-based target cost computation is not very effective mainly for two reasons. The first reason is that likelihood based target selection tends to significantly favor smooth trajectories which reduces the variability and, therefore, naturalness. The second reason is that different morphemes are selected for pitch and LSF parameters in the ML approach. However, the mismatch between the two creates significant degradation in quality.

Another observation with the maximum-likelihood based morpheme selection is that the most significant improvement in the listening tests were perceived during stressed syllables. Therefore, it is found that it is important to design a hybrid system that can model well the rapid pitch variations during stressed syllables while not having discontinuities at the morpheme boundaries.

Using the lessons learned with the maximum-likelihood based morpheme selection approach, another morpheme selection algorithm is developed. The new algorithm uses pitch concatenation costs instead of the ML-based target costs. Target costs are replaced with a decision-tree based morpheme filtering algorithm. The morphemes that survive the filtering process are then selected using a Viterbi algorithm. LSF and pitch features are selected from the same morpheme which reduced the need for smoothing, increased clarity and naturalness compared to the ML-based approach. The decision tree based pre-filtering and the two morpheme selection algorithms are described in detail below.

4.3.1 Decision Tree based Prefiltering

Similar to HMM states, morphemes can be clustered using decision trees depending on their contexts. One can use the same syllable, word, and phrase level features given in Section 4.1.1 during the decision tree building process. However, because there are not too many morphemes in each morpheme class, the questions are restricted to linguistically most relevant ones. The questions that are used here are shown in Table 3.

In the decision tree approach, for each morpheme class, all instances of the morpheme are pooled together at the root node. Then the root node is split using an entropy and minimum number of occurrence criteria. To calculate entropy, distributions of each morpheme instance are needed. Two algorithms are investigated to model the distributions of morphemes. In the first approach, the distributions of the pitch features for each state i, $(\mu_{p,i}^{(j)}, \Sigma_{p,i}^{(j)})$ of each morpheme instance j can be be concatenated to obtain the super vectors $\mu_p^{(j)} = [\mu_{p,1}^{(j)}\mu_{p,2}^{(j)} \ldots \mu_{p,N_s^{(j)}}^{(j)}]$ and $\Sigma_p^{(j)} = diag([\Sigma_{p,1}^{(j)}\Sigma_{p,2}^{(j)} \ldots \Sigma_{p,N_s^{(j)}}^{(j)}])$ where the $diag(.)$ operator creates a block diagonal matrix with $\Sigma_{p,i}^{(j)}$ at the diagonal position i. However, this approach does not work well in practice because the distributions of states heavily favors smooth transitions and cannot model rapid pitch variations well. Therefore, modeling such variations, for example in stressed morphemes, with those distributions results in inaccurate clustering.

For building a more accurate decision tree, the following distribution estimation algorithm is proposed. Each morpheme is state-aligned with the HMM states. If state i is aligned with $N_{f,i}$ frames, the feature vector at frame $\lfloor N_{f,i}/2 + 1 \rfloor$ is used

Table 3: Linguistic Questions Used in the Decision Tree based Clustering of Morphemes

Sylable-Level	Stress: What is the stress level of the syllable that contains the morpheme?
Word-Level	Position in the word: Is the morpheme at the end of the word?
Phrase-Level	Position in the phrase: Is the word containing the morpheme at the end of the phrase?

to represent the mean vector for that state where $\lfloor . \rfloor$ is the floor operator. Because there are typically not enough samples to estimate the covariance within a state, $\Sigma_p^{(j)}$ is used as the covariance matrix. This approach picks the mean vectors from the natural morphemes. Therefore, smoothing problem is completely eliminated and comparison with real vectors become possible as opposed to synthetic mean vectors used in the first approach. The second approach to probability distribution is used in the proposed system.

4.3.2 Maximum-likelihood based Morpheme Selection (MLMS)

When synthesizing an utterance u, the set of morphemes $\{m^{(k)}\}$ in the utterance are determined using a morphological analyzer where $k = 1, 2, ..., K$, and K is the total number of morphemes in the utterance. For the j^{th} morpheme, the initial set of available units in the database is denoted by $\{M_1^j\}$. The initial set is generated using the decision-tree based pre-filtering described in Section 4.3.1. The goal in pre-filtering is to reduce the set of available units that best match the context of the phoneme.

Candidates in the morpheme database are represented by two features, LSF and pitch. Given a synthetic morpheme and a target cost, a candidate may be best fit for one of its features while not optimal for other feature. Therefore, targets for LSF and pitch parameters are selected independently. Moreover, the cost calculation for those features are also different. The proposed unit selection algorithm uses a maximum likelihood (ML) criterion as the target cost. However, it is found in the experiments that the ML criterion does not always return a proper morpheme that has a good concatenation cost. To reduce the possibility of an artifact, for the pitch parameter, two heuristics is used to filter out the set of available units in the database for a given morpheme. The heuristics are described below.

During parameter generation, the pitch trajectory of the selected unit is time-warped so that it can fit into the synthetic duration estimated with SSS. In the experiments, expanding the pitch trajectory did not cause any audible artifacts. However, compressing the pitch trajectory occasionally caused sudden pitch changes which are perceived as artifacts by the listener. To avoid that problem, the units in $\{M_1^j\}$ that are R_d percent longer than the synthetic duration of the morpheme M^j are filtered out. The reduced set of units after filtering is denoted by $\{M_2^j\}$.

Finally, an weighted ML (w-ML) criterion is used to select the morpheme from $\{M_2^j\}$. In the proposed w-ML method, for each unit $M_{j,k}$ from $\{M_2^j\}$, the average weighted log-likelihood is computed by

$$\begin{aligned} L_{j,k} = \frac{1}{N_{j,k}} \sum_{s=1}^{S} w_{j,k}^{s} \sum_{f=1}^{N_s} log \left[\frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma_s|^{1/2}} \right] \\ -\frac{1}{2}(X_s^f - \mu_s)^T \Sigma_s^{-1} (X_s^f - \mu_s) \end{aligned} \tag{22}$$

where $N_{j,k} = \sum_{s=1}^{S} w_{j,k}^{s} N_s$,

$$w_{j,k}^{s} = \begin{cases} \gamma_{j,k}^{s}/f_m & \text{if } s \leq 2 \text{ or } s \geq (S-1) \\ 1 & \text{otherwise} \end{cases} \tag{23}$$

and

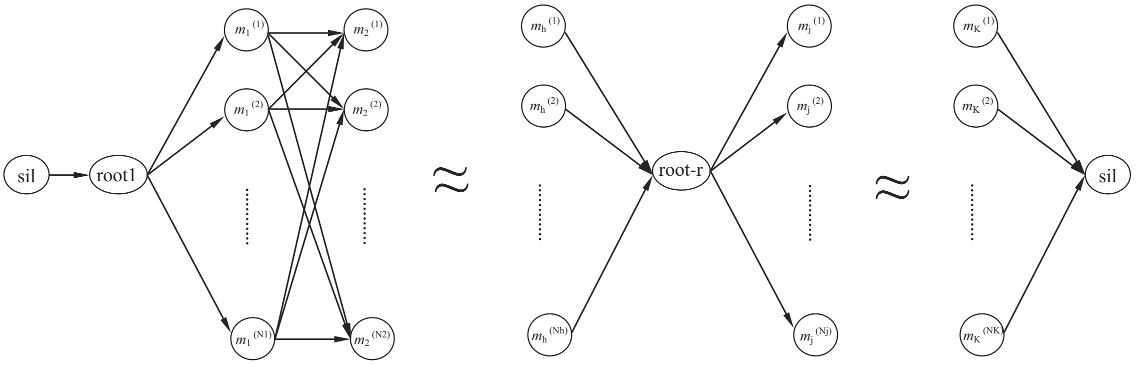

Figure 4: An illustration of the search graph used in the Viterbi algorithm for selecting the morphemes. Root words are synthesized with the SSS system so there is only one alternate for the root words. Different number of natural units exist for the morpheme units. More than one morpheme can follow a root word.

$$\gamma_{j,k}^{s} = \begin{cases} 5 & \text{if } N_s \leq 5 \\ N_s & \text{if } N_s > 5 \end{cases} \tag{24}$$

S is the total number of states, N_s is the total number of frames in state s, Σ_s is the covariance matrix and μ_s is the mean vector in state s. X_s^f is the f^{th} observation of state s. X_s^f contains static, delta, and delta-delta features. The numbers 2 and 5 are found experimentally. The w-ML measure helps smooth out the concatenation points by assigning higher weight in likelihood computation to states around those points.

The heuristics used in calculating the cost function for pitch are not used for the LSF features. Thus, the ML cost is the only criterion in selecting the appropriate morphemes for LSFs. Since pitch and LSF features are selected from different candidates, duration of them are different. They time-warped to match the synthetic duration. Warping of the features with huge duration differences may distort the signal. This is avoided by filtering due to duration difference in the pre-filtering method described above.

4.3.3 Morpheme Selection Using the Viterbi Algorithm (MSVA)

In conducted experiments with the MLMS approach, two important phenomenons are observed: the ML approach tends to pick morphemes that change smoothly especially for the LSF features. The over smooth feature trajectory problem that is a result of the ML-based parameter generation algorithm becomes an issue when picking the morphemes using the ML approach. In the listening tests, it is observed that, almost all of the gain was related to capturing the sudden variations in the pitch contour with the stressed morphemes which could sometimes be modelled accurately despite the smoothing effect of the ML approach. Details of the experiments with the MLMS approach are given in Section 5.1.2.

Because the ML-based target cost favors smooth trajectories, and pitch variations make the most impact in listener preference, a second algorithm is proposed here that is focused only on concatenation costs of pitch trajectories without restricting the shape of the pitch contours other than the pre-filtering method described above. One can use the concatenation cost for the LSF features and pick different morphemes for the LSF features as is done in the MLMS algorithm. However, it is found that using the LSF contours from the same segments where the pitch contours are selected provides good performance and reduces the computational effort. Thus, pitch and LSF contours are obtained from the same morpheme in the MSVA approach.

In the first step of the MSVA algorithm, synthetic pitch contour is generated for a given utterance. Because pitch is defined only for voiced speech, linear interpolation is used between the voiced segments to obtain a continuous pitch contour. There is only one candidate for the root word position. However, for each morpheme position in the utterance, there are many alternative morphemes as shown in Fig. 4 where the search space is organized as a graph and each node represents either a morpheme or a root word. The best path with the lowest total concatenation cost through the graph is selected with the Viterbi algorithm. Concatenation cost between i^{th} candidate of morpheme k, $m_k^{(i)}$, and j^{th} candidate of next morpheme, $m_{k+1}^{(j)}$, is defined as weighted Euclidean distance

$$d_{con}(m_k^{(i)}, m_{k+1}^{(j)}) = \sum_{f=0}^{F} w(f)|P_k^{(i)}(N_i^f - f) - P_{k+1}^{(j)}(1+f)| \tag{25}$$

$w(f)$ is the weight of the frame at a distance f to the boundary. $P_k^{(i)}(f)$ represents the f^{th} frame of $m_k^{(i)}$ and $N_k^{(i)}$ is the number of frames of $m_k^{(i)}$, Concatenation cost of morphemes and root words are computed similarly.

Using the distance metric above and the Viterbi decision rule, the selected morphemes $\mathcal{M}$ for a given utterance is

$$\mathcal{M} = \arg\min_{\mathcal{M}} \sum_{j=1}^{J-1} d_{con}(s_j, s_{j+1}) \tag{26}$$

where J is the total number of segments (morphemes and root words) in the utterance and s_j denotes the j^{th} segment. If the last morpheme occurs at the end of the sentence the concatenation cost with the following silence segment is not taken into account because last morphemes are selected from the available morphemes at the end of sentences as discussed in the pre-filtering section above.

During parameter generation, LSF and pitch features are time-warped to fit into the synthetic durations in the MLMS approach because they are selected from different morphemes. In the MSVA approach, pitch and LSF features are obtained from the same morpheme and time-warping is not needed which was found to be beneficial in the listening tests. However, some care is required in morpheme selection because gross mismatch in duration can hurt the naturalness of speech. Here, it is required that the selected morphemes to be at least as long as the synthetic ones and not longer than ζ_d times the synthetic morpheme durations where ζ_d is set experimentally.

There are two commonly used techniques to calculate the target costs. In one approach, the HMM parameters can be used to calculate the likelihood of a feature segment as done in the MLMS approach. The other approach is to use the distance

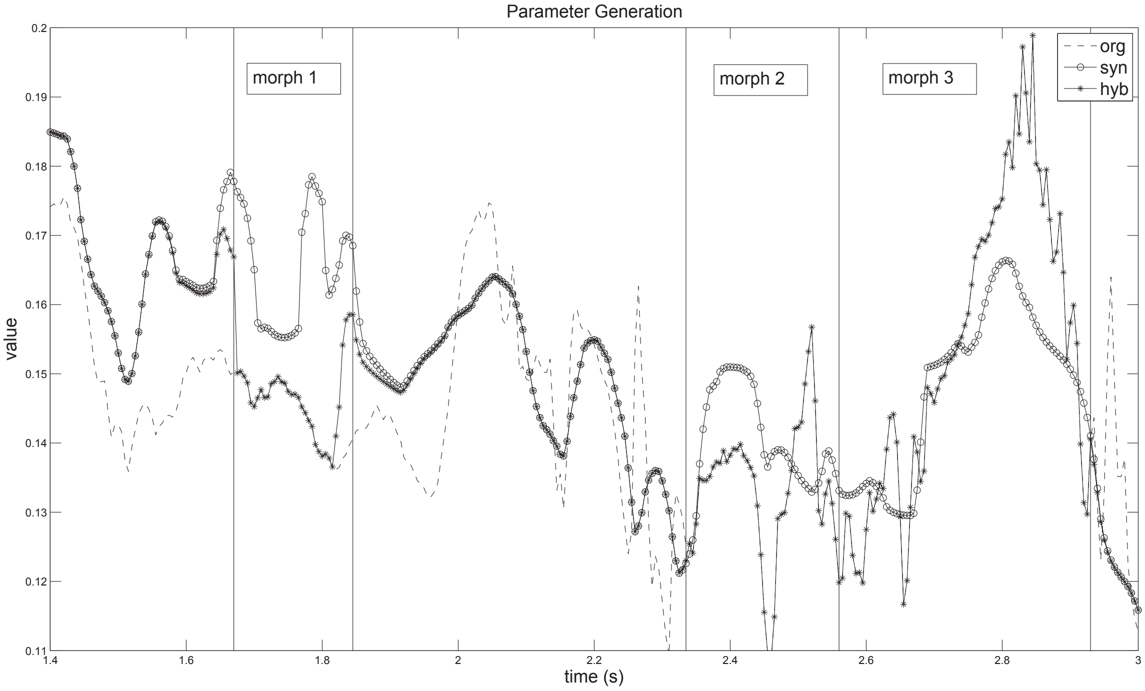

Figure 5: Parameter trajectories for a spectral feature using the baseline SSS system and hybrid system compared with the recorded speech. Hybrid system follows the natural trajectory during the morphemes and synchronizes back with the synthetic trajectory when a morpheme is not available.

between the features of the natural morpheme units with the parameters generated with the HMM. Both of those approaches are biased in favor of smooth trajectories. Here, the target cost is not directly included in the Viterbi search to avoid the smoothing effect. However, decision-tree based pre-filtering, described below, is used to reduce the set of possible morphemes at each position to potential candidates that are most appropriate for the context.

4.4 Hybrid Parameter Generation

4.4.1 Segment Based Constrained Parameter Generation (SBCPG)

In sec. 2.3, parameter generation problem is solved under dynamic constraints. A similar approach can be used to formulate parameter generation problem in a segment based hybrid system. In a segment based hybrid system, in proposed system segments are morphemes, natural speech segments are scattered throughout utterances while synthetic speech is used for the rest. In such system, k^{th} segment of features $c_{(k_m,k_n)}$ from frame k_m to frame k_n can be constrained to be equal to natural speech segments $c_{nat,k}$ during the parameter generation process. If there are a total of K such segments scattered across an utterance, hybrid parameter generation can be formulated as the constrained optimization problem

$$\widehat{c}_h = \arg\max_{c} p(Wc|\widehat{Q}, \lambda). \tag{27}$$

such that

$$A\widehat{c}_h = c_{nat} \tag{28}$$

where $c = [c_{(1_m,1_n)};\ c_{(2_m,2_n)};\ \dots\ ;\ c_{(K_m,K_n)}]$, $c_{nat} = [c_{(nat_1)};\ c_{(nat_2)};\ \dots\ ;\ c_{nat_K}]$, and A is a design matrix. Each row k of A, a_k, corresponds to k^{th} natural speech segment. To perfectly generate the K natural segments, $a_k = [0_{1.(k_m-1)}\ 1_{1.(k_n-k_m+1)}\ 0_{1.(N.25-k_n)}]$. Using the Lagrange multiplier γ, the parameter generation problem becomes

$$\widehat{c}_h = \arg\max_{c} p(Wc|\widehat{Q}, \lambda) - \gamma(Ac - c_{nat}). \tag{29}$$

Solution to Eq. 29 is [54]

$$\widehat{c}_h = \widehat{c} + (W^T U^{-1} W)^{-1} A^T \gamma \tag{30}$$

where

$$\begin{aligned} \gamma = (A(W^T U^{-1} W)^{-1} A^T)^{-1} c_{nat} \\ -(A(W^T U^{-1} W)^{-1} A^T)^{-1} A \\ (W^T U^{-1} W)^{-1} W^T U^{-1} M \end{aligned} \tag{31}$$

An example trajectory created with SBCPG is given in fig. 5. It can be seen that transition between natural and synthetic segments can be unnatural because of the strict constraints in the natural segments. In [54], they proposed and iterative algorithm to handle the transition boundary problem. However, in proposed system more intuitive approach is adopted to handle the problem which is described in sec 4.4.3.

4.4.2 Proposed Hybrid Parameter Generation

In the MSVA method, morpheme selection is based on the concatenation cost of the pitch contours while both LSF and pitch contours are used during the hybrid parameter generation phase. In the MLMS method, pitch and LSF contours are selected independently using the ML approach. After morphemes are selected, the next phase is to generate LSF and pitch parameters using the selected the morphemes. The algorithm described in Section 4.4.1 is used to generate the pitch contours. However, more care had to be taken with the LSF parameters to avoid discontinuities.

Although the Viterbi-based approach takes into account the concatenation cost for the pitch contours, that is not the case for the LSF features. Therefore, LSF discontinuities can occur for the with the MSVA approach. Similar problems have been observed with the MLMS approach since it does not take the concatenation cost into account. The proposed approach to solving this problem is to relax the constraints at the B_s number of initial frames and B_s number of final frames in the morpheme. The constraints are relaxed for those transition frames as follows. Morphemes are first state aligned and the mean vectors of the emission pdf's of the transition states (the ones that contain the transition frames) are then replaced with the LSF vector in the middle of the state in the state-aligned original morpheme. This approach encourages the system to pass through the original LSF vector in the middle of the transition states. However, it also lets the parameter generation process to smooth out the trajectory and not necessarily force the parameters to the original LSF vectors at the transition segments.

In some cases, even after smoothing, significant discontinuities still remain at the morphemes. In those cases, synthetic LSF parameters are used instead of the natural parameters. Detection of discontinuity was done using the $L2$ norm of the difference of the LSF vectors at the morpheme boundaries. For each morpheme in the training database, L2 norm of the difference vectors at the morpheme boundaries are measured for all instances. Then, distribution of the L2 norms is modelled with a Gaussian function. The maximum allowable L2 norm at the transition to/from for a morpheme m

$$L2^{m}_{min} = \mu_{L2,m} + 3\sigma_{L2,m} \qquad (32)$$

where $\mu_{L2,m}$ is the mean and $\sigma_{L2,m}$ is the standard deviation of the distribution. Different thresholds are used for the initial and final transition frames. Morphemes are required to satisfy the condition above for both cases.

Between any two root words, more than one morpheme can, and typically do, exist. Therefore, decision for a current morpheme should be considered in context of other decisions. Here, a brute force approach is taken and for all possible combinations of synthetic and natural segments between any two words, on the combination that has the maximum number of natural segments while satisfying the constraint above is decided. Because decisions are fixed for the root words, only local decisions are required which makes this approach feasible.

Energy of the morphemes typically do not match with the energy contours of the synthesized morphemes. To solve the issue, first the energy feature is multiplied with a scaling factor such that the average energy of the selected morpheme is equal to the average energy of the synthetic morpheme. Natural speech units tend to vary more than the synthetic ones. Therefore, even when the average energies are equal, selected morpheme sounds louder because the segments where the speech energy peaks cannot be suppressed sufficiently. Therefore, a second scaling factor is used so that the ratio of peak energies

$$\frac{PE_{syn}}{PE_{hyb}} \geq PE_{max}. \tag{33}$$

4.4.3 Parameter Generation by Statistical Model Modification

Examining MLPG and SBCPG parameter generation formulas and the trajectories generated with them, following observations are made.

- When there are no natural segments constraints in SBCPG, it yields the same result with MLPG.
- They reflect the dynamic, speed and acceleration, constraints in the generated trajectories.
- Distribution statistics can be explicitly modified per frame via the mean and inverse variance matrices.
- They exhibit the variance characteristic of the distribution. Such as, when variance of a distribution in a synthetic frame is explicitly set to a value which is almost zero, ϵ, trajectories are enforced to pass from almost mean of that distribution. Relaxing this variance constraints, gives a change to deviate from the mean.

Using these facts, boundary problem is overcome by relaxing the variance of the boundary frames. Hence, system is able to create smooth transitions. Also in MLMS approach, during the parameter generation of LSF features, mean of middle frame of each state is replaced. Mean of each middle frame is replaced with the frame with highest likelihood value among the corresponding state frames of selected morpheme. To ensure that the generated trajectory passes as close as possible to the imposed frame, variance of the replaced frames are set to ϵ. Since, only the middle frames of each state are constrained, algorithm generates a smooth trajectory while passing very close to mean at constrained frames.

Table 4: Morpheme Counts in the unit selection database.

Total number of morphemes	1,346
Total number of morphemes that have at least two phonemes	1,324
Total number of morphemes that have at least $\nu = 15$ instances and have more than one phoneme	181
Total number of stored LSF vectors	540,493

4.4.4 Global Variance Adjustment

To reduce the smoothness of the SSS-based feature trajectories and increase the variance of the features, a global variance (GV) adjustment algorithm was proposed [55]. The objective function in Eq. 29 is modified with

$$\widehat{c} = \arg\max_{c} \log\{p(Wc|\widehat{Q}, \lambda)^{w} p(v(c)|\lambda_v)\} \tag{34}$$

where $v(c)$ is the variance of the features c throughout the utterance and w adjusts the weights between classical parameter generation and variance adjustment. In implementation, features are generated with the classical approach first and then an iterative algorithm is used to modify the features to increase the variance according to Eq. 34.

After morphemes are selected and the hybrid parameter trajectories are generated by described algorithms, global variance algorithm is used in the proposed system. However, the natural pitch trajectories and the LSF features in the morphemes are not modified during the GV iterations.

CHAPTER V

EXPERIMENTS

5.1 Tests

All systems in the experiments were trained with 30 dimensional vectors consisting of 24 LSFs, 1 log F0 coefficient and 5 voicing strength parameters. Voicing strengths are computed using normalized auto-correlation measure for five evenly spaced spectral bands between 0 and 8 kHz. Recordings were done at 44.1 kHz sampling rate. Speech signal is amplitude-normalized and down sampled to 16 kHz before training. Forced alignment is used to asses the phoneme boundaries. The HTS 2.1 toolkit is used in training and synthesis [1]. Global variance and mixed-excitation are used in addition to post-filtering to improve the speech quality.

2300 utterances were recorded by a female speaker. Total duration of the recorded speech is approximately 190 minutes. The speaker is a professional actress speaking with Istanbul accent. Recording is done in a professional studio environment with an high-quality microphone.

Morpheme database is created using the same training data. A morpheme class is required to have at least ν number of instances in the database where ν is experimentally set to 15. Moreover, morphemes are required to have more than one phoneme. It is found that short morphemes that contain only one phoneme can potentially create discontinuous contours. Thus, those short morphemes are eliminated from the database. Total number of morphemes that were available in the training data is shown in Table 4. The database size roughly 50MB without any compression. The size can be reduced substantially using parametric speech compression techniques.

[1] http://hts.sp.nitech.ac.jp/

Table 5: Parameters of the MLMS and MSVA algorithms

R_d (MLMS)	30
F (MSVA)	2
w (MSVA)	[1 0.5 0.3]
PE_{max} (MSVA)	0.8
B_s (MSVA)	7
ζ_d (MSVA)	1.5

Table 6: MOS Test Results of the baseline system

Mean MOS Score	3.27
Median MOS Score	3
Variance of the MOS Score	1.02

However, such compression techniques are not investigated here since it is out of the scope of this work. Parameters of the MLMS and MSVA algorithms are given in Table 5.

Experiments are performed in three phases. In the first phase, performance of the baseline system is assessed. In the second phase, the MLMS approach is tested and compared with the baseline system. In the third phase, MSVA approach is tested and compared with the baseline system. The effects of hybrid pitch and LSF features are analyzed separately in the third phase of testing.

5.1.1 Baseline System Performance

After the baseline system is built, several issues have been noted. The first issue is the discontinuities during vowel transitions in diphthongs and glide-vowel transitions. The second issue is the annoying clicking sounds that randomly pop up in the middle of some of the samples. It is observed that errors in the alignment process is mostly responsible for those issues. For example, the /m/ sound occurs very frequently at the end of Turkish sentences and some of the issues with the /m/ sounds were found to occur because silence is erroneously labeled as part of the some of the /m/ sounds at the end of sentences by the aligner. Those issues are fixed by manually correcting the segmentation of problematic sounds in the training data.

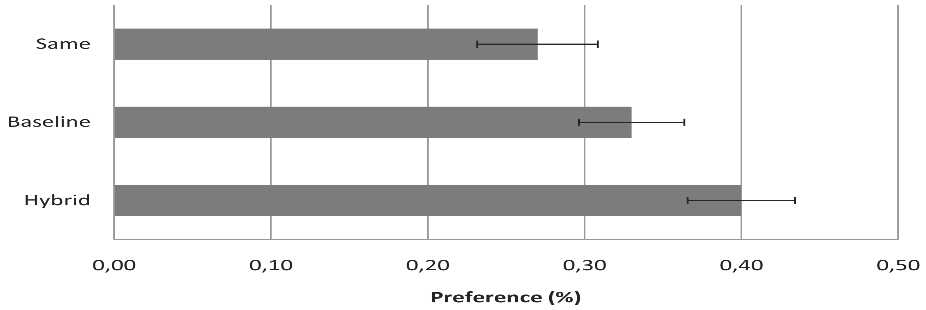

Figure 6: AB preference test results for the hybrid MLMS algorithm and the baseline systems.

Mean Opinion Score (MOS) test is used to test the quality of the baseline system. 8 male and 8 female listeners took the listening tests. All of the listeners were native speakers of Turkish. In the MOS test, subjects were presented 2 sample voices for each MOS score case for calibration purposes. Listeners were then presented an utterance and asked to give it a score which represents how natural the sentence sounded. 12 test sentences were selected from news domain and 18 sentences were selected from novel domain. Results are shown in Table 6. The MOS performance of the baseline Turkish system is similar to the scores obtained for English [56].

5.1.2 Performance of the MLMS Algorithm

In order to assess the quality improvement with the hybrid MLMS approach, AB preference test is performed. There was no significant listener preference for the MLMS approach compared to the baseline case. Analyzing the results, the underlying reason was found to be the smooth trajectories that were selected with the maximum likelihood approach as discussed in Section 4.3.2. To alleviate the effect, another voice is created from a different female speaker with more variation in speech. The idea is to train HMM models using a a more dramatic/poetic speech database that puts higher weight on pitch and LSF variations in parameter generation. All system parameters are same in this test with the baseline system. 70 minutes of training data is used with the new speaker.

Using the new voice with the MLMS algorithm, AB test results are shown in Fig. 6 with confidence intervals. Results are found to be significant according to Pearson's chi-squared test. Thus, there is a preference for the hybrid system over the baseline system. However, the difference is small.

In Turkish, question sentences typically have special morphemes, such as /mi/, /midir/, at the end of the verbs. In some significant number of cases with the baseline system, it is noticed that over-smoothed question tags which significantly hurt the listener preference. Most of those issues are resolved with the hybrid system since stress patterns of the question sentences are captured better by the hybrid system. An example case is shown in Fig. 7 where the hybrid system better modeled the pitch rise at the end of a question utterance.

Another interesting syntactic morpheme in Turkish is /de/, /da/ which means "also" in English. They are written as if they are independent words while they are treated as a morpheme of the word that they come after in this work. Those tags are very commonly used in Turkish and using correct prosody for them is important to convey the correct semantic message. The hybrid system generated more natural prosody for those morphemes since their intonation patterns are selected from the natural units in the morpheme database.

The hybrid system improved the intonation contours which, analyzing the listener feedback, made the most difference in the improved perceptual quality. Another example to pitch contour improvement with the hybrid system is shown in Fig. 8.

In synthesis, severe and frequent discontinuities were observed for the LSF features since the concatenation cost was not taken into account. To minimize the discontinuities, the smoothing idea described in Section 4.3.3 was used for all frames. However, in this case, clarity in the LSF features was lost significantly and listeners could not hear the difference between the hybrid LSF features and the baseline LSF features. Therefore, significant improvement was not obtained for the LSF features

Table 7: Variance of the logarithm of pitch for the baseline and hybrid systems.

Baseline System (MSVA)	0.035
Hybrid System (MSVA)	0.042
Baseline System (MLMS)	0.038
Hybrid System (MLMS)	0.039

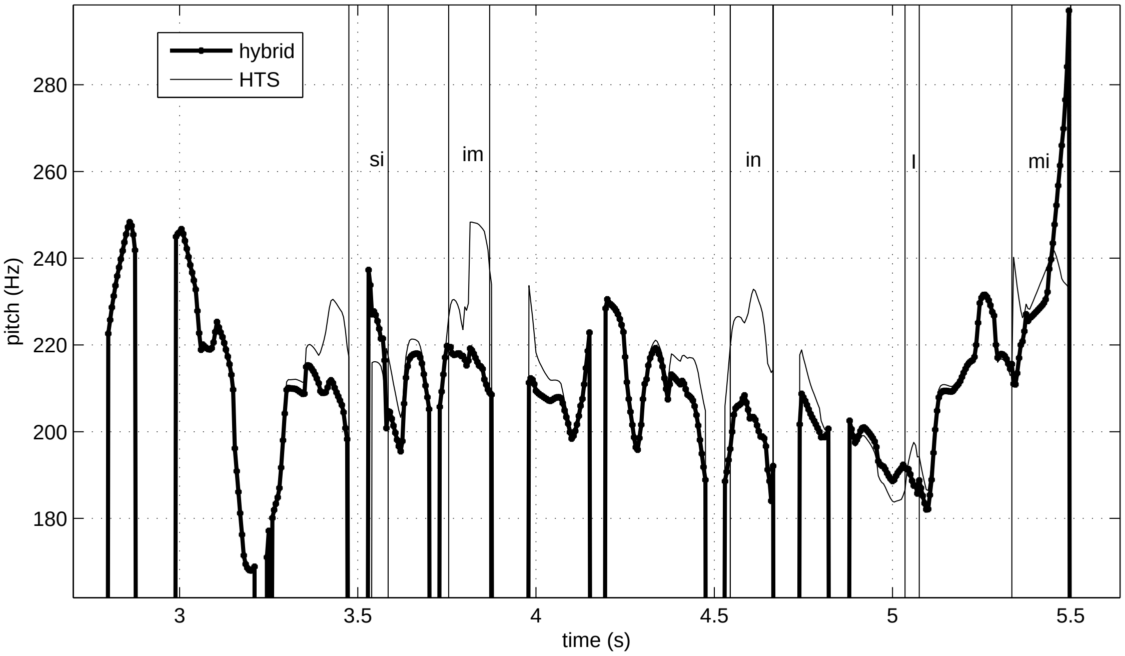

Figure 7: Comparison of pitch trajectories for the baseline and hybrid systems. Borders of the five morphemes occurring in the utterances are shown. The final morpheme /mi/ indicates a question. Sudden pitch rise that is expected at the end of the question utterance is better modelled with the MLMS-based hybrid system.

in the MLMS approach.

5.1.3 Performance of the MSVA Algorithm

Similar to the MLMS algorithm, AB preference tests are performed to assess the performance of the MSVA algorithm. Tests are conducted in two phases. In the first phase, hybrid pitch features are used with the baseline LSF features. In the second phase, both pitch and LSF features are generated with the hybrid algorithm to assess the additional improvement with the LSF features. 30 sentences are used and 10 listeners took the tests.

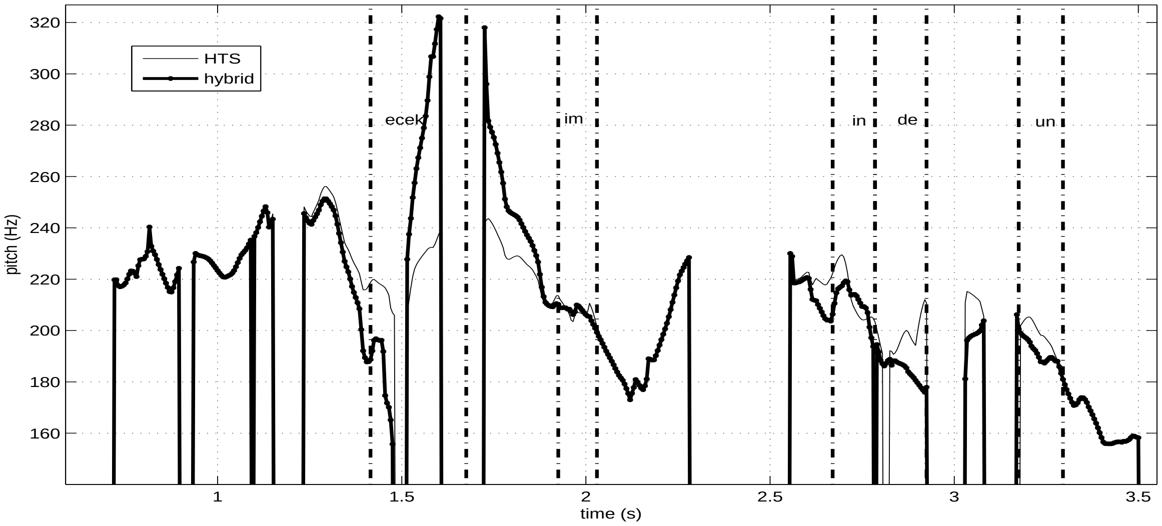

Figure 8: Comparison of pitch contours for the baseline and hybrid MLMS systems. Borders of the five morphemes occurring in the utterances are shown. Sudden pitch variation on the morpheme is modeled better with the MLMS-based hybrid system. Synthetic speech with the hybrid pitch contour was perceived as more natural by the listeners.

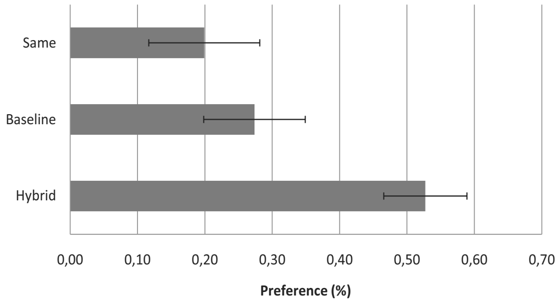

Figure 9: AB preference test results for the hybrid MSVA algorithm where only the pitch feature is synthesized with the hybrid method and LSF parameters are same in the baseline and hybrid systems.

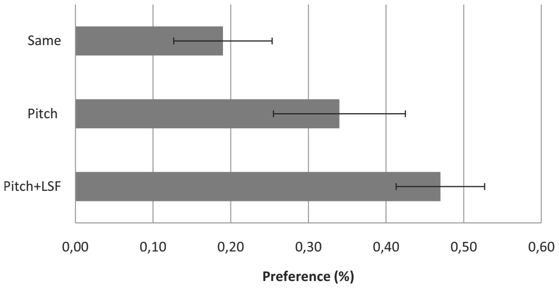

Figure 10: AB preference test results for the hybrid MSVA algorithm and the baseline systems when both LSF and pitch are generated with the hybrid method.

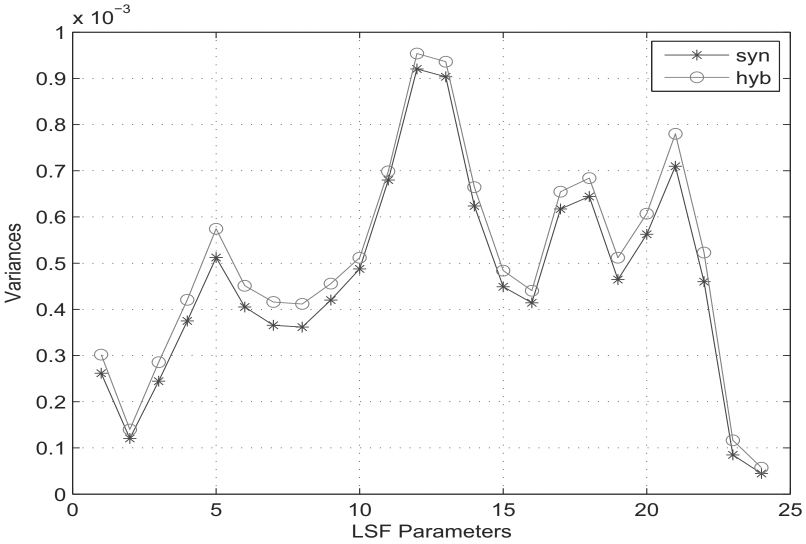

Figure 11: Comparison of variances for the LSF parameters generated with the hybrid MSVA method and baseline systems. Variance is higher for the hybrid method for all 24 LSF parameters.

Results of the AB test with hybrid pitch features are shown in Fig. 9. The hybrid system significantly outperformed the baseline SSS system in these tests. It was found that improvement in the musicality of speech that are related to the variance of the pitch contours throughout utterances were clearly noticed by the listeners. Fig. 12 shows how the pitch contour changes for an utterance with the hybrid approach compared to the baseline approach during the morpheme segments. Rise in pitch in stressed morphemes are seen in Fig. 12 especially for the /ler/ and /dan/ morphemes. Note that time-warping is required when only the pitch feature is generated with the hybrid method since the LSF features are still obtained from the baseline system. However, the MSVA algorithm has constraints on the duration of the selected morphemes, as described in Section 4.3.3, which reduces the artifacts due to time-warping.

To assess the overall improvement in pitch variability, variance of the lf0 feature is computed for the 30 test utterances. Average of the variances is then computed and compared with the baseline system. Results are shown in Table 7. The perceptual

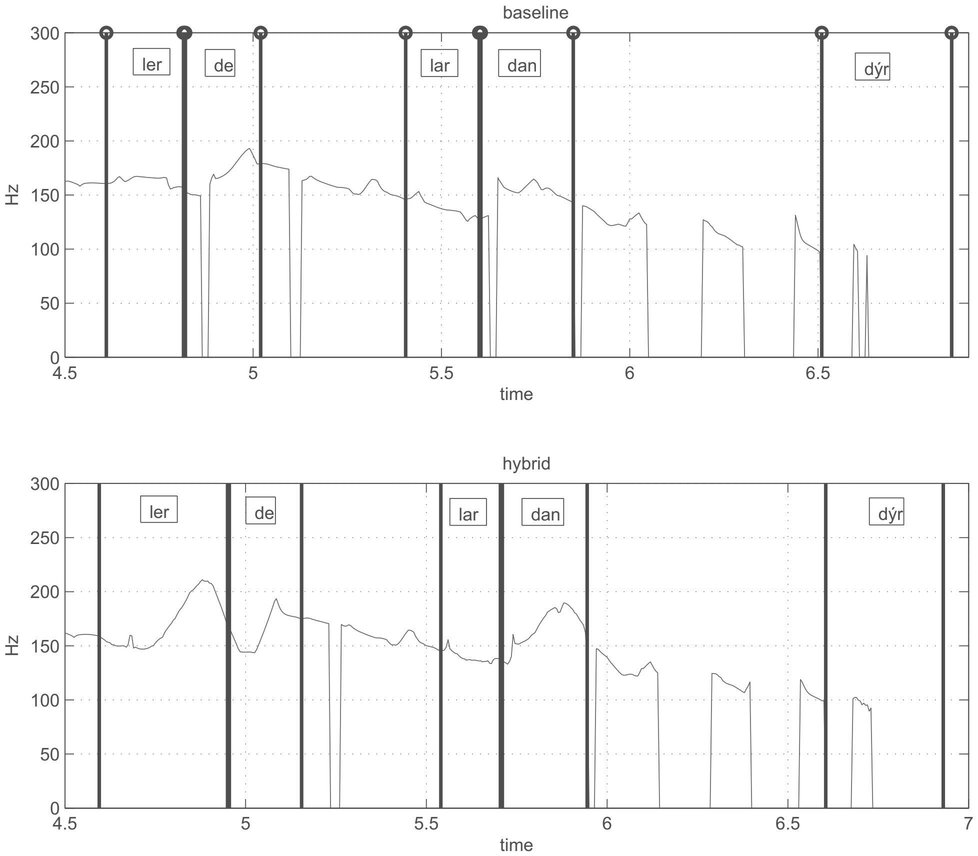

Figure 12: Comparison of pitch contours for the hybrid MSVA method and baseline systems. Morphemes and their boundaries are indicated in the figure.

improvement in pitch variability is confirmed objectively in Table 7. Such improvement in variability was not observed for the MLMS case primarily because of the smoothing effect of the likelihood based selection algorithm.

The samples where the listeners preferred the baseline system compared to the hybrid system are analyzed. Almost all of those samples corresponded to cases where the pitch contour from the selected morpheme is stressed and longer than the synthetic morpheme duration. In that case, pitch contour is time-warped before parameter generation which caused glitches in some cases. This problem has been resolved when both LSF and pitch contours are used from the natural morpheme since time-warping is not done in that case as discussed in Section 4.3.3.

Results of AB tests for comparing the MSVA-based hybrid pitch contours and

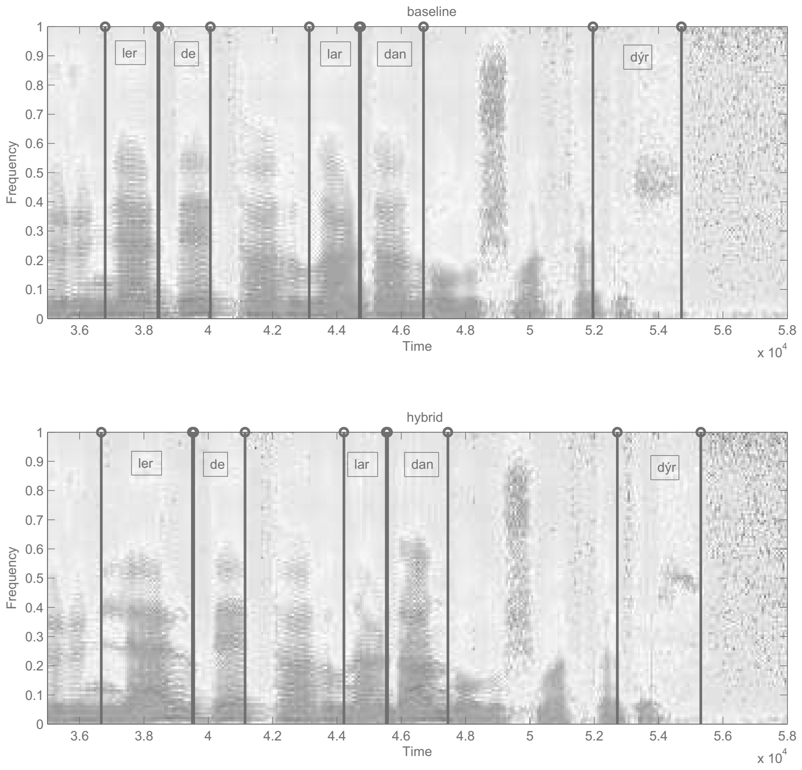

Figure 13: Comparison of spectrograms for the hybrid MSVA method and baseline systems. Morphemes and their boundaries are indicated in the figure.

MSVA-based hybrid pitch and LSF contours are shown in Fig. 10. Using hybrid LSF contours helps for improving the quality. However, discontinuities in speech, when it occurs, hurts the listeners' preference. Some algorithms are proposed to reduce the discontinuities and severe cases of discontinuities do not occur. However, while some of the listeners perceived those as natural speech variability and preferred it, some others perceived them as artifacts. That increased the confidence intervals in the preference tests as can be in Fig. 10. The pitch glitches that were observed in the pitch-only case above was not observed in the hybrid pitch and LSF case which also increased the speaker preference for the hybrid pitch and LSF case. On average eight speakers had higher preference for the hybrid LSF and pitch system and two speakers had preference for the pitch-only system.

Similar to the pitch feature, variance of the LSF features increased with the hybrid approach. Variance of the LSF features are compared with the baseline system in Fig. 11. Not only the LSF variance but also a related parameter, formant variance, improves with the hybrid approach as expected. An example is shown in Fig. 13 where the spectrograms of the baseline system and hybrid system are compared. Improvement in the formant trajectories can be observed in the morpheme /ler/ for example. Not only the formant trajectories but also the formant bandwidths are improved during natural morphemes even though smoothing was applied at the morpheme boundaries. Those improvements were typically frequent enough throughout utterances and most listeners did not complain about fluctuations in speech quality but perceived them as natural variations in speech.

CHAPTER VI

CONCLUSION

A hybrid statistical/unit selection speech synthesis system which can be used to significantly improve the quality of the HTTS systems for morphologically rich languages is proposed. As opposed to common hybrid techniques, the proposed technique does not significantly hurt the small memory footprint advantage of the HTTS systems which makes it attractive for embedded applications. The proposed idea is tested with Turkish which is a morphologically rich language.

Morphemes are the fundamental units of the UTTS system in the hybrid approach. Two morpheme selection algorithms are proposed. The MLMS approach is based on maximum-likelihood based target cost calculation. Experiments showed that MLMS generates overly smooth trajectories in many cases. Learning from the experience with the MLMS approach, another morpheme selection technique, MSVA algorithm, is proposed which is based on Viterbi-based concatenation of the pitch contours. The MSVA algorithm not only modelled the pitch variations better than the MLMS algorithm, but also the LSF trajectories of the morphemes selected with the MSVA approach fit better in the morpheme contexts and required less smoothing which helped improve the quality. The improvements in pitch and LSF contours are verified by subjective listening tests and objective measurement of the parameter variances.

APPENDIX A

SCRIPTS OF MORPHEME BASED HYBRID SYSTEM

- OZULibrary
  - **OZU_CreateNLPFiles.py** : Program creates nlp files for each sentence using XFST morphological analyzer. Nlp files are morpheme based representations of the sentences.
  - **OZU_CreateMorphFolder.py** : Program creates morph files for each sentence using nlp files, HTS state level alignments. Morph file is a morpheme based label file of a sentence including state level alignment and context informations.
  - **OZU_CreateMorphDatabase.m** : Program creates morpheme database from the extracted morph files of each train sentence.
  - **OZU_SolveMorphViterbi.m** : Program first calculates the costs of the each viterbi for each morph file of the test sentence. If flag is set, then it solves the viterbi. It gives the ID of the selected morphemes in a text file for each sentence.
  - **OZU_HybridParamGenerate_Viterbi.m** : Program generates the hybrid speech parameters for each sentence using its viterbi solution.
  - **OZU_ParamGen_Vocode.py** : This is the wrapper program to extract speech parameters from HTS model and also vocode speech from given parameters using modified HTS engine. Hybrid speech is generated using this wrapper.

Bibliography

[1] P. Taylor, *Text-to-Speech Synthesis.* Cambridge University Press, 2009.

[2] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis," in *Eurospeech*, pp. 2347–2350, 1999.

[3] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. Black, and K. Tokuda, "The HMM-based speech synthesis system (HTS) version 2.0," in *ISCA SSW6*, pp. 294–299, 2007.

[4] A. Ljolje and F. Fallside, "Synthesis of natural sounding pitch contous in isolated utterances using hidden Markov models," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. vol. ASSP-34, pp. 1074–1080, 1986.

[5] M. Giustiniani and P. Pierucci, "Phonetic ergodic HMM for speech synthesis," in *EUROSPEECH*, pp. 349–352, 1991.

[6] T. Fukada, Y. Komori, T. Aso, and Y. Ohora, "A study of pitch pattern generation using HMM-based statistical information," in *ICSLP*, pp. 723–726, 1994.

[7] R. E. Donovan and P. C. Woodland, "Automatic speech synthesiser parameter estimation using HMMs," in *ICASSP*, pp. 640–643, 1995.

[8] K. Tokuda, T. Masuko, T. Yamada, T. Kobayashi, and S. Imai, "An algorithm for speech parameter generation from continuous mixture hmms with dynamic features," in *EUROSPEECH*, pp. 757–760, 1995.

[9] J. Odell, *The Use of Context in Large Vocabulary Speech Recognition.* Ph.D. Thesis, University of Cambridge, 1995.

[10] K. Tokuda, T. Kobayashi, and S. Imai, "Speech parameter generation from hmm using dynamic features," in *ICASSP*, pp. 660–663, 1995.

[11] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Mixed excitation for HMM-based speech synthesis," in *Seventh European Conference on Speech Communication and Technology*, 2001.

[12] O. Abdel-Hamid, S. Abdou, and M. Rashwan, "Improving Arabic HMM based speech synthesis quality," in *ISCA SSW5*, pp. 1332–1335, 2006.

[13] H. Kawahara, I. Masuda-Katsuse, and A. Cheveigne, "speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based f0 extraction: possible role of a repetitive structure in sounds," *Speech Comm.*, vol. 27 (3), pp. 187–207, 1999.

[14] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, "Multispace probability distribution HMM," *IEICE Trans. Inform. Systems*, vol. E85-D (3), pp. 455–464, 2002.

[15] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Duration modeling for hmm-based speech synthesis," in *ICSLP*, pp. 29–32, 1998.

[16] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for hmm-based speech synthesis," in *ICASSP*, vol. 3, pp. 1315–1318, IEEE, 2000.

[17] K. Koishida, K. Tokuda, T. Masuko, and T. Kobayashi, "Vector quantization of speech spectral parameters using statistics of dynamic features," in *ICSP*, pp. 247–252, 1995.

[18] S. Imai, K. Sumita, and C. Furuichi, "Mel log spectrum approximation (mlsa) filter for speech synthesis," *Electron. Comm. Jpn.*, vol. 66 (2), pp. 10–18, 1983.

[19] A. J. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *ICSLP*, pp. 373–376, 1996.

[20] T. Hirai and S. Tenpaku, "Using 5 ms segments in concatenative speech synthesis," in *ISCA SSW5*, 2004.

[21] R. Donovan and P. Woodland, "Improvements in an hmm-based speech synthesiser," in *Eurospeech*, pp. 573–576, 1995.

[22] B. Beutnagel, A. Conkie, J. Schroeter, Y. Stylianou, and A. Syrdal, "The AT&T Next-Gen TTS system," in *Joint ASA, EAA and DAEA Meeting*, pp. 15–19, 1995.

[23] A. Black and P. Taylor, "Automatically clustering similar units for unit selection in speech synthesis," in *Eurospeech,*, pp. 601–604, 1997.

[24] H. Segi, T. Takagi, and T. Ito, "A concatenative speech synthesis method using context dependent phoneme sequences with variable length as search units," in *ISCA SSW5*, pp. 115–120, 2004.

[25] H. Zen, K. Tokuda, and A. W. Black, "Review: Statistical parametric speech synthesis," *Speech Commun.*, vol. 51, pp. 1039–1064, Nov. 2009.

[26] Y. Stylianou, "Assessment and correction of voice quality variabilities in large speech databases for concatenative speech synthesis," in *ICASSP*, pp. 377–380, 1999.

[27] J. Gauvain and C. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Trans. Speech Audio Process*, vol. 2, pp. 291–298, 1994.

[28] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Comput. Speech Language*, vol. 9, pp. 171–185, 1995.

[29] J. Yamagishi, *Average-Voice-Based Speech Synthesis.* Ph.D. Thesis, Tokyo Institute of Technology, 2006.

[30] H. Kawai, T. Toda, J. Ni, M. Tsuzaki, and K. Tokuda, "XIMERA: A new TTS from ATR based on corpus-based technologies," in *ISCA SSW5*, 2004.

[31] S. Krstulovic, J. Latorre, and S. Buchholz, "Comparing QMT1 and HMMs for the synthesis of American English prosody," in *Speech Prosody*, p. 6770, 2008.

[32] H. Hon, A. Acero, X. Huang, J. Liu, and M. Plumpe, "Automatic generation of synthesis units for trainable text-to-speech systems," in *ICASSP*, pp. 293–296, 1998.

[33] T. Okubo, R. Mochizuki, and T. Kobayashi, "Hybrid voice conversion of unit selection and generation using prosody dependent HMM," *IEICE TRANSACTIONS on Information and Systems*, vol. 89, no. 11, p. 2775, 2006.

[34] Z. Ling and R. Wang, "HMM-based hierarchical unit selection combining KullbackLeibler divergence with likelihood criterion," in *ICASSP*, pp. 1245–1248, 2007.

[35] S. Rouibia and O. Rosec, "Unit selection for speech synthesis based on a new acoustic target cost," in *INTERSPEECH*, pp. 2565–2568, 2005.

[36] C. Yang, Z. Ling, Y. Jiang, L. Dai, Y. Hu, L. Chen, and R. Wang, "The USTC System for Blizzard Challenge 2011," in *Blizzard Challenge Workshop*, 2011.

[37] S. Pan, M. Zhang, and J. Tao, "A novel hybrid approach for mandarin speech synthesis," in *11th Annual Conference of the ISCA*, pp. 182–185, 2010.

[38] Y. Qian, Z. Yan, Y. Wu, F. Soong, X. Zhuang, and S. Kong, "An hmm trajectory tiling (htt) approach to high quality tts," in *INTERSPEECH*, p. 422*425, 2010.

[39] Z. Ling and R. Wang, "Minimum unit selection error training for HMM-based unit selection speech synthesis system," in *ICASSP*, pp. 3949–3952, 2008.

[40] R. Blum, "Multidimensional stochastic approximation method," *Ann. Mat. Stat.*, vol. 25, pp. 737–744, 1954.

[41] M. Plumpe, A. Acero, H. Hon, and X. Huang, "HMM-based smoothing for concatenative speech synthesis," in *ICSLP*, p. 27512754, 1998.

[42] J. Wouters and M. Macon, "Unit fusion for concatenative speech synthesis," in *ICSLP*, pp. 302–305, IEEE, 2000.

[43] T. Raitio, A. Suni, H. Pulakka, M. Vainio, and P. Alku, "Utilizing glottal source pulse library for generating improved excitation signal for hmm-based speech synthesis," in *ICASSP*, pp. 4564–4567, 2011.

[44] X. Gonzalvo, A. Gutkin, J. Socoro, I. Iriondo, and P. Taylor, "Local minimum generation error criterion for hybrid HMM speech synthesis," in *INTERSPEECH*, pp. 416–419, 2009.

[45] H. Kawahara, J. Estill, and O. Fujimura, "Aperiodicity extraction and control using mixed mode excitation and group delay manipulation for a high quality speech analysis, modification and synthesis system straight," in *MAVEBA*, 2001.

[46] Z. Ling, L. Qin, H. Lu, Y. Gao, L. Dai, R.Wang, Y. Jiang, Z. Zhao, J. Yang, J. Chen, and G. Hu, "The USTC and iFlytek speech synthesis systems for Blizzard Challenge 2007," in *The Blizzard Challenge 2007 Workshop*, 2007.

[47] Y. Wu and R. Wang, "Minimum generation error training for hmm-based speech synthesis," in *ICASSP*, pp. 89–92, 2006.

[48] M. Aylett and J. Yamagishi, "Combining statistical parametric speech synthesis and unit-selection for automatic voice cloning," in *LangTech*, 2008.

[49] V. Pollet and A. Breen, "Synthesis by generation and concatenation of multiform segments," in *INTERSPEECH*, pp. 1825–1828, 2008.

[50] P. Taylor, "Unifying unit selection and hidden Markov model speech synthesis," in *Ninth International Conference on Spoken Language Processing*, pp. 1758–1761, 2006.

[51] I. Ergenc, *Spoken Language And Dictionary Of Turkish Articulation.* MULTILINGUAL, 2002.

[52] K. Oflazer and S. Inkelas, "A finite state pronunciation lexicon for Turkish," in *Proceedings of the EACL Workshop on Finite State Methods in NLP, Budapest, Hungary*, vol. 82, pp. 900–918, 2003.

[53] D. Yuret and F. Ture, "Learning morphological disambiguation rules for Turkish," in *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pp. 328–334, Association for Computational Linguistics, 2006.

[54] S. Tiomkin, D. Malah, S. Shechtman, and Z. Kons, "A Hybrid Text-to-Speech System That Combines Concatenative and Statistical Synthesis Units," *IEEE Transactions on Audio Speech and Language Processing*, vol. 19, pp. 1278–1288, JUL 2011.

[55] T. Toda and K. Tokuda, "A speech parameter generation algorithm considering global variance for hmm-based speech synthesis," *IEICE - Trans. Inf. Syst.*, vol. E90-D, pp. 816–824, May 2007.

[56] S. Kinga and V. Karaiskosb, "The Blizzard challenge 2009," in *The Blizzard Challenge 2009 Workshop*, 2009.

VITA

Ekrem Güner graduated from Control Engineering program of Istanbul Technical University (İTÜ) in 2008. He took remedial courses from Computer Engineering department for one year. In 2009, he started Msc. in Computer Engineering in İTÜ. After one year, he started to work in a project at Özyeğin University and he left İTÜ and started Msc. in Özyeğin University in 2010. During two years in İTÜ, He studied computer vision and video processing in a Mechatronics Laboratory. In the laboratory, he developed and implemented algorithms for lane detection and car tracking. He started to work on Text-to-Speech (TTS) topic in Özyeğin University. In 2011 with his advisor, he co-founded NEOSES, a company focused on developing high-tech speech processing solutions. He has been working on specialized embedded solutions for TTS applications. His main research interests are video processing, speech processing and pattern recognition.