Demiroğlu and Güner *EURASIP Journal on Audio, Speech, and Music Processing* (2016) 2016:4
DOI 10.1186/s13636-016-0082-0

EURASIP Journal on Audio, Speech, and Music Processing
a SpringerOpen Journal

RESEARCH Open Access

Hybrid statistical/unit-selection Turkish speech synthesis using suffix units

Cenk Demiroğlu* and Ekrem Güner

Abstract

Unit selection based text-to-speech synthesis (TTS) has been the dominant TTS approach of the last decade. Despite its success, unit selection approach has its disadvantages. One of the most significant disadvantages is the sudden discontinuities in speech that distract the listeners (Speech Commun 51:1039–1064, 2009). The second disadvantage is that significant expertise and large amounts of data is needed for building a high-quality synthesis system which is costly and time-consuming. The statistical speech synthesis (SSS) approach is a promising alternative synthesis technique. Not only that the spurious errors that are observed in the unit selection system are mostly not observed in SSS but also building voice models is far less expensive and faster compared to the unit selection system. However, the resulting speech is typically not as natural-sounding as speech that is synthesized with a high-quality unit selection system. There are hybrid methods that attempt to take advantage of both SSS and unit selection systems. However, existing hybrid methods still require development of a high-quality unit selection system. Here, we propose a novel hybrid statistical/unit selection system for Turkish that aims at improving the quality of the baseline SSS system by improving the prosodic parameters such as intonation and stress. Commonly occurring suffixes in Turkish are stored in the unit selection database and used in the proposed system. As opposed to existing hybrid systems, the proposed system was developed without building a complete unit selection synthesis system. Therefore, the proposed method can be used without collecting large amounts of data or utilizing substantial expertise or time-consuming tuning that is typically required in building unit selection systems. Listeners preferred the hybrid system over the baseline system in the AB preference tests.

Keywords: Statistical speech synthesis, Hybrid speech synthesis, Suffix selection, Turkish

1 Introduction

The HMM-based text-to-speech (SSS) approach has been shown to generate good quality and intelligible speech [1]. However, well-tuned unit selection systems generated with substantially larger amounts of training data compared to SSS systems typically produce more natural speech compared to SSS-based systems. Still, spurious errors in unit selection systems can significantly hurt listener preference [2]. Hybrid SSS/unit selection methods typically attempt to improve the quality of the unit selection systems by generating speech that is smooth as in the SSS approach but also natural-sounding as in the unit selection approach.

*Correspondence: cenk.demiroglu@ozyegin.edu.tr
Electrical and Computer Engineering Department, Ozyegin University, Orman Street, 34794 Istanbul, Turkey

Hybrid methods can be divided into several categories. In one approach, SSS system is used for computing the target cost in unit selection. In that case, parameters of the SSS acoustic model can be used to compute likelihoods of candidate units [3, 4] or distance of candidate unit parameters to SSS-generated parameters can be used for target cost computation [5]. In a second approach, SSS-generated waveforms are interweaved with the speech units selected from the database [6, 7]. The idea is to use smooth SSS-generated waveforms when a unit with a low cost cannot be found in the database. There are also hybrid systems that aim to smooth out the transitions between the units in the concatenative approach using the smooth trajectories of the SSS approach [8].

Excitation signal is important for generating natural sounding speech. In [9], excitation signal is extracted from natural speech and stored in a database. During synthesis with the SSS approach, the closest excitation signal in the

database to the target synthetic excitation signal is used for synthesis. Careful labelling, excitation extraction, and tuning is required to obtain high-quality speech with [9]. In [10], prosodic parameters are generated with the SSS approach while rest of the synthesis is done using waveform concatenation. Similarly, in [11], prosody prediction using SSS was done for better unit selection.

The work proposed in [12] is one of the very few examples where the goal of the hybrid approach is to improve the SSS system as opposed to improving the unit selection system. In [12], during synthesis time, the target utterance is first synthesized with a unit selection system and then the SSS parameters are modified to generate parameters that are as close as possible to the synthesized parameters with unit selection. Thus, for each target utterance, parameters are retuned for best results.

The proposed system exploits the morphologically rich structure of the Turkish words to keep the unit selection database small. In Turkish, many different words can be generated from the same root word by using a limited set of suffixes. Given a typical Turkish utterance, a significant number of the words contain one or more suffixes. Moreover, ignoring silences, approximately one fourth of the speech is composed of suffixes. Furthermore, suffixes contain significant linguistic information such as word stress. Using a limited set of suffix units within the proposed hybrid approach, significant improvements in the quality of the SSS system is obtained without requiring additional data collection or careful tuning of the system.

Syllable-based speech synthesis has been successful both in unit selection and HMM-based systems [13]. However, suffixes in Turkish are not necessarily syllables. Moreover, hybrid systems that focus only on particular set of syllables, suffixes, do not exist in the literature. Even though the work in [14] exploits the perceptually-important consecutive voiced speech (CVS) segments in a hybrid Mandarin synthesis algorithm, the segments in [14] are more general compared to the suffixes used here. Moreover, a complete unit selection system is still used in [14].

The proposed system is novel in several aspects. As opposed to most of the existing hybrid methods that are focused on improving the quality of a unit selection system, here, we propose a hybrid SSS/unit selection algorithm to boost the quality of our Turkish SSS system. Moreover, in the existing hybrid systems, because a unit selection system is required, cheap and fast voice building is not possible. The goal of the proposed approach is to take advantage of the hybrid synthesis idea to improve the quality of the SSS system while retaining its cheap and fast voice building advantage. A key novelty in this work is that the proposed system does not increase the training data requirements compared to the SSS systems which is far less than what is needed for building a good-quality unit selection system.

This paper is organized as follows. We first do a brief review of the existing parameter generation algorithms for SSS in Section 2. An overview of the proposed hybrid system is given in Section 3. The morphological analyzer used here is described in Section 4. The proposed suffix prefiltering and suffix selection algorithms are presented in Section 5. The proposed hybrid parameter generation algorithm is described in Section 6. Experimental results of the SSS and hybrid systems are reported and discussed in Section 7. Finally, a conclusion is done in Section 8.

2 Review of parameter generation algorithms

Statistical speech synthesis systems use parametric vocoders for synthesis. Therefore, before vocoding, speech parameters should be generated using a parameter generation algorithm. Below, we first describe the SSS approach to parameter generation. Then, the hybrid statistical/unit selection approach using a constrained optimization technique is described.

2.1 Statistical parameter generation

In the statistical approach, the first phase of synthesis is to generate a sequence of phonemes from text with an associated context for each phoneme. Phonemes are modelled with hidden Markov models (HMMs) that are concatenated to represent the final utterance. Because pitch and spectral envelope are modelled independently, separate sequences of HMMs are used for them.

During acoustic model training, HMM states that are observed in the training data are clustered using decision trees to avoid overfitting. Sequence of states for a given phoneme is identified using the decision tree and context of the phoneme.

Once the states, therefore their emission and duration pdf parameters, are known, the parameter sequence O for spectral envelope and pitch can be generated using

$$\widehat{O} \approx \arg\max_{O} \max_{Q} p(O|Q,\lambda)p(Q|\lambda), \tag{1}$$

where $Q = \left[q_1, q_2, \ldots, q_{N_u}\right]$ is a vector that contains the id of each HMM state, q_i, at frame i and λ corresponds to HMM model parameters. N_u is the total number of frames in the utterance.

Equation (1) can be simplified by maximizing the state-sequence independently. In this case,

$$\widehat{Q} = \arg\max_{Q} p(Q|\lambda) \tag{2}$$

and

$$\widehat{O} = \arg\max_{O} p\left(O|\widehat{Q},\lambda\right). \tag{3}$$

Parameter O contains static, delta, and delta-delta features. However, the vocoder only needs the static features c to generate speech. To estimate c, Eq. (3) can be written as

$$\widehat{c} = \arg\max_{c} p\left(Wc|\widehat{Q},\lambda\right), \tag{4}$$

where W is used to derive the delta and delta-delta features from the static features c.

The solution to Eq. (4) is

$$\widehat{c} = \left(W^T U^{-1} W\right)^{-1} W^T U^{-1} M, \tag{5}$$

where $M = \left[\mu_{q_1}^T,\ \mu_{q_2}^T,\ \ldots.,\ \mu_{q_{N_u}}^T\right]^T$ and the block diagonal matrix $U^{-1} = diag\left[U_{q_1}^{-1}, U_{q_2}^{-1}, \ldots.,\ U_{q_{N_u}}^{-1}\right]$. μ_{q_i} is the mean vector and $U_{q_i}^{-1}$ is the inverse covariance matrix of the emission pdf of the HMM state q_i at frame i.

Once the parameters are estimated independently for spectral envelope and pitch for the whole utterance, a parametric speech vocoder is used to synthesize the speech signal.

2.2 Hybrid parameter generation

Although SSS generates smooth feature trajectories, which eliminate the annoying glitches that are typically observed in the unit selection system, the quality of speech is higher in the unit selection systems when these glitches do not occur. Hybrid systems attempt to generate high-quality speech without the glitches using a combination of unit selection and SSS approaches.

For hybrid synthesis, we use the approach in [15] where hybrid parameter generation is posed as a constrained optimization problem. In that approach, natural speech frames are scattered throughout the utterance and the rest of the frames are generated using SSS. The parameter generation algorithm is formulated such that features that constitute the k^{th} frame, c_k, are constrained to be equal to the natural speech frame $c_{l,nat}$ if it exists. Given a total of K frames, L of which are natural frames, a hybrid estimate of the static features can be formulated as the constrained optimization problem

$$\widehat{c}_h = \arg\max_{c} p\left(Wc|\widehat{Q},\lambda\right), \tag{6}$$

provided that

$$A\widehat{c}_h = c_{nat}, \tag{7}$$

where $c = \left[c_1^T,\ c_2^T,\ \ldots,\ c_K^T\right]^T$, the vector of natural features $c_{nat} = \left[c_{1,nat}^T,\ c_{2,nat}^T,\ \ldots,\ c_{L,nat}^T\right]^T$, and the block matrix $A = \left[A_1^T,\ A_2^T,\ \ldots,\ A_L^T\right]^T$ where A_l is also a block matrix. The block matrix A_l is a $F \times FK$ matrix consisting of K square matrices $A_l = [A_l(1), \cdots, A_l(K)]$, each square matrix of which is given by

$$A_l(k) = \begin{cases} I_{F\times F} & k = b_l \\ 0_{F\times F} & \text{otherwise} \end{cases}$$

where b_l^{th} frame is constrained to be equal to the natural speech frame $c_{l,nat}$, $I_{F\times F}$ is an identity matrix, and F is the number of dimensions of the feature vector per frame. Using the Lagrange multipliers $\gamma = [\gamma_1, \cdots, \gamma_L]^T$, the parameter generation problem becomes

$$\widehat{c}_h = \arg\max_{c} p\left(Wc|\widehat{Q},\lambda\right) - \gamma^T\left(Ac - c_{nat}\right). \tag{8}$$

Solution to Eq. (8) is [15]

$$\widehat{c}_h = \widehat{c} + \left(W^T U^{-1} W\right)^{-1} A^T \gamma, \tag{9}$$

where

$$\gamma = \left(A\left(W^T U^{-1} W\right)^{-1} A^T\right)^{-1} c_{nat} - \left(A\left(W^T U^{-1} W\right)^{-1} A^T\right)^{-1} A \left(W^T U^{-1} W\right)^{-1} W^T U^{-1} M.$$

A comparison of hybrid synthesis and SSS is shown in Fig. 1. Hybrid trajectory follows the natural trajectory during suffixes and synchronizes back with the synthetic trajectory in the other morphemes in Fig. 1.

3 Overview of the proposed system

An overview of the training and synthesis algorithms of the proposed system is shown in Fig. 2. A brief description of the proposed system is given below. Details of the morpheme analyzer, the suffix selection, and the hybrid parameter generation algorithms are described in Sections 4, 5, and 6, respectively.

The proposed system is based on exploiting the suffixes in Turkish in a hybrid synthesis approach. Turkish has a special suffix structure where a small number of suffixes occur frequently in text. Hence, using suffixes for hybrid synthesis makes a significant impact on the performance. Because a complete unit selection synthesis system is not needed, the expensive processes of building a unit selection system or additional data collection are avoided.

Suffix database is created as follows. First, statistical synthesis models are generated for a target speaker using a speaker-dependent training algorithm. Then, a morphological analyzer is used to identify the suffixes in the training database. To create a suffix database, feature segments that correspond to the suffixes labeled by the morphological analyzer should be extracted from speech. To that end, forced alignment is used to align text and speech features using the speaker-dependent HMM models. Using the alignment output, feature segments for each suffix are extracted and stored in the suffix database.

Each suffix entry in the suffix selection database contains feature vectors for the suffix segment in addition to the context information of the suffix. Feature vectors are composed of line spectral frequencies (LSF) and fundamental frequency. Context information includes features

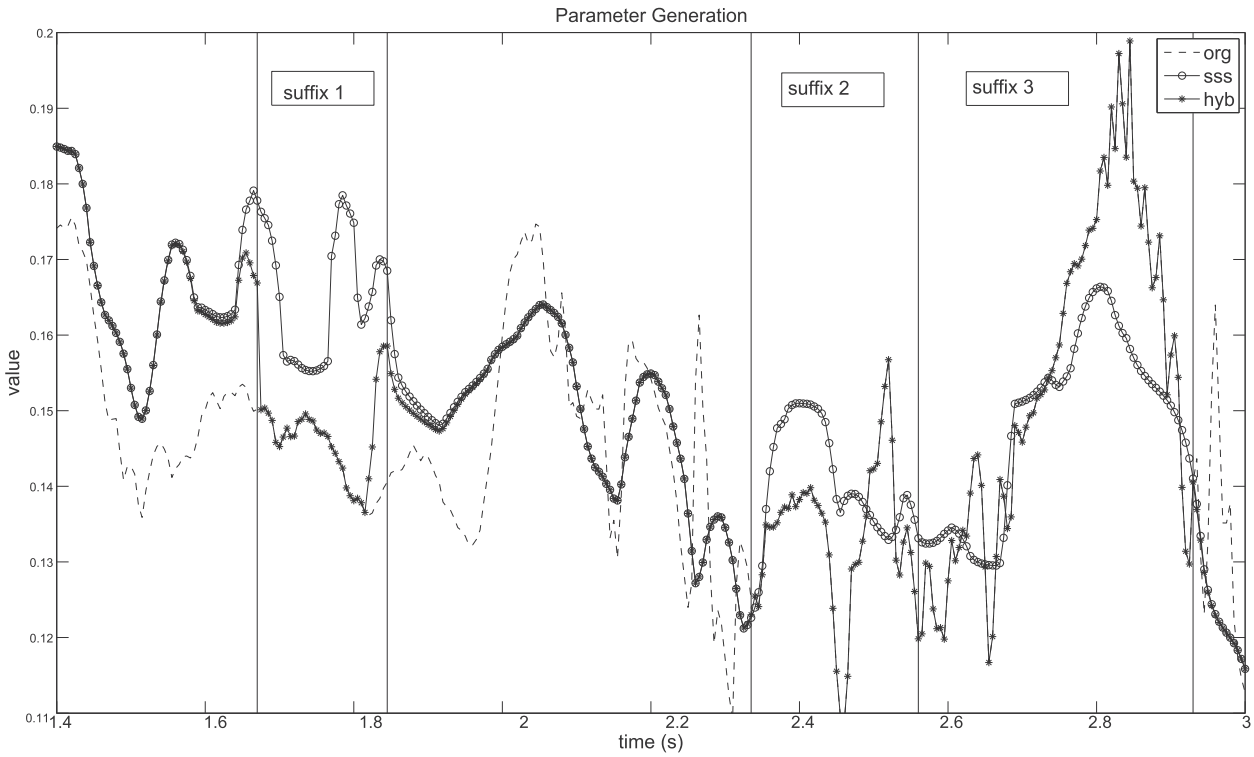

Fig. 1 Parameter trajectories for a spectral feature using the baseline SSS system and hybrid system compared with the original speech. Hybrid system follows the natural trajectory during suffixes and synchronizes back with the SSS trajectory when a suffix is not available

such as the position of the suffix in the phrase and the presence of stress. A complete list of context features are shown in Table 1. State-level duration information for each suffix is also stored in the database.

At synthesis time, input text is analyzed using the morphological analyzer. For each suffix in the synthesized utterance, the best fitting suffix is selected from the suffix database using the algorithm proposed in Section 5.2.2. After selecting the suffixes, the hybrid parameter generation algorithm described in Section 6 is used, and the generated parameter sequences are fed to a vocoder to synthesize speech.

The algorithms used in the synthesis process are described in more detail below.

4 Morphological analyzer

The finite state transducer (FST)-based morphological analyzer described in [16] is used here to identify suffixes. The analyzer generates the root morpheme and the suffixes 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 output of the morphological analyzer is

```
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 is composed of four suffixes. Derivational phonemes are indicated by the DB tag. Note that after every derivation, the new part-of-speech tag 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 suffixes 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 [17]. Here, manual disambiguation is done.

5 Suffix selection

Suffix selection is done in two steps. In the first step, context-dependent prefiltering is done to reduce the set of alternative units for a suffix. Then, a suffix selection algorithm is used to select suffixes for each suffix position. Algorithms that are used in both steps are described below.

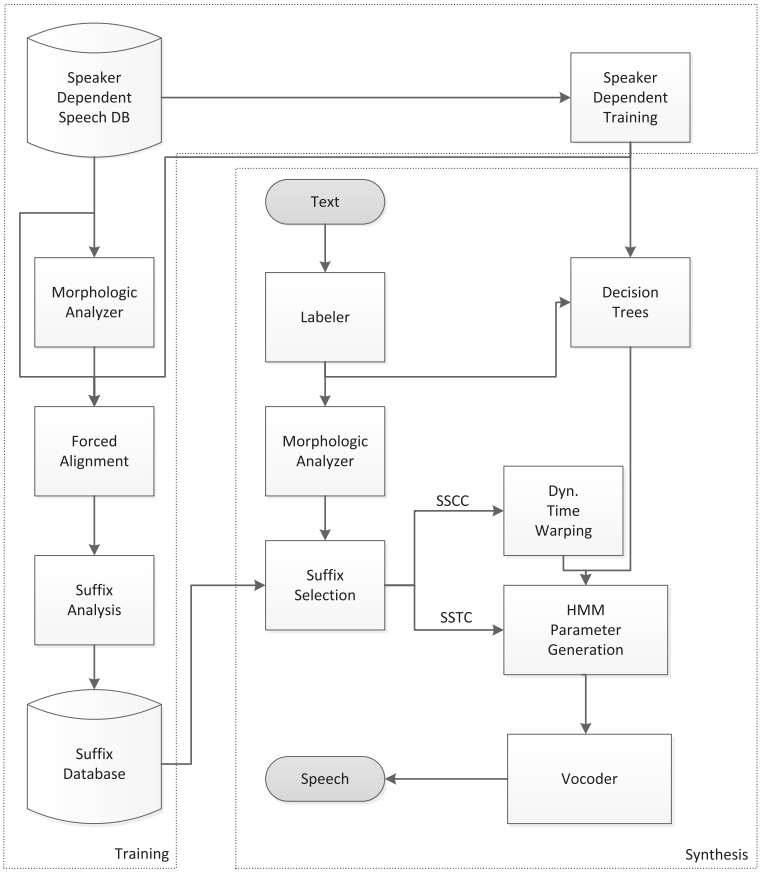

Fig. 2 Overview of the proposed hybrid system

5.1 Suffix prefiltering

A decision tree-based suffix prefiltering algorithm is used. In this approach, suffixes are clustered using decision trees depending on their contexts. The syllable, word, and phrase level features shown in Table 1 are used during the decision tree-building process. Note that because there are not too many instances available for each suffix, a restricted set of questions are used to avoid overfitting.

Table 1 Linguistic questions used in the decision tree-based clustering of suffixes

Syllable-level	Stress: What is the stress level of the syllable that contains the suffix? (0, 1, 2)
Word-level	Position in the word: Is the suffix at the end of the word? (yes, no)
Phrase-level	Position in the phrase: Is the word containing the suffix at the of the phrase? (yes, no)

Note that classical state-tying methods of the speech recognition field cannot be used here for clustering the suffixes because each suffix is made up of more than one HMM state. Moreover, the goal here is to generate speech and not recognize it, which requires a different perspective in selecting the distance measure for clustering.

In the decision tree approach, for each suffix, all instances of the suffix are pooled together at a root node. Then, starting from the root node, nodes are split using a minimum Kullbeck-Leibler (KL) divergence criterion. Splitting stops when one of the leaf nodes have less instances than a threshold.

The linguistic question that minimizes the sum of average KL divergences, $\mathrm{D}_{KL}(S_y) + \mathrm{D}_{KL}(S_n)$, of the children nodes is used to split each node in the decision tree. S_y and S_n are the sets of suffixes clustered in the children nodes. Average KL divergence of a set of suffixes S is defined as

$$\mathrm{D}_{KL}(S) = \frac{1}{N_S} \sum_{i=1}^{N_S} \mathrm{D}_{KL}(S||S_i), \tag{10}$$

where S_i are each of the suffixes in the suffix set S and N_S is the total number of suffixes in the set. $D_{KL}(S||S_i)$ is defined as

$$\mathrm{tr}\left(\Sigma_i^{-1}\Sigma_s\right)+(\mu_i-\mu_s)^T\,\Sigma_i^{-1}\,(\mu_i-\mu_s)+\ln\frac{|\Sigma_i|}{|\Sigma_s|}, \quad (11)$$

where tr(.) is the trace operator, $\{\mu_i, \Sigma_i\}$ are the parameter vectors representing the suffix instance S_i and $\{\mu_S, \Sigma_S\}$ are the parameter vectors representing the suffix set S. μ_s is the average of the mean vectors of S_i and Σ_s is the average of the covariance matrices of S_i. The problem then is how to compute μ_i and Σ_i for each S_i.

Two algorithms are investigated to estimate μ_i and Σ_i. In the first approach, parameters of the emission pdf of pitch features for each state j of each suffix instance i, $\left\{\mu_{p,i}^{(j)}, \Sigma_{p,i}^{(j)}\right\}$, can be concatenated to obtain the super vectors

$$\mu_i=\left[\mu_{p,i}^{(1)^T},\mu_{p,i}^{(2)^T},\cdots,\mu_{p,i}^{(N_{s,i})^T}\right]^T \quad (12)$$

and

$$\Sigma_i=diag\left(\Sigma_{p,i}^{(1)}\ \Sigma_{p,i}^{(2)}\ \ldots\ \Sigma_{p,i}^{(N_{s,i})}\right), \quad (13)$$

where the $diag(.)$ operator creates a block diagonal matrix with $\Sigma_{p,i}^{(j)}$ at the diagonal position j and $N_{s,i}$ is the total number of states in the suffix.

The first approach did not work well because state emission pdfs are obtained by averaging which causes smoothing in parameter trajectories. Therefore, rapid variations in the training instances are not represented well with the emission pdf parameters. That results in degradation in measuring distances between suffixes which in turn causes inaccurate clustering.

In the second approach, the training data for each suffix instance S_i is used directly as opposed to the emission pdf parameters. In this approach, each suffix S_i is first state-aligned with the HMM states using forced alignment. Then, frames that occur in the middle of each state are used to represent the mean, $\mu_{p,i}^{(j)}$, of state j. This has the advantage of not losing rapid variations that occur within S_i. Because there are typically not enough samples to estimate the covariance within a state, covariance matrix of the emission pdf is used.

Note that pitch is only defined for voiced states. To define the pitch parameter for unvoiced states, linear interpolation is used. For the first approach above, μ_p parameters in the neighboring voiced states are interpolated to define the mean pitch for the unvoiced states. For the second approach, pitch parameters during unvoiced states are found by linear interpolation of pitch parameters extracted in the neighboring voiced states.

Gross mismatch between the selected suffix duration and the synthetic suffix duration can significantly hurt the naturalness of speech. To avoid the issue, during suffix selection, additional prefiltering is applied to suffixes to ensure that the selected suffixes are at least as long as the synthetic ones and not longer than ζ_d times the synthetic suffix durations. ζ_d is set experimentally.

5.2 Suffix selection algorithms

In a typical unit selection based TTS system, target cost and concatenation cost are used in selecting the units. 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. The total cost of using unit j in the suffix database for suffix position k in the utterance is

$$\phi_{j,k}=w_1C_{j,k}^{(p)}+w_2C_{j,k}^{(s)}+w_3T_{j,k}^{(p)}+w_4T_{j,k}^{(s)}, \quad (14)$$

where $C_{j,k}^{(p)}$ is the concatenation cost of the pitch parameter, $C_{j,k}^{(s)}$ is the concatenation cost of the spectral parameters, $T_{j,k}^{(p)}$ is the target cost of the pitch parameter, $T_{j,k}^{(s)}$ is the target cost of the spectral parameters and w terms are the weights.

Two suffix selection algorithms are investigated here. In the first approach, only target costs are used and weights of concatenation costs are set to zero. In the second approach, both target and concatenation costs were used. The two algorithms for suffix selection are described below.

5.2.1 *Suffix selection with target cost (SSTC)*

When synthesizing an utterance u, suffixes $s^{(l)}$ in the utterance are first determined using a morphological analyzer, where $l=1,2,\ldots,N_{suf}$, and N_{suf} is the total number of suffixes in the utterance. For the j^{th} suffix, the set of candidate units in the database is denoted by $\left\{S_1^j\right\}$. The candidate set is generated using the decision tree-based prefiltering described in Section 5.1.

Two different suffixes are selected for LSF and pitch parameters. Durations of those suffixes are time-warped to fit the duration predicted by SSS. Such time-warping did not cause significant artifact with LSF features. However, that was not the case with the pitch features. Even though expanding the pitch trajectory did not cause any audible artifacts, compressing the pitch trajectory occasionally caused sudden pitch changes which were perceived as artifacts by the listener. To avoid that problem, the units in $\left\{S_1^j\right\}$ that are R_d percent longer than the synthetic duration of the suffix are filtered out. The reduced set of suffixes after filtering is denoted by $\left\{S_2^j\right\}$.

The proposed suffix selection algorithm uses a maximum likelihood (ML) criterion as the target cost. For

the j^{th} candidate unit and suffix position l, the average log-likelihood is computed by

$$\mathrm{LL}_j^{(l)} = \frac{1}{\Gamma_j}\sum_{i=1}^{I_l}\sum_{f=1}^{\Gamma_{j,i}} log\left[\frac{1}{(2\pi)^{D/2}}\frac{1}{|\Sigma_{i,l}|^{1/2}}\right] -\frac{1}{2}\left(X_{j,i}^{(f)}-\mu_l^{(i)}\right)^T \Sigma_l^{(i)^{-1}}\left(X_{j,i}^{(f)}-\mu_l^{(i)}\right), \quad (15)$$

where Γ_j is the total number of frames in the unit, I_l is the total number of states in the suffix, $\Gamma_{j,i}$ is the total number of frames in state i of unit j, $\Sigma_l^{(i)}$ is the covariance matrix in state i, $\mu_l^{(i)}$ is the mean vector in state i, and $X_{j,i}^{(f)}$ is the f^{th} observation of state i and unit j. $X_{j,i}^{(f)}$ contains static, delta, and delta-delta features.

SSTC algorithm described above has the advantage of not requiring any tuning of weights in Eq. (14). This has two reasons. The first reason is that different suffix units are used for pitch and LSF features and suffix selection is done independently for those two features. The second reason is that concatenation cost is not used.

Since a limited set of prefiltered suffixes are used, we initially hypothesized that a suffix selection algorithm without the concatenation cost would work well. However, experimental results showed that likelihood-based target cost computation results in selecting suffixes with overly-smooth trajectories which causes reduction in speech quality. To avoid the problem and enable selection of suffixes with more variability, the SSCC algorithm is proposed below.

5.2.2 Suffix selection with concatenation costs (SSCC)

The SSCC method is designed to use both concatenation and target costs in unit selection. The search space for suffix selection is organized as a graph where each node represents either a candidate suffix or a root morpheme as shown in Fig. 3. There is only one candidate for the root morpheme position, which is what the SSS algorithm generates. However, for the suffix positions in the utterance, there are typically many alternative paths. Viterbi algorithm is used to search the best sequence of morphemes for the utterance.[1]

Both target costs and concatenation costs are used for computing the total cost of a suffix during dynamic search. Thus, the total cost of a sequence of morphemes $\mathcal{M}$ for a given utterance is

$$\mathcal{M} = \arg\min_{\mathcal{M}} \sum_{j=1}^{J} \phi(m_j), \quad (16)$$

where J is the total number of morphemes in the utterance, m_j denotes the j^{th} morpheme, and $\phi(m_j)$ is the cost of morpheme m_j.

To compute concatenation costs $C_{j,k}$, synthetic parameters are generated first. The concatenation cost of morpheme m_j at morpheme position k, is the weighted Euclidian distance

$$C_{j,k} = \sum_{f=0}^{V} \sigma\left(f\right)\left(\delta P_{j,k}\left(f\right)\right)^T\left(\delta P_{j,k}\left(f\right)\right), \quad (17)$$

where $\delta P_{j,k} = P_j\left(f\right) - P_{k-1}\left(f_e - f\right)$. $P_{k-1}\left(f_e\right)$ represents the final frame of parameters corresponding to morpheme $k-1$. Similarly, $P_j(0)$ represents the initial frame of the candidate morpheme m_j. To make a more robust cost computation, weighted sum of parameters around the concatenation point are used where σ represents the weights. V is set experimentally.

Note that pitch is defined only for voiced speech. To generate a continuous parameter contour for pitch, linear interpolation is used between the voiced segments.

The likelihood-based target costs that are defined in Eq. (15) are initially used in the SSCC approach. In this case, because all four costs in Eq. (14) are used, four

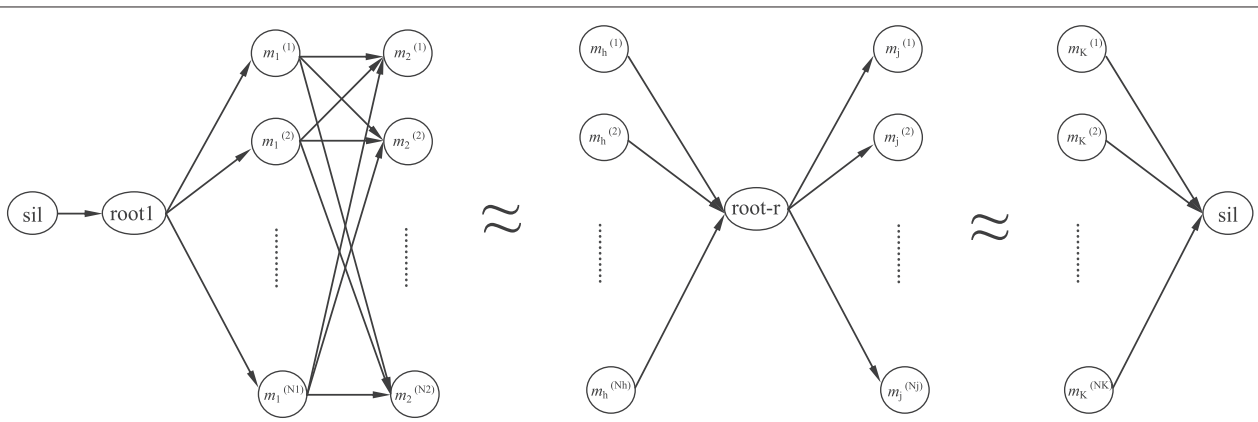

Fig. 3 An illustration of the search graph used in the Viterbi algorithm for selecting the suffixes. Root words are synthesized with the SSS system so there is only one alternate for the root words. Different numbers of natural units exist for the suffix units. More than one suffix can follow a root word

weights should be tuned for best performance. However, we have found that best performance is achieved when only the concatenation cost of the pitch parameter is used. This is related to the fact that a very limited set of suffixes were used and they were further prefiltered for best fit to the context. Such a hybrid approach removed the need for the costly weight-tuning process. Moreover, in the experiments, most of the improvement was obtained with the pitch parameter which explains the critical role of its concatenation cost.

The SSCC and SSTC algorithms are essentially different in two aspects. Even though they both implement the same objective function in Eq. 14, the SSTC algorithm only uses target costs for pitch and spectral parameters whereas the SSCC algorithm only uses the concatenation cost of the pitch parameter. Moreover, the SSTC algorithm selects units for pitch and spectral parameters independently and does time-warping to match the durations. The SSCC algorithm, however, selects one unit and uses pitch and spectral parameters from the same unit. Thus, time-warping was not needed for SSCC which avoided some of the artifacts observed in the SSTC approach. Moreover, selection based on concatenation costs eliminated the overly-smooth trajectories that were selected by the SSTC algorithm which improved the perceived speech quality. Note that the simple selection criterion of the SSCC algorithm was effective because the selection was done for specific parts of speech, frequently occurring suffixes, from a pool of prefiltered units.

6 Proposed hybrid parameter generation

Once suffix selection is done, the hybrid algorithm described in Section 2.2, can be used for parameter generation. However, the tightly-constrained approach creates discontinuities at the boundaries when a proper suffix for a context does not exist in the unit selection database. Moreover, the suffix selection algorithm can fail to select a good-fitting suffix even when it exists in the database. Discontinuity problems have been observed both with the SSCC and SSTC approaches. Therefore, here, we first propose two methods for alleviating the effects of such perceptually disturbing discontinuities. Then, energy normalization and global variance adjustment issues are addressed.

6.1 Smoothing at the transitions

The first method for addressing the discontinuity problem is to remove the constraints at the B number of initial frames and B number of final frames in a suffix. Those are the transitional frames in the suffix and removing the constraints on those enables the parameter generation algorithm smooth the discontinuities at the boundaries.

If the constraints are removed in the transitional frames, then effectiveness of the hybrid approach is reduced. To minimize that, another approach is proposed where the emission pdf parameters of the transitional states, which are the states that contain the transitional frames[2], are computed using the selected suffix instead of the HMM parameters that are available in the voice model. This approach helps more accurate modelling thanks to exploiting the selected suffix for parameter estimation. Moreover, the discontinuities are avoided since the parameter generation algorithm can smooth them.

Because state durations are typically short, only the mean parameter is estimated for state i of suffix s. To that end, suffixes are first state aligned using the SSS voice model. Then, for each transitional state i of suffix s, the mean parameter is computed as

$$\hat{\mu}_{s,i} = \frac{1}{K}\sum_{k=1}^{K} P_s^{(i)}(k), \tag{18}$$

where $P_s^{(i)}(k)$ is the k^{th} parameter vector in suffix s that is aligned with state i which is K frames long.

Because there is not enough numbers of observed frames for computing the covariance matrices, the matrices that are present in the SSS voice model are used in the proposed approach.

6.2 SSS for poorly fitting morphemes

In some cases, even after smoothing, significant discontinuities can still remain in some of the suffixes. In those cases, suffix selection is not done. Instead, the whole suffix is generated with the SSS algorithm using the parameters of the SSS model.

Detection of discontinuity was done using the L_2 norm of the difference of parameter vectors, δP, at the suffix boundaries. Unit selection is not used for a suffix if δP is above the threshold $L_{2,max}$.

The parameter $L_{2,max}$ is learned from the training data as follows. For each suffix instance in the training database, δP is computed. Then, $L_{2,max}$ is set to

$$L_{2,max} = \mu_{L_2} + 3\sigma_{L_2}, \tag{19}$$

where μ_{L_2} is the mean and σ_{L_2} is the standard deviation of the δP values computed from the training database.

Between any two root words, more than one suffix can, and typically do, exist. Therefore, decision for a current suffix should be considered in context of other decisions in the neighboring suffixes. Here, we took a brute-force approach and for all possible combinations of synthetic and natural suffixes between any two words, we decided on the combination that has the maximum number of natural segments while satisfying the $L_{2,max}$ constraint above. Because root words are always generated with SSS, only local search between the root words is enough for

Table 2 Suffix counts in the unit selection database

Total number of suffixes	1346
Total number of suffixes that have at least two phonemes	1324
Total number of suffixes that have at least 15 instances and have at least two phonemes	181

the brute-force approach. Hence, the search algorithm can be divided into smaller local searches which substantially speeds up the search algorithm.

6.3 Energy normalization

Energy of the selected suffix units typically do not match with the energy contours of the synthesized suffixes which can cause annoying energy fluctuations. To solve the issue, we first multiplied the energy feature with a scaling factor such that the average energy of the selected suffix is equal to the average energy of the synthetic suffix. However, simple amplitude scaling is usually not enough since natural speech units tend to vary more than the synthetic ones, and, even when the average energies are equal, selected suffix may sound louder. This is because the natural units can make larger peaks than synthetic ones even when the average energies are same. Therefore, a second amplitude scaling factor is used so that the ratio of peak energies $\mathrm{PE}_{syn}/\mathrm{PE}_{hyb}$ is larger than PE_{max} which is set experimentally.

6.4 Global variance adjustment

To increase the variability of SSS-based feature trajectories and reduce oversmoothing, a global variance (GV) adjustment algorithm was proposed [18]. In the GV approach, the objective function in Eq. 8 is modified with

$$\widehat{c} = \arg\max_{c} \log \left\{ p\left(Wc|\widehat{Q},\lambda\right)^{\tau} p(v(c)|\lambda_v) \right\}, \tag{20}$$

where $v(c)$ is the covariance of the static features c throughout the utterance and τ adjusts the weights between ML-based parameter generation and variance adjustment.

A two step algorithm is used for implementing GV. First, features are generated with the ML approach. Then, gradient descent algorithm is used to iteratively modify the features to maximize the objective function in Eq. 20 and increase the variance [18].

In the hybrid approach, GV algorithm cannot be used directly since it also modifies the natural segments. Instead, a modified GV algorithm is proposed here. In the first step, the algorithm described in Section 2.2 is used to generate the hybrid parameter trajectories. In the second step, the same iterative algorithm proposed in [18] without modifying the parameters in the natural units. Thus, at every iteration, after computing the new features with global variance, natural features are set back to their original values.

Note that the proposed GV algorithm may create artifacts at the boundaries of natural segments since only the synthetic segments are modified. To avoid such artifacts, τ should be high enough so that smoothness of features are preserved. However, setting τ too high can also limit the effectiveness of the GV algorithm. Here, τ was manually

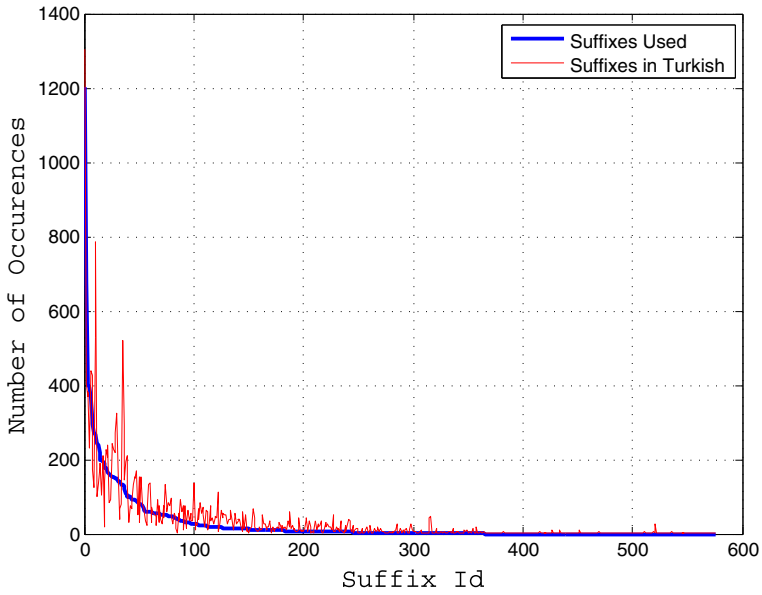

Fig. 4 Number of suffixes observed in a Turkish text database with two million words is compared with the number of suffixes observed in the training database of the proposed system. The suffix counts in the large database is scaled down for comparison purposes

tuned to $(1/N_u)$ where N_u is the total number of frames in the utterance.

7 Experiments

All systems in the experiments were trained with 30 dimensional vectors consisting of 24 line spectral frequencies (LSFs), 1 log F0 coefficient, and 5 voicing strength parameters. Voicing strengths were computed using normalized autocorrelation measure for 5 evenly spaced spectral bands between 0 and 8 kHz. Recordings were done at 44.1 kHz sampling rate. Speech signal was amplitude-normalized and downsampled to 16 kHz before training. EHMM labeling tool was used to align the phonemes with the audio files before training. The HTS toolkit was used in training and synthesis[3]. Global variance and mixed-excitation were used in addition to postfiltering to improve the speech quality.

Speaker-dependent SSS model was generated using 2300 utterances that were recorded by a female speaker. Total duration of the recorded speech is approximately 190 min. The speaker is a professional actress speaking with Istanbul accent. Recording was done in a professional studio environment with a high-quality condenser microphone.

Turkish has one-to-one relationship between its graphemes and phonemes in most cases. However, Turkish sounds have nuances and there are exceptions to the rules. Therefore, classification and regression trees (CART) were used to model grapheme-to-phoneme mappings of Turkish. A pronunciation lexicon [19] was used to train the CART tree.

Turkish stress markers typically follow a limited number of rules. Those linguistic rules [19] were used in the system for marking primary and secondary stress.

After the baseline system was built, several issues were noted such as discontinuities during vowel transitions in diphthongs and glide-vowel transitions. Moreover, there were annoying clicking sounds that randomly pop up in the middle of some of the samples. Those problems were found to be related to the errors in the automatic alignment process. Short silences between words or some of the silences at the beginning or end of the utterances were sometimes appended to the phonemes during automatic alignment. Once those misalignment problems in the training database were fixed manually, annoying clicking sounds and discontinuities disappeared.

Table 3 Parameters of the SSTC and SSCC algorithms

R_d (SSTC)	30
F (SSCC)	2
w (SSCC)	[1 0.5 0.3]
PE_{max} (SSCC)	0.8
B (SSCC)	7
ζ_d	1.5

Table 4 MOS test results of the Turkish statistical speech synthesis system

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

Suffix database was created using the same training database. Thus, no additional data collection or manual annotation was done for the hybrid approach. Each suffix type was required to contain at least 15 instances in the database before it could be used in the unit selection database. Moreover, suffixes in the database were required to contain at least two phonemes because short suffixes that contain only one phoneme occasionally caused discontinuous contours. Total number of suffixes in the database is shown in Table 2.

Suffixes used in this work were obtained from a small database as discussed above. In order to check if the suffix distribution is representative of Turkish text, suffixes were extracted from a large Turkish text database that contains two million words from newspapers, eBooks, and Wikipedia. Then, distribution of the suffixes in the large text was compared with the distribution in the small database used for training here. Comparison shown in Fig. 4 indicates that the small database roughly follows the same pattern observed in the large database.

Experiments were performed in three phases. In the first phase, performance of the ML-based parameter generation was assessed. In the second phase, SSCC approach was tested and compared with the baseline system. In the third phase, the SSTC approach was tested and compared

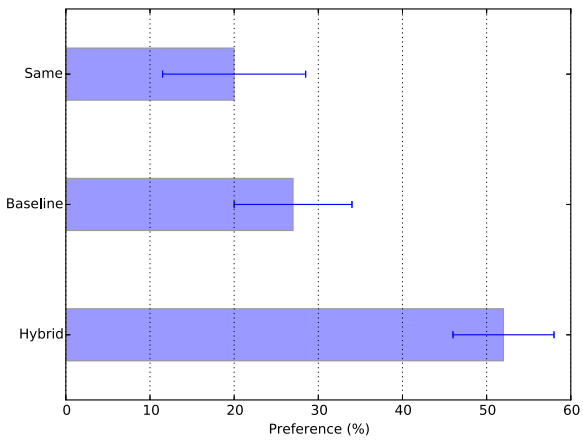

Fig. 5 AB preference test results for the hybrid SSCC algorithm where only the pitch feature is synthesized with the hybrid method. LSF parameters are same in the baseline (SSS) and hybrid systems. The 95 % confidence intervals are also shown

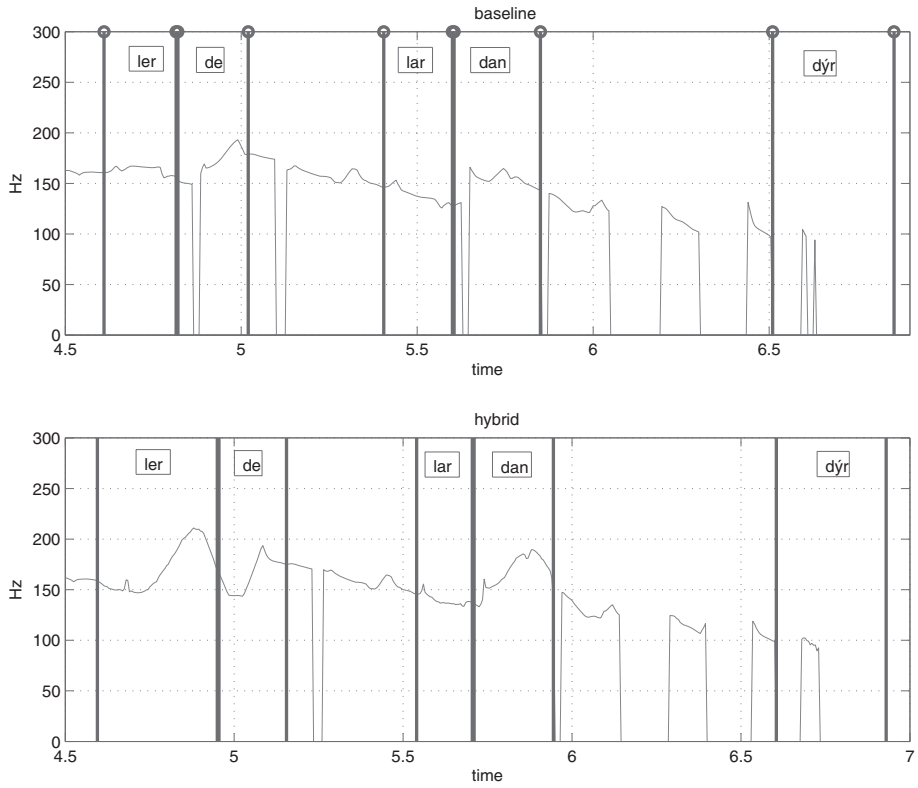

Fig. 6 Comparison of pitch contours for the hybrid SSCC method and baseline systems. Suffixes and their boundaries are indicated in the figure

with the SSS and SSCC systems. Parameters of the SSTC and SSCC systems are shown in Table 3.

7.1 Statistical speech synthesis performance

Mean opinion score (MOS) is used to test the quality of the SSS system. Eight male and eight female listeners took the listening tests. All of the listeners were native speakers of Turkish.

For calibration purposes, subjects were presented two samples for each of the five MOS scores before they took the tests. Subjects were asked to score speech samples based on how natural they sounded. Twelve test sentences were selected from news domain and 18 sentences were selected from novel domain. Results are shown in Table 4.

7.2 Performance of the hybrid system with SSCC

AB quality preference tests were done to compare the performance of the hybrid SSCC algorithm with the statistical synthesis system. Tests were conducted in two parts. In the first part, hybrid pitch features were used with the baseline SSS-generated LSF features. In the second part, both pitch and LSF features were generated with the hybrid algorithm to assess the additional improvement

Table 5 Variance of the logarithm of pitch for the baseline and hybrid systems

Baseline system (SSS)	0.035
Hybrid system (SSCC)	0.042
Hybrid system (SSTC)	0.038

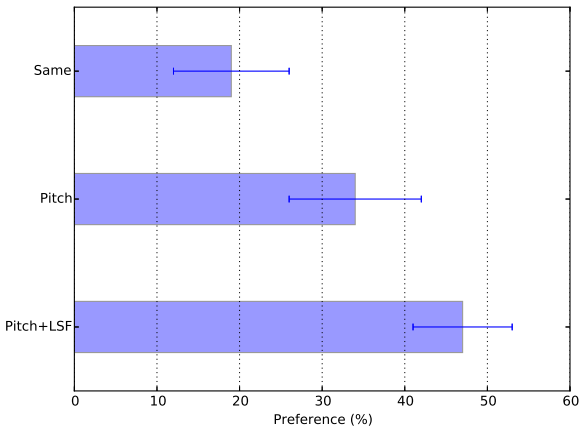

Fig. 7 AB preference test results for the hybrid SSCC algorithm. In one system (Pitch), pitch is generated with hybrid SSCC approach and LSFs are generated with the SSS approach. In the second system (Pitch+LSF), both LSF and pitch are generated with the hybrid SSCC method. 95 % confidence intervals are also shown

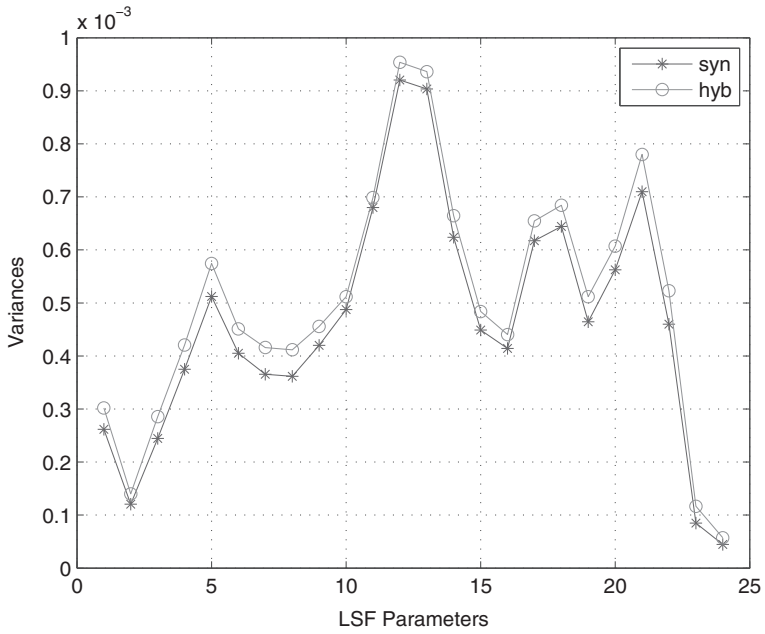

Fig. 8 Comparison of variances for the LSF parameters generated with the hybrid SSCC method and baseline (SSS) systems. Variance is higher for the hybrid method for all 24 LSF parameters

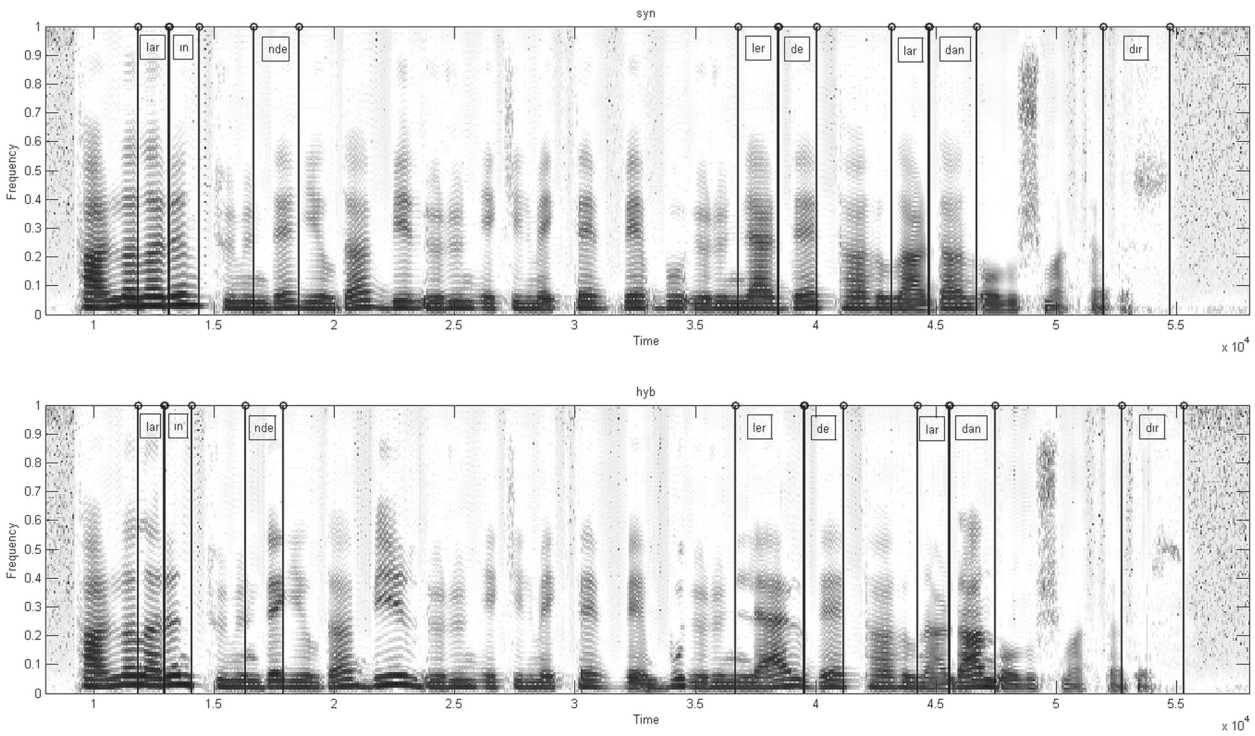

Fig. 9 Comparison of spectrograms for the hybrid SSCC method and baseline (SSS) systems. Suffixes and their boundaries are indicated in the figure

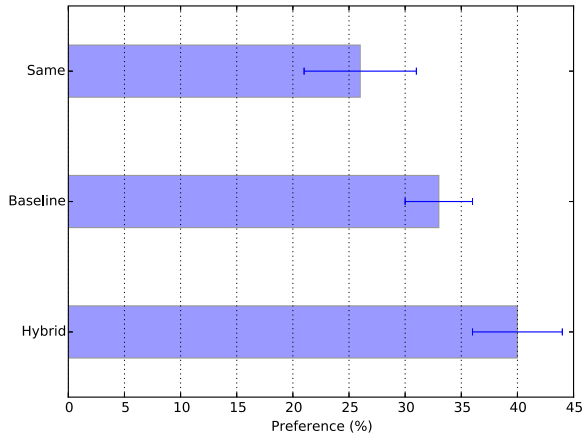

Fig. 10 AB preference test results for the hybrid SSTC algorithm and the baseline systems. The 95 % confidence intervals are also shown

with the LSF features. Thirty sentences were used and ten listeners took the tests.

Results of the AB test with hybrid pitch features are shown in Fig. 5. The hybrid system significantly outperformed the statistical system in the AB tests. Analysis of speech samples revealed that the improvement of perceived speech quality was related to improved intonation patterns. Not only pitch variability increased in the hybrid approach but also stress was more audible and clear. Figure 6 shows a comparison of the pitch contour for an utterance with the hybrid SSCC approach versus the SSS approach. In Fig. 6, difference in pitch changes are clearly visible for the /ler/ and /dan/ suffixes both of which are stressed in the utterance.

To assess the overall improvement in pitch variability, variance of the log-f0 feature is computed for the 30 test utterances. Average of the variances is then computed. Comparison of the average variance for the statistical system and hybrid system are shown in Table 5. The hybrid SSCC system has significantly higher variance compared to the statistical system.

Samples where listeners preferred the SSS system compared to the hybrid system were also analyzed. In many of those samples, at least one of the selected stressed suffixes is longer than the synthetic suffix that it replaces. In those cases, pitch contour of the selected suffix is time-warped which sometimes caused artifacts in pitch contours.

In the second phase of hybrid SSCC tests, both pitch and LSF features were generated with the hybrid SSCC approach and compared with the case where only the pitch feature is generated with the hybrid SSCC approach. Results of the AB test are shown in Fig. 7. Using hybrid LSF contours in addition to pitch improves the average perceived quality compared to the hybrid pitch-only case.

When both LSF and pitch contours are generated with hybrid SSCC, time-warping was not required, which resolved the pitch artifacts related to time-warping. That increased the speaker preference for the hybrid pitch+LSF system. On average eight listeners had higher preference for the hybrid pitch+LSF system and two speakers had preference for the pitch-only system.

Similar to the pitch-only case, we analyzed the samples where the listeners preferred the hybrid pitch-only case compared to the hybrid pitch+LSF features. We have found that while variations in LSF features were perceived as natural speech variability and preferred by some of the

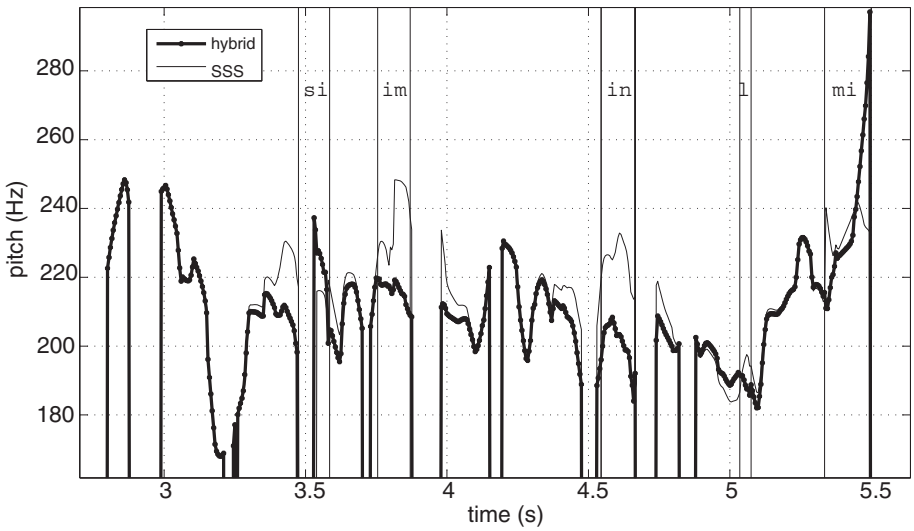

Fig. 11 Comparison of pitch contours for the baseline and hybrid systems. Borders of the five suffixes occurring in the utterances are shown. The final suffix /mi/ indicates a question. Sudden pitch rise that is expected at the end of the question utterance is better modelled with the SSTC-based hybrid system

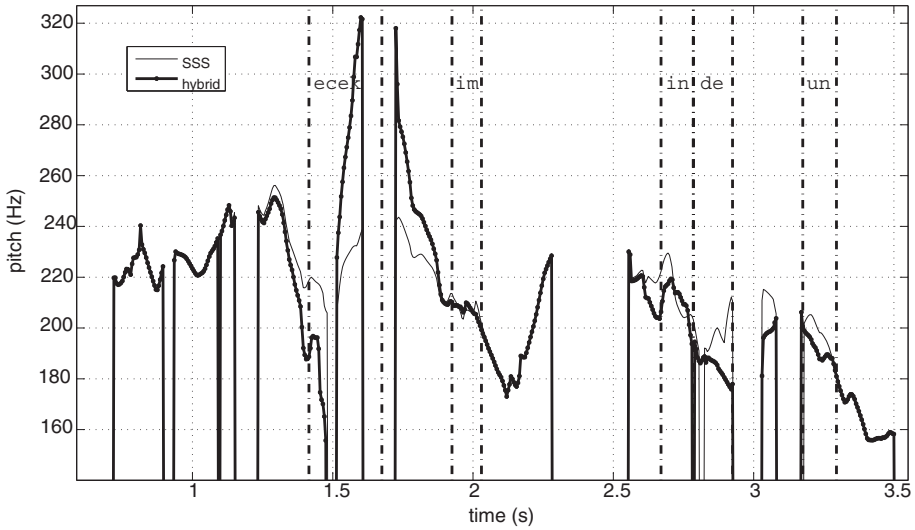

Fig. 12 Comparison of pitch contours for the baseline and hybrid SSTC systems. Borders of the five suffixes occurring in the utterances are shown. Sudden pitch variation in the suffix is modeled better with the SSTC-based hybrid system. Synthetic speech with the hybrid pitch contour was perceived as more natural by the listeners

listeners, some others perceived them as artifacts. This difference in perception resulted in a preference for the pitch-only case in some of the test samples.

Variance of the LSF features are compared with the statistical system in Fig. 8. LSF variability, therefore the formant variability, improves with the hybrid approach as expected. An example is shown in Fig. 9 where the spectrograms of the baseline system and hybrid system are compared. For example, improvement in the formant trajectories can be observed in the suffix /ler/ in Fig. 9. Those formant fluctuations were mostly perceived as natural variations in speech by the listeners.

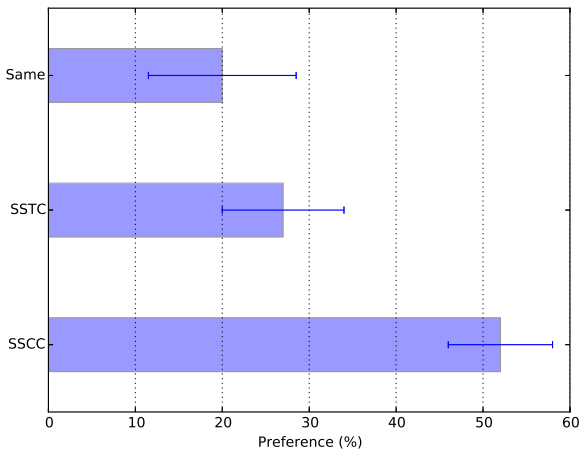

Fig. 13 AB preference test results for comparing the hybrid SSTC and SSCC algorithms. The 95 % confidence intervals are also shown

7.3 Performance of the hybrid system with SSTC

To assess the quality improvement with the hybrid SSTC approach, AB preference test was performed. Results are shown in Fig. 10 with 95 % confidence intervals. Even though the SSTC algorithm improved the performance slightly, the improvement is not as large as what was obtained with the SSCC algorithm. Still, it was found to be statistically significant using the Student t test.

Test samples and listener preferences were analyzed to explore the major factors behind the test results. One of the factors was found to be improvement in question sentences. In Turkish, question sentences typically have special suffixes, such as /mi/, /midir/, at the end of the verbs. Those question suffixes are usually stressed. In some significant number of cases with the SSS system, question suffixes were over-smoothed which hurt the listener preference. Most of those issues were resolved since stress patterns of the question suffixes were captured better by the hybrid system. An example case is shown in Fig. 11 where the hybrid system modelled the pitch rise better at the end of a question utterance.

A second factor behind improved quality was the improvement in the /de/, or /da/, suffix which means "also" in English. They are written as if they are independent words while they are pronounced as a suffix of the word that they come after. Those suffixes are very commonly used in Turkish and using correct prosody for them is important to convey the correct semantic message. The SSTC system generated more natural pitch variation for those suffixes.

Besides the two specific suffixes discussed above, the SSTC system improved the intonation contours in general.

That improvement made the most difference in the improved perceptual quality based on listener feedback. Another example to pitch contour improvement with the hybrid system is shown in Fig. 12. Pitch variation is significantly higher in the hybrid system compared to baseline system in Fig. 12.

The average log-f0 variance for the SSTC hybrid system is shown in Table 5. The log-f0 parameter has significantly higher variance with the SSTC system compared to the SSS system. However, its variance not as large as the SSCC system as expected since the SSTC algorithm favors smooth intonation patterns in suffix selection.

In synthesis, severe and frequent discontinuities were observed for the LSF features since the concatenation cost was not taken into account during suffix selection. To minimize the discontinuities, the smoothing algorithm described in Section 5.2.2 was used for all frames. However, in that 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 in the SSTC approach.

The SSTC system was also compared with the SSCC system using AB tests and results are shown in Fig. 13. The SSCC system significantly outperformed the SSTC system. Higher pitch variability with the SSCC algorithm had a significant effect in listener preference. Moreover, additional improvement with the LSF features using the SSCC algorithm was not feasible using the SSTC algorithm and that also affected the listener preference.

8 Conclusions

A hybrid statistical/unit selection speech synthesis system is proposed that significantly improved the quality of a Turkish SSS system. As opposed to other hybrid techniques, the technique here does not require the costly and time-consuming process of unit selection system development. Similarly, no additional speech data was collected or annotated for the unit selection system. Even though the idea is applied to Turkish, it could be used for other agglutinative languages such as Finnish and Estonian.

Suffixes were used as the fundamental units in selection. Two suffix selection algorithms are proposed. The SSTC approach is based on maximum-likelihood based target cost calculation and it generated overly smooth trajectories in many cases which reduced its performance. The second algorithm, SSCC, is based on suffix selection using the concatenation cost of the pitch parameter. The SSCC algorithm improved the intonation better than the the SSTC algorithm. Moreover, LSF trajectories of the suffixes selected with the SSCC approach fit better in the suffix contexts and required less smoothing than the suffixes selected with the SSTC approach which helped further improve the quality. The substantial improvement in both pitch and LSF contours with the SSCC approach are verified by subjective listening tests. Most of the improvement was found to be related to improvements in the perceived stress and prosody.

Endnotes

[1]Note that the term morpheme includes both roots and suffixes.

[2]A state is transitional even if part of it contains non-transitional frames.

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

Competing interests
The authors declare that they have no competing interests.

Acknowledgments
This work has been supported by TUBITAK 3501 program under Grant 109E281.

Received: 22 July 2015 Accepted: 22 January 2016
Published online: 02 February 2016

References

1. H Zen, K Tokuda, AW Black, Review: statistical parametric speech synthesis. Speech Commun. **51**(11), 1039–1064 (2009)
2. AW Black, H Zen, K Tokuda, in *Proc. ICASSP*. Statistical parametric speech synthesis, vol. 4 (IEEE, Honolulu, Hawaii, USA, 2007), pp. 1229–1232
3. L-H Chen, C-Y Yang, Z-H Ling, Y Jiang, L-R Dai, Y Hu, R-H Wang, in *Blizzard Challenge Workshop*. The USTC System for Blizzard Challenge 2011 (ISCA, Turin, Italy, 2011)
4. S Rouibia, O Rosec, in *INTERSPEECH*. Unit selection for speech synthesis based on a new acoustic target cost (ISCA, Lisbon, Portugal, 2005), pp. 2565–2568
5. S Pan, M Zhang, J Tao, in *INTERSPEECH*. A Novel Hybrid Approach for Mandarin Speech Synthesis (ISCA, Makuhari, Chiba, Japan, 2010), pp. 182–185
6. S Tiomkin, D Malah, S Shechtman, Z Kons, A hybrid text-to-speech system that combines concatenative and statistical synthesis units. IEEE Trans. Audio Speech Lang. Process. **19**(5), 1278–1288 (2011)
7. V Pollet, A Breen, in *INTERSPEECH*. Synthesis by generation and concatenation of multiform segments (ISCA, Brisbane, Australia, 2008), pp. 1825–1828
8. M Plumpe, A Acero, HW Hon, X Huang, in *INTERSPEECH*. HMM-based Smoothing for Concatenative Speech Synthesis (ISCA, Lisbon, Portugal, 1998)
9. T Raitio, A Suni, H Pulakka, M Vainio, P Alku, in *ICASSP*. Utilizing glottal source pulse library for generating improved excitation signal for hmm-based speech synthesis (IEEE, Prague, Czech Republic, 2011), pp. 4564–4567
10. H Kawai, T Toda, J Ni, M Tsuzaki, K Tokuda, in *Fifth ISCA Workshop on Speech Synthesis*. XIMERA: A New TTS from ATR Based on Corpus-based Technologies (ISCA, Pittsburgh, PA, USA, 2004)
11. I Sain, D Erro, E Navas, J Adell, A Bonafonte, in *Proceedings of the Blizzard Challenge*. Buceador hybrid tts for blizzard challenge (ISCA, Turin, Italy, 2011)
12. X Gonzalvo, A Gutkin, J Claudi Socoro, I Iriondo, P Taylor, in *INTERSPEECH*. Local minimum generation error criterion for hybrid HMM speech synthesis (ISCA, Brighton, United Kingdom, 2009), pp. 404–407
13. A Pradhan, A Prakash, S Aswin Shanmugam, GR Kasthuri, R Krishnan, HA Murthy, in *Communications (NCC), 2015 Twenty First National Conference On*. Building speech synthesis systems for indian languages (IEEE, Mumbai, India, 2015), pp. 1–6
14. R Zhang, Z Wen, J Tao, Y Li, B Liu, X Lou, in *INTERSPEECH*. A hierarchical viterbi algorithm for mandarin hybrid speech synthesis system (ISCA, Singapore, 2014), pp. 795–799. 894
15. S Tiomkin, D Malah, S Shechtman, Z Kons, A hybrid text-to-speech system that combines concatenative and statistical synthesis units. IEEE Trans. Audio Speech Lang Process. **19**(5), 1278–1288 (2011)

16. K Oflazer, S Inkelas, in *Proceedings of the EACLWorkshop on Finite State Methods in NLP*. A Finite State Pronunciation Lexicon for Turkish, vol. 82 (Association for Computational Linguistics, Budapest, Hungary, 2003), pp. 900–918
17. D Yuret, F Ture, in *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. Learning Morphological Disambiguation Rules for Turkish, (2006), pp. 328–334. Association for Computational Linguistics
18. T Toda, K Tokuda, A speech parameter generation algorithm considering global variance for hmm-based speech synthesis. IEICE - Trans. Inf. Syst. **E90-D**(5), 816–824 (2007)
19. İ Ergenç, *Spoken Language and Dictionary of Turkish Articulation*. Multilingual. Yabancı dil yayınları. (Multilingual, Istanbul, 2002)