Turk J Elec Engin, VOL.14, NO.2 2006, © TÜBİTAK

# A Corpus-Based Concatenative Speech Synthesis System for Turkish

**Haşim SAK**[1], **Tunga GÜNGÖR**[1], **Yaşar SAFKAN**[2]
[1] *Boğaziçi University, Department of Computer Engineering,*
*34342, Bebek, İstanbul-TURKEY*
*e-mail: hasim_sak@yahoo.com, gungort@boun.edu.tr*
[2] *Yeditepe University, Department of Computer Engineering,*
*81120, Kayışdağı, İstanbul-TURKEY*
*e-mail: ysafkan@cse.yeditepe.edu.tr*

**Abstract**

*Speech synthesis is the process of converting written text into machine-generated synthetic speech. Concatenative speech synthesis systems form utterances by concatenating pre-recorded speech units. Corpus-based methods use a large inventory to select the units to be concatenated. In this paper, we design and develop an intelligible and natural sounding corpus-based concatenative speech synthesis system for the Turkish language. The implemented system contains a front-end comprised of text analysis, phonetic analysis, and optional use of transplanted prosody. The unit selection algorithm is based on commonly used Viterbi decoding algorithm of the best-path in the network of the speech units using spectral discontinuity and prosodic mismatch objective cost measures. The back-end is the speech waveform generation based on the harmonic coding of speech and overlap-and-add mechanism. Harmonic coding enabled us to compress the unit inventory size by a factor of three. In this study, a Turkish phoneme set has been designed and a pronunciation lexicon for root words has been constructed. The importance of prosody in unit selection has been investigated by using transplanted prosody. A Turkish Diagnostic Rhyme Test (DRT) word list that can be used to evaluate the intelligibility of Turkish Text-to-Speech (TTS) systems has been compiled. Several experiments have been performed to evaluate the quality of the synthesized speech and we obtained 4.2 Mean Opinion Score (MOS) in the listening tests for our system, which is the first unit selection based system published for Turkish.*

## 1. Introduction

Speech synthesis is the process of converting written text into machine-generated synthetic speech. In the literature, there are three main approaches to speech synthesis: articulatory, formant, and concatenative [1-4]. Articulatory synthesis tries to model the human articulatory system, i.e. the vocal cords, the vocal tract, etc. Formant synthesis employs some set of rules to synthesize speech using the formants that are the resonance frequencies of the vocal tract. Since the formants constitute the main frequencies that make sounds distinct, speech is synthesized using these estimated frequencies. On the other hand, concatenative speech synthesis is based on the idea of concatenating pre-recorded speech units to construct the utterance. Concatenative systems tend to be more natural than the other two since original speech recordings are used

instead of models and parameters. In concatenative systems, speech units can be either fixed-size diphones or variable length units such as syllables and phones. The latter approach is known as unit selection, since a large speech corpus containing more than one instance of a unit is recorded and variable length units are selected based on some estimated objective measure to optimize the synthetic speech quality.

Corpus-based concatenative speech synthesis (unit selection) has emerged as a promising methodology to solve the problems with the fixed-size unit inventory synthesis, e.g. diphone synthesis [3-7]. In corpus-based systems, the acoustic units of varying sizes are selected from a large speech corpus and concatenated. The speech corpus contains more than one instance of each unit to capture prosodic and spectral variability found in natural speech; hence the signal modifications needed on the selected units are minimized if an appropriate unit is found in the unit inventory. The use of more than one instance of each unit requires a unit selection algorithm to choose the units from the inventory that match best the target specification of the input sequence of units. The unit selection algorithm favors choosing consecutive speech segments in order to minimize the number of joins.

The output speech quality of unit selection in terms of naturalness is much better than fixed-size unit inventory synthesis. However, the unit selection presents some challenges to speech synthesis. Although the speech quality is acceptable most of the time, it is not consistent. If the unit selection algorithm fails to find a good match for a target unit, the selected unit is needed to undergo some prosodic modifications which degrade the speech quality at this segment join. Some systems even choose not to do any signal modifications on the selected units [8]. To ensure a consistent quality, a good speech corpus design that covers all the prosodic and acoustic variations of the units that can be found in an utterance has to be addressed. It is not feasible to record larger and larger databases given the complexity and combinatorics of the language; instead we need to find a way for optimal coverage of the language [9]. Another point is that concatenating the speech waveforms results in some glitches at the concatenation points in the synthesized utterance. Therefore, to ensure smooth concatenation of speech waveforms and to enable prosodic modifications on the speech units, a speech model is generally used for speech representation and waveform generation [10-12].

ATR v-Talk speech synthesis system developed at ATR laboratories introduced the unit selection approach from a large speech database [3]. The selection of units was based on minimizing an acoustic distance measure between the selected units and target spectrum. In CHATR speech synthesis system, prosodic features like duration and intonation have been added to the target specification to choose more appropriate units [4]. Hunt and Black have contributed to the area the idea of applying Viterbi decoding of best-path algorithm for unit selection [13]. The Next-Gen speech synthesis system developed at the AT& T laboratories is one of the commercial systems that use unit selection [5]. The front-end, i.e. the text and linguistic analysis and prosody generation is from Flextalk, the unit selection is a modified version of CHATR, and the framework for all these was borrowed from the Festival. As an improvement to the CHATR unit selection, the system uses half phones compared to phonemes as the basic speech units [14]. This allows phoneme or diphone concatenation at a unit boundary. For the back-end, a Harmonic plus Noise Model (HNM) representation of the speech has been developed [11]. Unit selection based concatenative speech synthesis approach has also been used in the IBM Trainable Speech Synthesis System [7]. The system uses the Hidden Markov Models (HMMs) to phonetically label the recorded speech corpus and aligns HMM states to the data. The units used in the unit selection process are HMM state sized speech segments. The unit selection is a dynamic programming based search, which uses decision trees to facilitate the choice of appropriate units, with a cost function to optimize. The segments in the speech database are coded into Mel-Frequency Cepstrum Coefficients (MFCCs).

In this paper, we propose an intelligible and natural sounding corpus-based speech synthesis system for Turkish. The system consists of an analysis component which converts the text into a linguistic and prosodic description, a unit selection component based on Viterbi decoding, and a waveform generation component based on the harmonic coding of speech and the overlap-and-add mechanism. The research in this paper is directed towards agglutinative languages in general and Turkish in particular. Speech synthesis systems are currently being developed for languages like English and successful results are obtained. However, the studies on Turkish which is an agglutinative language and has a highly complex morphological structure are quite limited. In this study, we take the special characteristics of Turkish into account, propose solutions for them, and develop a speech synthesis system for the language. To the best of our knowledge, this is the first unit selection based system published for Turkish.

The paper is organized as follows: Section 2 presents the overall architecture of the proposed system and gives the details of the text and speech corpora. Section 3 explains the analysis component comprised of text analysis, phonetic analysis, and transplanted prosody. In Sections 4 and 5, we explain the methodologies and the algorithms used for unit selection and waveform generation, respectively. Section 6 covers the details and the results of the experiments. The last section is for the conclusions.

## 2. System Architecture

The architecture of the system is shown in Figure 1. The components shown are common in most of the speech synthesis systems that use unit selection. The system can be mainly divided into three parts: analysis (front-end), unit selection, and generation (back-end). The analysis module is responsible for producing an internal linguistic and prosodic description of the input text. This description is fed into the unit selection module as the target specification. The unit selection module uses this specification to choose the units from the speech database such that a cost function between the specification and the chosen units is minimized. The waveforms for the selected units are then concatenated in the generation module, where the smoothing of concatenation points is also handled.

The system uses an internal data structure to store the information for the text to be synthesized. This structure is communicated between components and each component appends extracted information using the already existing information in the structure. This enables each system component to be developed independently and makes it flexible to improve the functionalities of each component if required.

### 2.1. Text corpus

The fragments that form the text corpus have been collected from online Turkish text materials. These text fragments have been preprocessed and divided into phrases by making use of the punctuation marks. They have been checked manually and only the phrases that were complete and well-formed have been included while the rest have been discarded. Then a Greedy algorithm has been employed which aims to choose the phrases according to their phonetic context. The algorithm assigns a score to each phrase, calculated as the total frequency of the triphone contexts found in the phrase normalized by the number of the triphones. Then the phrase having the greatest score is selected. The algorithm updates the frequencies of the triphones in the selected phrase to zero and runs on the remaining phrases. The algorithm produced 30000 phrases. In this way, each recording script was formed of phrases (word groups) rather than full sentences in order to prevent repetition of common words, which increases the size of the database but adds little to the overall synthesis quality.

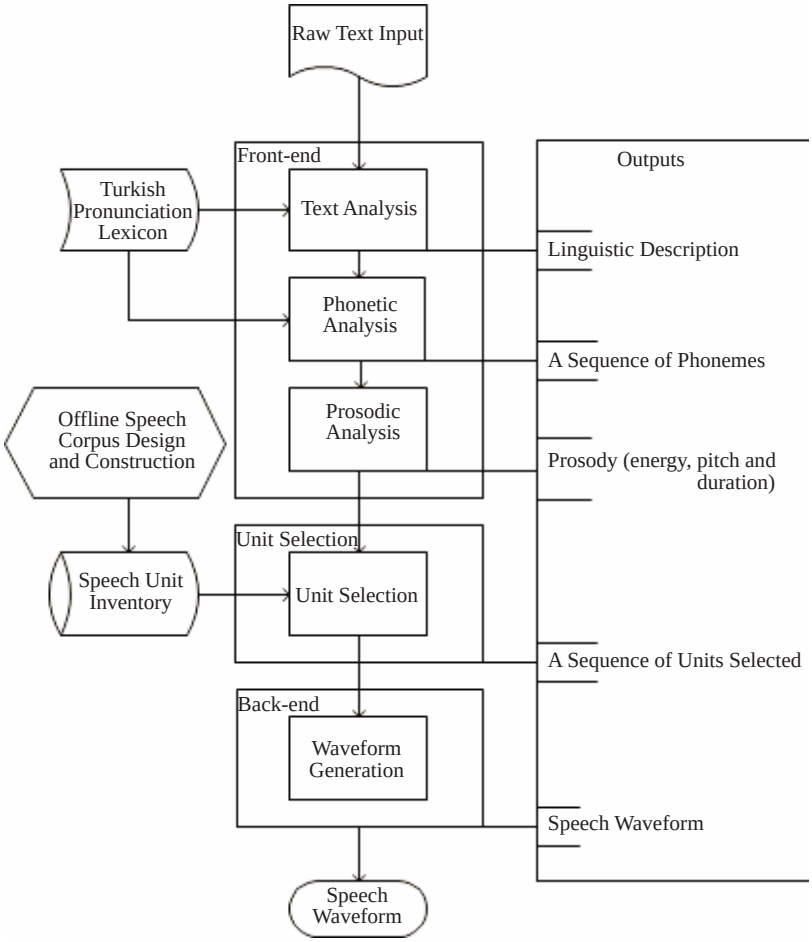


**Figure 1.** Corpus-based concatenative Turkish speech synthesis system architecture.

In order to observe the effect of the corpus size on the output quality and on the performance of the algorithms, we have also constructed a smaller corpus of 5000 phrases. The phrases in this corpus were selected among the 30000 phrases in the original corpus (after the original corpus has been divided into training and test sets – see below). For this purpose, a Greedy algorithm similar to the one used in forming the original corpus was used. The algorithm tries to identify and thus choose the phrases containing syllables that have not yet been covered in the selected phrases. In Turkish, syllables have a prosodic integrity in themselves. We can categorize syllables in Turkish as having the patterns $v$, $vc$, $vcc$, $cv$, $cvc$, and $cvcc$, where $c$ designates a consonant phoneme and $v$ a vowel phoneme. We have also considered syllable boundaries, sentence start and end points, and word boundaries. In this way, we have identified the subset of the corpus that covers all variations of these patterns.

## 2.2. Speech corpus

The speech corpus used by the algorithms developed in this research contains about 20 hours of speech recorded by a professional female speaker covering the 30000 Turkish phrases in the text corpus. The speech corpus has been phonetically aligned by a speech recognition engine and then the phone boundaries have been corrected manually. The corpus has been divided into two sets: training set and test set. The test set contains 1000 phrases used for the purpose of evaluating the synthesis quality. From the remaining

recordings (training set), two speech unit inventories of different sizes have been constructed. One contains all the recordings in the training set (about 19 hours of speech) and the other contains 5000 phrases (about 3 hours of speech) extracted as explained above.

The content of the speech corpus has a major effect on speech quality. During unit selection process, finding units that match best the target specification is more probable if sufficient prosodic and acoustic variability for the phones can be represented in the corpus. The speech quality is severely degraded when appropriate units cannot be found and significant prosodic modifications are performed.

# 3. Forming Linguistic and Prosodic Description

In a language, phonemes are the smallest units of sound that distinguish one word from another [1]. Turkish alphabet contains 29 letters classified as 8 vowels (a, e, ı, i, o, ö, u, ü) and 21 consonants (b, c, ç, d, f, g, ğ, h, j, k, l, m, n, p, r, s, ş, t, v, y, z). However, Turkish orthography cannot represent all the sounds in Turkish. In this study, for phonetic transcriptions we developed a new phoneme set based on the SAMPA phonetic alphabet [15]. The SAMPA identifies 8 vowels and 24 consonants (excluding two consonantal allophones /w/ of /v/ and /N/ of /n/) for representing Turkish sounds and designates a length mark /:/ to represent the lengthening of some vowels in loanwords in Turkish. The new phoneme set is shown in Table 1, with example words and corresponding SAMPA phonemes. The set includes new symbols for some of the SAMPA phonemes and introduces three more phonemes, /öo/, /üu/, /ea/, corresponding to allophones of the phonemes /o/, /u/, /a/, respectively. The decision to extend the SAMPA phoneme set with these three allophones was based on our experiments with the unit selection process. For instance, instead of considering the vowels 'o' in the words *kol* (*arm*) and *alkol* (*alcohol*) as corresponding both to the same phoneme /o/, representing the first one with /o/ and the second one with /öo/ enabled the unit selection algorithm to differentiate between these two phones.

## 3.1. Turkish pronunciation lexicon

A Turkish lexicon has been built containing about 3500 root words and their pronunciations. The lexicon is used to determine the pronunciations of the words and to expand the abbreviations and acronyms. The small size of the lexicon is because of the relatively simple pronunciation schema of Turkish compared to English. Turkish is a phonetic language in the sense that a simple grapheme-to-phoneme conversion (i.e. one-to-one mapping of letters to phonemes) is possible for most of the words due to the close relationship between orthography and phonology. Most of the words in the lexicon are those for which such a direct mapping cannot yield the correct pronunciation due to vowel lengthening, palatalization, etc., and most of them are loanwords originated from languages like Arabic and Persian [16]. For instance, the word *fedakarlık* (*sacrifice*) is pronounced as /f e d aa k ea r ll ı kk/, where the phonemes /aa/ and /ea/ are used for the fourth and sixth letters, respectively, instead of the standard phoneme /a/.

## 3.2. Text-to-phoneme conversion

The input text is first parsed into sentences and words by making use of space characters and punctuation marks. It is then stored in an internal data structure which is a linked list of sentence nodes, each of which is a linked list of word nodes. The sentence node structure was designed to hold sentence level information such as sentence type and the word node structure was designed to hold word level information such as POS

tagging and word pronunciation. At this stage, text normalization was also performed. The nonorthographic symbols are converted into orthographic ones in the sense that abbreviations and acronyms are expanded into full forms and digit sequences are converted into written forms. The characters that cannot be represented in speech are discarded. The punctuation marks are preserved.

**Table 1.** Turkish phoneme set.

| Phoneme | Example word | SAMPA | Phoneme | Example word | SAMPA |
|---|---|---|---|---|---|
| a | aşk | a | r | renk | r |
| b | bugün | b | s | ses | s |
| c | cuma | dZ | ş | şans | S |
| ç | çamur | tS | t | tat | t |
| d | dünya | d | u | uyku | u |
| e | evet | e | ü | ülke | y |
| f | futbol | f | v | veda | v |
| g | gece | gj | y | yeni | j |
| ğ | doğa | G | z | zaman | z |
| h | hayat | h | aa | alim | a: |
| ı | ışık | 1 | öo | alkol | N/A |
| i | insan | i | üu | sükunet | N/A |
| j | jüri | Z | uu | kanunen | u: |
| k | kedi | c | ii | milli | i: |
| l | lider | l | ea | kamil | N/A |
| m | mavi | m | ee | memur | e: |
| n | nisan | n | gg | gaga | g |
| o | oyun | o | kk | akıl | k |
| ö | özgürlük | 2 | ll | alkış | 5 |
| p | para | p | | | |

Turkish is an agglutinative language, that is, given a word in its root form, we can derive a new word by adding an affix (usually a suffix) to this root form, then derive another word by adding another affix to this new word, and so on. This iteration process may continue several levels. A single word in an agglutinative language may correspond to a phrase made up of several words in a non-agglutinative language. Thus, the text should be analyzed morphologically in order to determine the root forms and the suffixes of the words before further analysis [16, 17]. We used a morphological analyzer based on Augmented Transition Network (ATN) formalism [18]. The root word pronunciations are then looked up in the lexicon. If a root word cannot be found in the lexicon, the pronunciation is formed by a direct mapping of letters to phonemes in the phoneme set. This is also the case for suffixes: the pronunciations of all suffixes are formed in a direct manner. In this study, no linguistic analysis on syntax and semantics was done.

## 3.3. Prosodic analysis

Although the system was designed to use a prosodic analysis component, currently it does not include such a component. Prosody module can provide pitch, duration, and energy information which can be used in the unit selection process to synthesize the text. We plan to add pitch contour synthesis and duration modeling in future research. However, to evaluate the effect of using prosodic analysis, we tailored the system in such a way that it can optionally use transplanted prosody from the original speech utterances. Transplanted prosody means that the duration and intonation values from recorded speech are used in the unit selection

process [19]. This approach was used in the experiments to see the effect of real prosody on the output speech quality.

# 4. Unit Selection Using Viterbi Algorithm

The output of the analysis module is a sequence of phonemes corresponding to the input text, each having energy, pitch, and duration values. We refer to this sequence as the target sequence. The phones are used as the basic units in this research. The speech corpus had already been processed to build a unit inventory storing the phonemes with the same prosodic features (energy, pitch, duration) and the context information. Since we use a large speech database, there is more than one instance for each phoneme, each possibly having different phonetic context, and prosodic and acoustic realizations. Therefore, for each phoneme in the target sequence, there exist a large number of choices from the unit inventory. In concatenative speech synthesis, choosing the right units is very important for the quality of the synthesized voice. An appropriate selection of units may also allow to get rid of prosodic modifications of the selected units, which generally degrade the output speech quality. The unit selection module tries to choose the optimal set of units from the unit inventory that best match the target sequence.

Optimal unit selection from the unit inventory resembles the best-path decoding algorithm commonly used in HMM-based speech recognizers [13]. The speech unit inventory is analogous to the grammar network in HMM-based recognizers and can be considered as a state transition network. The best-path decoding of the words in the grammar is very similar to determining optimal unit sequence in the network of units, where the units form a trellis structure. The transition cost and the state observation cost in speech recognizers correspond, respectively, to the concatenation cost and the target cost in unit selection. This analogy guides us to the use of dynamic programming to find the optimal unit sequence. The pruned Viterbi search algorithm commonly used in HMM-based speech recognizers can be easily adapted to the problem of unit selection. The algorithm we used is a Viterbi best-path decoding algorithm that is very similar to the one used in CHATR speech synthesis system and is described below [13].

## 4.1. Measuring the similarity between a target sequence and a unit sequence

Given a target sequence $t_1^n = (t_1, ..., t_n)$ of phones with target duration, pitch and energy values, the problem is finding the unit sequence $u_1^n = (u_1, ..., u_n)$ that optimizes a cost function of the distance between the two sequences. As stated above, there are two kinds of cost function in unit selection, namely target cost and concatenation cost. Target cost (unit cost) is an estimate of the cost of using a selected unit in place of the target unit. This cost is a measure of how well the unit from the unit inventory suits the corresponding target unit in the target sequence. This cost can be calculated as a weighted sum of the target sub-costs, where each target sub-cost corresponds to a single feature of the units such as duration, pitch, energy, etc. and measures the cost (with respect to that feature) of using a selected basic unit in place of the target basic unit as follows:

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

where P is the number of target sub-costs and $w_j^t$ are the corresponding weights. We used the following form of this equation where P=4:

$$C^t(t_i, u_i) = \frac{20}{ContextMatchScore} + \left|\frac{f(t_i) - f(u_i)}{30}\right| + \left|\frac{D(t_i) - D(u_i)}{20}\right| + 10 * |E(t_i) - E(u_i)|,$$

where *ContextMatchScore* is the length of the matching context between the target unit and the selected unit, $f$ is the pitch frequency of a unit, $D$ is the duration of a unit and $E$ is the energy of a unit. The weights of the target sub-costs (i.e. coefficients in the equation above) were determined empirically by subjective listening tests.

The concatenation cost (join cost) is an estimate of the cost of concatenating two consecutive units. This cost is a measure of how well two units join together in terms of spectral and prosodic characteristics. The concatenation cost for two units that are adjacent in the unit inventory is zero. Therefore, choosing adjacent units in unit selection is promoted resulting in better speech quality. This cost can be calculated as a weighted sum of the concatenation sub-costs, where each concatenation sub-cost corresponds to a single feature of the units such as pitch, energy, etc. and measures the cost (with respect to that feature) of joining two units as follows:

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

where Q is the number of concatenation sub-costs and $w_j^c$ are the corresponding weights. We used the following form of this equation where Q=3:

$$C^c(u_i, u_{i+1}) = 10 * |c(u_i) - c(u_{i+1})| + \left|\frac{f(u_i) - f(u_{i+1})}{30}\right| + 10 * |E(u_i) - E(u_{i+1})|,$$

where $c$ is the cepstrum of a unit, $f$ is the pitch frequency of a unit and $E$ is the energy of a unit. The weights of the concatenation sub-costs (i.e. coefficients in the equation above) were determined empirically by subjective listening tests.

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

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

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

In calculating the target sub-costs $C_j^t(t_i, u_i)$, we use the context match length, energy, duration and pitch difference between the target and the selected units, and the location of the unit within the syllable, word and sentence. In calculating the cost for the context match, we also take into account the syllable boundaries in order to promote the selection of units having the same syllable structure. This is achieved by assuming the existence of a special character between two consecutive syllables and then determining the length of the match. For instance, for the words *elma* (*apple*) and *elim* (*my hand*), although the first

phonemes following *e* seem to match, when we consider these words as *el&ma* and *e&lim* (where & denotes syllable boundary), the first phonemes do not match. For the concatenation sub-costs $C_j^c(u_i, u_{i+1})$, we use the cepstral distance and the energy and pitch difference between the consecutive units. The cepstral distance at the concatenation points of two consecutive units is an objective measure of the spectral mismatch between these joining units. We use Mel-Frequency Cepstrum Coefficients (MFCCs) for this purpose. We extract the MFCC of the last frame of the first unit and the first frame of the second unit and use the distance between two MFCC vectors as the cepstral distance. For MFCC feature extraction, we used frames of 20 milliseconds.

### 4.2. Determining the optimal unit sequence

We implemented a Viterbi decoding algorithm to find the optimal unit sequence in the network of the nodes. A state transition network formed of the units in the speech inventory is shown in Figure 2, where the thick arrows indicate the connections between the selected units. The Viterbi algorithm tries to find the optimal path through the network [1, 13]. Since the number of units in unit inventory is very large, we employed some pruning methods to limit the number of units considered. By making use of a window size of three, for a target unit, we select only those units whose left and right three units are identical to those of the target unit. If there exist no such units, the search is repeated with a window size of two and finally with a window size of one.

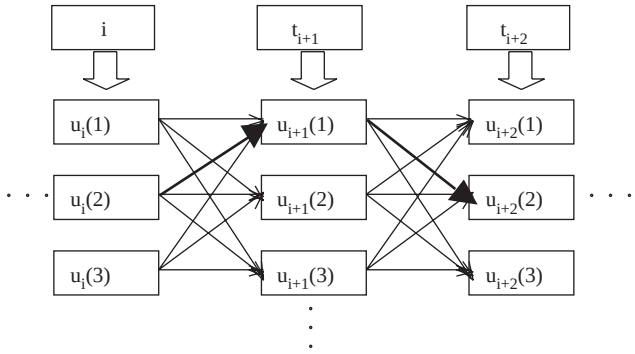


**Figure 2.** Unit selection using Viterbi algorithm.

## 5. Unit Concatenation and Waveform Generation

The unit selection module outputs a sequence of units from the speech inventory to be used for the generation of waveform for the input text. The waveform generation module concatenates the speech waveforms of the selected units. We used a speech representation and waveform generation method based on harmonic sinusoidal coding of speech [10, 11]. Analysis-by-synthesis technique was used for sinusoidal modeling.

The sinusoidal coding encodes the signal with a sum of sinusoids whose frequency, amplitude, and phase are adequate to describe each sinusoid. The harmonic coding is a special case of the sinusoidal coding where the frequencies of the sinusoids are constrained to be multiples of the fundamental frequency. The harmonic coding takes the advantage of the periodic structure of the speech and is very effective in coding voiced and unvoiced signals.

The harmonic coding is a parametric coding method. Unlike waveform coders which try to construct the original waveform, parametric coders (vocoders) try to encode the speech into a parametric representation that captures its perceptually important characteristics. Harmonic coders represent the speech signal using

the magnitudes and phases of its spectrum at multiples of the fundamental frequency. Low bit rate harmonic coders even use the synthetic phase rather than original phase to lower the bit rate. However, a high quality speech synthesis requires that the speech representation should be transparent to the listener. Therefore, we used the original phase in the harmonic coding of speech. The coded speech quality heavily depends on the correct parameter estimation. For robust parameter estimation, we used an analysis-by-synthesis methodology.

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

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

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

$$\varepsilon = \sum_{k=-T_0}^{T_0} \omega^2[k] \left(x[k] - \tilde{x}[k]\right)^2,$$

where $\omega$ is a Hamming window, $x$ is the real speech signal and $\tilde{x}$ is the harmonic model for the speech signal. For parameter estimation of the harmonic coding, we use this function for error minimization criterion. Finding model parameters is a least squares problem. The values for $A_k$ and $\phi_k$ that minimize the error are calculated by solving the linear set of equations obtained by differentiating the error function. The derivation of the linear equations is given in [11]. We used QR factorization method for solving the set of linear equations to obtain the model parameters.

The correct pitch period estimation is an important part of harmonic coding. The algorithm that we used for pitch estimation is based on the normalized autocorrelation method. The normalized autocorrelation is calculated as:

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

The search for the pitch was constrained to a region between 50Hz and 500Hz. We also performed some post-processing to smooth the pitch track, since the normalized autocorrelation method is error-prone. The smoothing process takes into consideration the factor that the pitch does not change drastically from frame to frame. We applied median smoothing that keeps a history of the pitch values, sorts it, and takes the one in the middle.

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

The waveform generation using the model parameters for speech waveforms of units is done by taking the inverse FFT of the parameters and then overlap-and-add mechanism is used for smooth concatenation of the units.

## 6. Experiments and Results

To evaluate the quality of the synthetic voice produced by the developed system, we carried out formal listening tests. The tests were of two type. The first one requires the listeners to rank the voice quality using a Mean Opinion Score (MOS) like scoring. The other test is a diagnostic rhyme test.

MOS tests are commonly used for both evaluating the effectiveness of speech coding algorithms and assessing the quality of synthesized speech. The MOS scores for speech synthesis are generally given in three categories: intelligibility, naturalness, and pleasantness.

The MOS test was carried out by synthesizing a set of 50 sentences that were selected from the speech corpus randomly and did not participate in the training set. The reason of choosing the sentences for which we have also available the original speech waveforms is that the original recordings are also used in the tests to ensure the reliability of the test results. 10 subjects (2 females) were used and they listened the sentences using headphones. The sentences were at 16kHz and 16 bits. The subjects were instructed to rate the sentences on a scale of 1-5 where 1 is very poor and 5 is excellent. Some speech samples of speech coders having different MOS scores were presented to the subjects to ensure consistency in evaluating the speech quality. The subjects were also familiarized with the speech synthesis by listening some example utterances of varying quality.

We built five different systems and evaluated their quality. The first system uses the original recordings from the test speech corpus that were coded by our harmonic coder and reconstructed. The second system uses the unit selection synthesizer with a speech unit inventory containing about 19 hours of speech recording. The third system uses a speech inventory containing about 3 hours of recording. The latter two systems do not use prosody information and no prosody targets are specified for the target units in unit selection. The last two systems are the same as the previous two, except that the original prosody from the original recordings is used in the unit selection process [19].

Each of the 50 test sentences were synthesized by each of the five systems.[1] Then five test sets were constructed in the following way: 10 sentences from each system were gathered to form a test set. Each set contained all of the 50 test sentences, i.e. repeating of the same sentence from different systems was not allowed. The subjects were also divided into five groups with two subjects in each. Then each test set was listened by a different group. The subjects gave ratings in terms of intelligibility, naturalness, and pleasantness to each sentence. The average MOS scores are shown in descending success rates in Table 2. Figures 3 and 4 show the scores for each system and category. The differences in system ratings were found to be significant using ANOVA analysis. The analysis yielded an F-value of about 21 whereas the critical F-values are about 3.3 and 5.0 for P=0.01 and P=0.001, respectively.

It is quite interesting that while system C is better than system E both of which use 3 hours of speech, this is not the case for systems D and B which use 19 hours of speech. In other words, for 3 hours of speech corpus, using original prosody improves the naturalness of generated speech, whereas for 19 hours of speech corpus, it degrades the generated speech quality. It can be argued that for systems that use relatively less amount of speech corpus, using prosody information in unit selection helps to select better units in terms of prosody, hence increasing the overall naturalness of synthetic speech. On the other hand, for larger corpus, we have more units in the corpus and the unit selection is more probable to find a better acoustic and prosodic match. In these systems, using prosody information may cause the unit selection to favor prosody over acoustic appropriateness which is probably more important than prosody for naturalness.

[1] The test sentences and their synthesized forms for each system can be found in http://www.cmpe.boun.edu.tr/∼gungort/publications/turkishttssamples.htm.

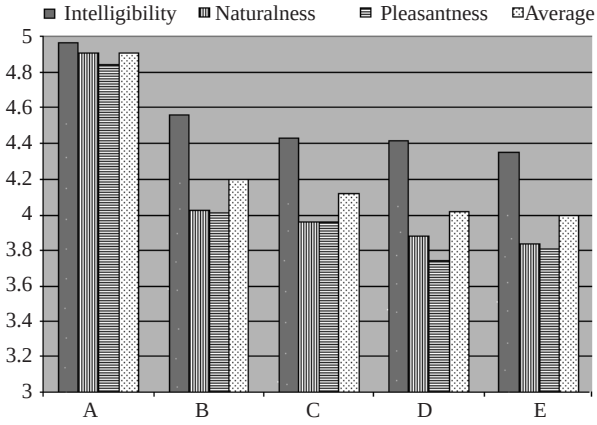


**Figure 3.** MOS scores with respect to system type.

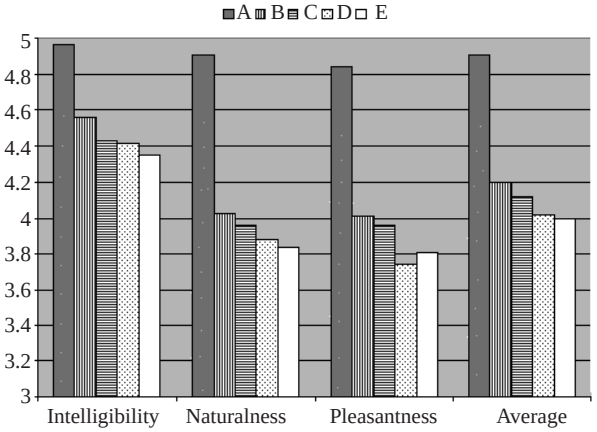


**Figure 4.** MOS scores with respect to test category.

**Table 2.** Systems and average scores for the MOS test.

| System | Description | MOS |
|---|---|---|
| A | The original recordings with harmonic coding | 4.91 |
| B | Speech synthesis using 19 hours of speech | 4.20 |
| C | Speech synthesis using 3 hours of speech with original prosody | 4.11 |
| D | Speech synthesis using 19 hours of speech with original prosody | 4.01 |
| E | Speech synthesis using 3 hours of speech | 4.00 |

We also conducted an intelligibility test. Diagnostic Rhyme Test (DRT) uses monosyllabic words that have consonant-vowel-consonant pattern. This test measures the capability of discrimination of the initial consonants for the system evaluated. The DRT word list of ANSI standard for English contains 192 words arranged in 96 rhyming pairs which differ only in their initial consonant sounds. The list has been divided into six categories depending on the distinctive features of speech. The categories have been constructed in terms of voicing, nasality, sustenation, sibilation, graveness, and compactness characteristics of the sounds. For assessing the intelligibility of the synthesized speech in Turkish, we constructed a DRT word list for Turkish based on the categories of the DRT word list of English as shown in Table 3. The DRT list was designed to exploit the distinctive features of Turkish speech at maximum.

Using the DRT word list for Turkish, we carried out an intelligibility test for our system. The randomly selected words from each pair of the DRT word list were synthesized using the system. The output speech

waveforms were played to 10 native Turkish listeners who were then asked to choose which one of the words given in pairs from the DRT list they heard. The listeners were assured to have a good hearing and discrimination of sounds. The test results are shown in Table 4 as the percentage of the number of correct selections for the two systems evaluated.

**Table 3.** DRT word list for Turkish.

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

**Table 4.** Systems and average scores for the DRT test.

| **System** | **Description** | **DRT** |
|---|---|---|
| B | Speech synthesis using 19 hours of speech | 0.95 |
| E | Speech synthesis using 3 hours of speech | 0.94 |

By analysing the MOS and DRT tests conducted, we have also identified the main problems and limitations of the developed system. The major sources of errors degrading synthesized speech quality are as follows: Misalignment of phones in the speech database, prosody related problems such as pitch contour discontinuities, timing errors for phones, energy differences between phones, and errors caused by acoustic variations of phones in different contexts. The latter one shows itself in the concatenation of phones from different contexts due to the lack of phones with similar contexts.

# 7. Conclusions and Future Work

In this paper, a corpus-based concatenative speech synthesis system architecture for Turkish has been proposed and implemented. A new Turkish phoneme set that is suitable and adequate for representing all the sounds in Turkish was given. A pronunciation lexicon for the root words in Turkish has been prepared. A text normalization module and a grapheme-to-phoneme conversion module based on morphological analysis of Turkish have been implemented. Speech corpus has been compressed by a factor of three with slight degradation on the voice quality using the harmonic coding based speech model. As the final system, a unit selection based concatenative speech synthesis system capable of generating highly intelligible and natural synthetic speech for Turkish has been developed. Subjective tests have been carried out to assess

the speech quality generated by the system. A DRT word list for Turkish has been constructed to carry out the intelligibility tests. The final system got 4.2 MOS like score and 0.95 DRT correct word discrimination percentage.

As future work, the developed system for Turkish can be enhanced by adding prosody generation module, which should address intonation and duration modeling of the language. Duration analysis and modeling of Turkish has been studied in [20]. Intonation and stress characteristics in Turkish sentences have also been investigated [21]. We may also use the methods in these studies to form some rules, which can be used in order to improve the prosody of synthetic speech in the absence of a prosody module for Turkish. The unit selection algorithm can be further improved by automatically training the target and transition cost weights using the objective cost measures. Furthermore, to reduce the runtime complexity of the unit selection process, some methods based on pre-selection of units can be implemented, which can also reduce the size of the speech unit database.

# References

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

[2] A.R. Greenwood, "Articulatory Speech Synthesis Using Diphone Units", IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1635-1638, 1997.

[3] Y. Sagisaka, N. Iwahashi and K. Mimura, "ATR v-TALK Speech Synthesis System", Proceedings of the ICSLP, Vol. 1, pp. 483-486, 1992.

[4] A.W. Black and P. Taylor, "CHATR: A Generic Speech Synthesis System", International Conference on Computational Linguistics, Vol. 2, pp. 983-986, 1994.

[5] M. Beutnagel, A. Conkie, J. Schroeter, Y. Stylianou and A. Syrdal, "The AT&T Next-Gen TTS System", Proceedings of the Joint Meeting of ASA, EAA, and DAGA, pp. 18-24, Berlin, Germany, 1999.

[6] R.E. Donovan and E.M. Eide, "The IBM Trainable Speech Synthesis System", International Conference on Spoken Language Processing, Vol. 5, pp. 1703-1706, Sydney, 1998.

[7] R.E. Donovan, "Current Status of the IBM Trainable Speech Synthesis System", Proceedings of the 4th ISCA Tutorial and Research on Speech Synthesis, Edinburgh, 2001.

[8] A.W. Black and P. Taylor, "Automatically Clustering Similar Units for Unit Selection in Speech Synthesis", European Conference on Speech Communication and Technology, Vol. 2, pp. 601-604, Rhodos, 1997.

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

[10] D.G. Rowe, Techniques for Harmonic Sinusoidal Coding, Ph.D. Thesis, University of South Australia, 1997.

[11] Y. Stylianou, "Applying the Harmonic Plus Noise Model in Concatenative Speech Synthesis", IEEE Trans. on Speech and Audio Processing, Vol. 9, No. 1, pp. 21-29, 2001.

[12] Y. Stylianou, "Removing Phase Mismatches in Concatenative Speech Synthesis", 3rd ESCA/COCOSDA Workshop on Speech Synthesis, Australia, 1998.

[13] A. Hunt and A.W. Black, "Unit Selection in a Concatenative Speech Synthesis System Using a Large Speech Database", IEEE International Conference on Acoustics and Speech Signal Processing, Vol. 1, pp. 373-376, Germany, 1996.

[14] A. Conkie, "Robust Unit Selection System for Speech Synthesis", Proceedings of the Joint Meeting of ASA, EAA and DEGA, Berlin, Germany, 1999.

[15] http://www.phon.ucl.ac.uk/home/sampa/home.htm.

[16] K. Oflazer, "Two-level Description of Turkish Morphology", Literary and Linguistic Computing, Vol. 9, No.2, 1994.

[17] K. Oflazer and S. Inkelas, "A Finite State Pronunciation Lexicon for Turkish", Proceedings of the EACL Workshop on Finite State Methods in NLP, Budapest, Hungary, 2003.

[18] T. Güngör, Computer Processing of Turkish: Morphological and Lexical Investigation, Ph.D. Thesis, Boğaziçi University, 1995.

[19] F. Spyns, F. Deprez, L.V. Tichelen and B.V. Coile, "Message-to-Speech: High Quality Speech Generation for Messaging and Dialogue Systems", Proceedings of the ACL/EACL Workshop on Concept to Speech Generation, pp. 11-16, 1997.

[20] Ö. Şaylı, L.M. Arslan and A.S. Özsoy, "Duration Properties of the Turkish Phonemes", International Conference on Turkish Linguistics, KKTC, 2002.

[21] E. Abdullahbeşe, Fundamental Frequency Contour Synthesis for Turkish Text-to-Speech, M.S. Thesis, Boğaziçi University, 2001.