Phonetically conditioned prosody generation for TTS: An unsupervised phonetic-to-prosodic mapping framework

D. N. Krishna, M. G. Khanum Noor Fathima, Mythri Thippareddy, A. Sricharan, V. Ramasubramanian†
PES Institute of Technology - Bangalore South Campus (PESIT-BSC)
Bangalore, 560100, India
†v.ramasubramanian@pes.edu

***Abstract*—We propose a framework for phonetically-conditioned prosody generation for TTS which has two phases: first, in an off-line phase, it learns the phonetic-to-prosodic mapping underlying a target prosody database in an unsupervised manner using a 'segmental K-means' algorithm; secondly, for a given input text, it uses a segmental unit-selection framework to select appropriate prosodic-profiles from the learnt mapping, conditioned on the input phonetic sequence, which are then transferred onto a conventional TTS output unit-sequence. This framework is derived from a 2-stage unit-selection framework we had proposed recently, which employs 'phonetic conditioning', wherein target prosody-profiles are identified from a given target prosody database, conditioned on their underlying phonetic content with varying degrees of matching and over variable length time-scales that are optimized to be as long as possible. In this paper, based on an equivalence result shown by us earlier in ultra-low bit-rate speech coding, we show how the unit-selection framework can be approximated by a segmental unit-selection framework, with advantages of longer phonetic conditioning and very low complexity. We characterize the performance of the proposed framework using various objective measures, compare it with the earlier unit-selection framework and show its practical viability.**

Index Terms: Prosodic-to-phonetic mapping, unsupervised learning, segmental unit-selection, prosody generation, prosody transplantation, TTS, phonetic conditioning.

I. Introduction

The problem of general expressive prosody, with a certain extent of control on the degree and style of prosody still remains a hard problem with high academic value and importance [1], [2]. The most direct approach to incorporating prosody in a TTS system for a given input text is to simply 'predict' the prosody from the input text. This has attracted considerable attention (see for example a review and recent approaches in [3]) and has solutions ranging from rule-based approaches (that use syntactic and semantic structures and information in the text to arrive at an appropriate prosody prediction) to data-driven machine-learning methods such as CART or neural-network based functional mappings.

Recently, we proposed a framework for transplantation of a 'target prosody' onto speech synthesized by a conventional unit-selection synthesis system (the 1st stage), using a 2nd stage 'prosodic database' which has speech in the desired prosody, and which is searched by a 2nd-stage unit-selection for appropriate units that match the output of the 1st stage unit-selection, so that, subject to this 'phonetic conditioning' (i.e. matching of the units output by the 1st stage and units selected from the prosodic-database to varying degrees depending on the efficacy of the 2nd stage unit-selection), the prosody associated with the selected units will have the desired target prosody which can then be transplanted on to the units derived from the 1st stage output [14], [15]. In essence, the higher the degree of match (in terms of exact phonetic match) and longer the contiguity of the match, the prosody associated with the resultant units from the prosodic-database, will be the desired prosody in the sense that 'it is the same text (i.e., phonetic sequence) as the input, but spoken with the prosody-style of the target prosodic-database', thus serving the objective of transplantation on to the input units to realize the target prosody.

In this paper, we propose a framework for prosody generation for an input text, by a two-phase approach, which are outlined as follows.

The first phase, termed here as 'Phonetic-to-prosodic mapping' is an off-line learning of a mapping from the given target prosodic-database, which essentially yields a restructuring of the prosody database, in such a way that the mapping represents how a phone-string (e.g. a word or phrase in the input text) has a very typical prosody associated with it. This mapping (or restructuring) is illustrated schematically in Fig. 1, which can be viewed as being made of two codebooks, namely, the primary codebook (or the 'phone-string' codebook) made of variable length phone-strings (that approximate all possible input phone strings in the text) and the other, the secondary codebook (or the 'prosody-profile' codebook) made of a cluster of prosodic-profiles for each phone-string in the first codebook, where a 'profile' is a speech segment from the prosody-database, represented in terms of spectral feature sequence, pitch-contour, gain-contour and associated duration. This phonetic-to-prosodic mapping is learnt in the form of these two codebooks (the 'phone-string' codebook and the 'prosody profile' codebook) using the unsupervised clustering 'segmental K-means' (SKM) algorithm which uses the 'modified K-means' (MKM) algorithm within it (the MKM algorithm is used to create one or more phone-string pseudo-centroid representatives of a given phone-cluster within the

978-1-5090-3646-2/16/$31.00 ©2016 IEEE

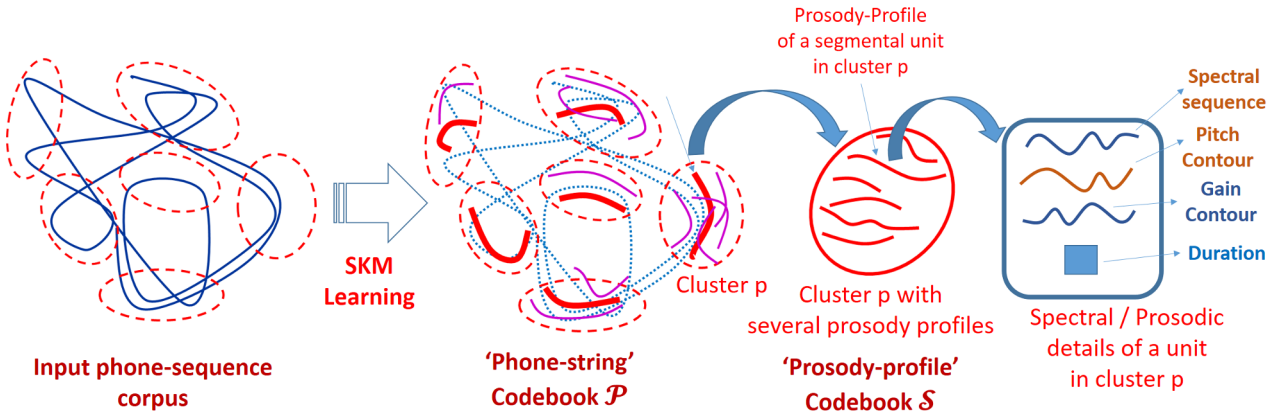

Fig. 1. *Schematic of proposed phonetic-to-prosodic mapping as learnt by a segmental K-means (SKM) algorithm from a desired target prosody database*

SKM iterations, as well as to create a specified number of cluster representatives that make up the secondary or 'prosody profile' codebooks after convergence of the SKM algorithm).

In the second phase, we employ a 'Segmental unit-seletion', which in itself has two steps - the first, termed 'Segment decoding', which decodes the input phonetic unit sequence into fragments of variable-length phone strings using the 'phone-string' codebook, and the second step, termed 'Prosody-profile selection', selects optimal speech segments from the 'prosody-profile' codebook, conditioned on the first step phone-string fragments, and minimizing various spectral and prosodic join-costs between the prosody-profiles. The prosody associated with the selected speech segments are further transplanted onto the unit-sequence of the conventional TTS output.

The segmental unit-selection framework proposed here offers specific advantages over the 'phone-level' unit-selection (described earlier above as our earlier work [14], [15]) in terms of the naturalness of the selected prosodic units and lower complexity. We derive this segmental version, along the lines of two unit-selection paradigms (adapted from concatenative TTS principles) that formed the basis of low bit-rate speech coding (specifically, low rate spectral quantization) in [4] and [5]. While [4] is a 'frame-level' unit-selection akin to the 'phone-level' unit-selection we proposed in [15] (and outlined in Sec. II), [5] is a segmental version of the unit-selection based on the principle of restructuting the unit-database into segmental units, and indexed and accessed via a pre-quantization segmental codebook. Such a realization offers significantly low complexity when compared to a full-scale unit-selection along with the intrinsic advantage of retaining within-unit co-articulations in segmental units. These two paradigms have been further generalized into unified frameworks and their relative optimality analyzed and studied in detail in the context of ultra low bit-rate speech coding [6], [7], [8], [9], [10], [11], [12]. We draw from these work in speech coding, in proposing here the phonetically-conditioned 'segmental unit-selection' for the prosody generation framework considered here and compare it with the 'phone-level' unit-selection we proposed earlier [15], and recently [16].

II. Phone-level unit-selection

Here, we outline the 'phone-level' unit-selection proposed earlier by us [14], [15], and which forms the basis for the 'segmental' version we propose in this paper. A schematic of this framework is shown in Fig. 2. In this framework, the 1st-stage unit-selection is the conventional part of a concatenative TTS (e.g. a mixed-Viterbi formalism proposed by us recently [13]). It is driven by the output of the G2P which generates the target unit sequence vectors from the given input text. The 1st-stage unit-selection then selects appropriate units from the 1st-stage unit-database, which can be used to drive a concatenative synthesizer such as LPC, PSOLA or MBROLA. However, the system does not have a synthesizer in the 1st-stage. Instead, the spectral component of the 1st-stage units selected from the 1st-stage unit-database are sent directly to a synthesizer at the 2nd-stage, for a subsequent combination with the prosodic information derived from the 2nd-stage unit-selection. The 2nd-stage unit-database carries the desired prosody and could be spoken by the same or different speaker as the 1st-stage unit-database. For instance, the 2nd-stage unit-database could be in different styles as desired in the synthesized speech of the overall system, such as, news-reading style, story-telling styles, or various emotional styles etc. It is important to note that the 1st-stage unit-database is of a typical size as is required for high quality synthesis, while the 2nd-stage unit-selection could be smaller, being able to adequately provide the prosodic-stylistic aspects of a given unit-sequence.

In the 'phone-level' unit-selection, the 2nd stage prosody transplantation in Fig. 2 takes in as input the unit-sequence output of the 1st-stage unit-selection given by the phonetic units $O = (\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_t, \ldots, \mathbf{o}_T)$. Each of these could be vectors with components indicating the primary phonetic unit label and any other contextual features that characterize the unit selected from the 1st-stage unit-database, and that is considered relevant for the 2nd-stage prosody-selection. Note that a vector $\mathbf{o}_t$ in this sequence is derived from the 1st-stage unit-database, and not from the input text, and hence can afford to have features derived by signal-processing on the 1st-stage unit-database.

Let the 2nd-stage unit-database be represented as a sequence of N units, $\mathcal{U} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_i, \ldots, \mathbf{u}_N)$ with durations

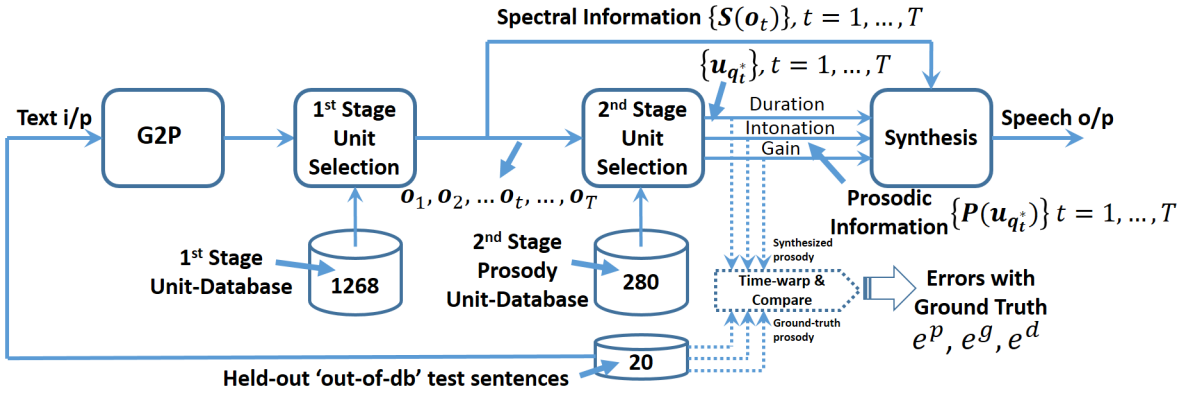

Fig. 2. *Schematic of proposed 2-stage unit-selection framework for prosody transplantation*

of the associated acoustic segments as $l_1, l_2, \ldots, l_i, \ldots, l_N$. For a given input unit-sequence $O = \mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_t, \ldots, \mathbf{o}_T)$, the 2^{nd} stage 'phone-level' unit-selection finds the optimal unit sequence $U^* = (\mathbf{u}_{q_1^*}, \mathbf{u}_{q_2^*}, \ldots, \mathbf{u}_{q_t^*}, \ldots, \mathbf{u}_{q_T^*})$ where $\mathbf{u}_{q_t^*} \in \mathcal{U}$ by determining the optimal unit indices $Q^* = (q_1^*, q_2^*, \ldots, q_t^*, \ldots, q_T^*)$ that minimizes the distortion between O and any U (from the 2nd stage unit-database) as given by,

$$Q^* = \arg\min_{Q} (1-\alpha) \sum_{t=1}^{T} D_u(\mathbf{o}_t, \mathbf{u}_{q_t}) + \alpha \sum_{t=2}^{T} D_c(q_{t-1}, q_t) \quad (1)$$

Here, $D_c(q_{t-1}, q_t)$ is the concatenation costs that takes on a value of 0 if $u_{q_{t-1}}$ and u_{q_t} are contiguous in the 2nd-stage unit database, and takes on a value of the Euclidean distortion between the last frame of the acoustic segment annotated as $u_{q_{t-1}}$ and the first frame of the acoustic segment annotated as u_{q_t}, if they are not contiguous. In [14], [15], we used a grouped-Viterbi realization for solving Q^* and these details are not reproduced here.

III. Proposed Segmental Unit-Selection Framework

Following the equivalence between the 'frame' level unit-selection [4] and 'segmental' unit-selection [5] for spectral quantization for low bit-rate speech coding, we adapt and propose here a 'segmental' version of the 'phone-level' unit-selection in the previous section. Fig. 3 shows the expansion of the '2nd stage unit-selection' block of 2 corresponding to the proposed 'segmental unit-selection' being proposed here. This is a 2-step procedure, with a 'segment-decoding' (using the 'phone-string' codebook $\mathcal{P}$), followed by an optimal 'prosody-profile selection' using a segment-Viterbi algorithm (on the 'prosody-profile codebook' $\mathcal{S}$), which together constitute the overall 'segmental unit-selection'.

The objective of this system is to take the input unit sequence $O = (\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_t, \ldots, \mathbf{o}_T)$ and yield a sequence of K indices $(\phi_1^*, \phi_2^*, \ldots, \phi_k^*, \ldots, \phi_K^*)$ which correspond to some 'segmental' units in the original unit database $\mathcal{U}$ in such a way that these units are the best approximation of the input unit sequence O, in terms of unit-cost and concatenation-costs, as will be outlined further. The prosody-profiles associated with these output 'segmental' units $(\phi_1^*, \phi_2^*, \ldots, \phi_k^*, \ldots, \phi_K^*)$ are further transplanted on to the corresponding phone-level units $O = \mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_t, \ldots, \mathbf{o}_T)$ through appropriate time-scale and pitch-scale modification to yield synthesized speech that should have the prosody style as the target style of the prosody database.

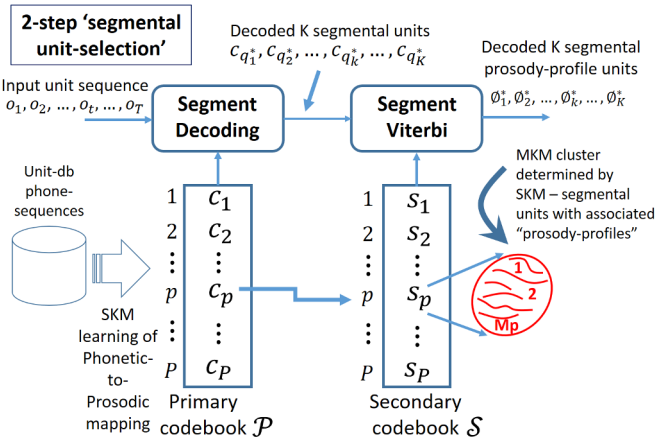

Fig. 3. *Schematic of proposed prosody generation framework*

In the following, we describe the two phases that go into this system outlined above (Fig. 3 - one, which is the off-line learning of the 'phonetic-to-prosody' mapping in the form of the two codebooks, the 'phone-string' codebook $\mathcal{P}$ and the 'prosody-profile codebook' $\mathcal{S}$ (see Fig. 1, and the other, the 'segmental unit-selection' that uses these two codebooks, representing the 'phonetic-to-prosodic mapping' to generate the desired prosody for an input text.

A. Phonetic-to-prosody mapping learning

We now describe the main components of this 'segmental unit-selection' system in Fig. 3, namely, the 'phone-string' codebook $\mathcal{P}$ and the 'prosody-profile codebook' $\mathcal{S}$ (which together represent the 'phonetic-to-prosody' mapping and which constitute a 'restructuring' of the prosody database $\mathcal{U}$ so as to enable 'segmental unit-selection' based prosody-selection), and how they are learnt using the unsupervised clustering 'segmental K-means' algorithm.

The 'phone-string' codebook $\mathcal{P} = (c_1, c_2, \ldots, c_p, \ldots, c_P)$ is a 'variable-length phone-string codebook' of size P, with each entry c_p being a variable-length phone string of length l_p (thick red-line in Fig. 1) in terms of the monophone units

that define any $\mathbf{o}_t$ in O, i.e., c_p is a sequence of l_p monophone units. This 'phone-string' codebook $\mathcal{P}$ is designed in the same way as a variable-length segment quantizer (VLSQ) (used in segment quantization) is trained, using a segmental K-means (SKM) algorithm [17], [18] from a large training data $\{O\}$. $\mathcal{P}$ can be specified by $(P, \bar{l})$, where $\bar{l}$ is the average length of the phone-strings in the codebook.

The unsupervised learning SKM algorithm performs two steps iteratively until convergence: i) segmentation of each input phone sequence into variable length segments using a current codebook $\mathcal{P} = (c_1, c_2, \ldots, c_p, \ldots, c_P)$, , and ii) clustering of all segments in the decoded output with a specific label p, i.e., decoded as belonging to codebook entry c_p (a variable length phone string), and updating the entry c_p by the pseudo-centroid of the cluster (a new variable length phone string), until convergence, which is determined as plateuing of the decoding score in step (i). The above iterative process is initialized either by starting with step (i) with an initial codebook $\mathcal{P}$ with arbitrary segments (phone-strings of a desired average length from $\mathcal{U}$) or by starting with step (ii) using an initial cluster derived by a flat-start segmentation of each phone-unit sequence of the training corpus (with the flat-start number of segments controlled based on a desired average length of the phone-strings in $\mathcal{P}$.

Each entry c_p points to a corresponding entry S_p in the 'prosody-profile' codebook $\mathcal{S} = S_1, S_2, \ldots, S_p, \ldots, S_P$, where S_p is a collection of all 'segmental' units in $\mathcal{U}$ which are mapped to the same cluster corresponding to c_p during the SKM training of $\mathcal{P}$ (thin dashed red-line oval in Fig. 1) , i.e., $S_p = \{s_k \in \mathcal{U} : L(s_k) = c_p\}$, where $L(\cdot)$ indicates the label assigned to a segment s_k by the SKM procedure at the end of its iteration and convergence. A segmental unit $s_k \in S_p$ is characterized by the following property: i) it is a segmental unit - made of a sequence of phonetic units, ii) its underlying phonetic sequence is very similar to c_p since this segmental unit is part of the cluster of segmental units in the database determined by the SKM algorithm to belong to the phone-string c_p and, iii) it is also associated with speech signal (or its parameterized representation), and hence has a sequence of spectral vectors, and prosody-profiles, namely, pitch contour, gain contour and duration associated with it (right most detail in Fig. 1). The phonetic-data is relevant for defining the unit-cost and the acoustic-data (spectral and prosody-profiles) is relevant for the join-cost in the 'segment Viterbi' stage to appropriately select the final sequence of optimal units that approximate the given input sequence (as will be outlined along with Eqn. (4).

B. Segmental unit-selection for prosody generation

As shown in Fig. 3, the overall segmental unit-selection takes place in two passes - the first being the 'segment decoding' of the input unit sequence O into a sequence of K entries from $\mathcal{P}$ followed by a 'prosody-profile selection' using a 'segment Viterbi' (which is a forced-alignment Viterbi) through the groups in $\mathcal{S}$ corresponding to the K unit sequence obtained from the 'segment decoding'. These are detailed as follows.

1) Segment decoding: The 'segment decoding' segments the input unit sequence $O = (\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_t, \ldots, \mathbf{o}_T)$ into K segments $w_1, w_2, \ldots, w_k, \ldots, w_K$ defined by the segment boundaries $B = ((b_0 = 0), b_1, b_2, \ldots, b_{k-1}, b_k, \ldots, (b_K = T))$, where the partial phone-sequence $w_k = (o_{b_{k-1}+1}, \ldots, o_{b_k})$ and each w_k is labeled by a phone-string entry c_{q_k} in $\mathcal{P}$, i.e., in the process O is decoded into a sequence of phone-string indices $(c_{q_1}, c_{q_2}, \ldots, c_{q_k}, \ldots, c_{q_K})$, in a minimum distortion sense, given by

$$D(O, Q) = \min_{K,B,Q} \sum_{k=1}^{K} d_{seg}(w_k, c_{q_k}) \tag{2}$$

where $d_{seg}(w_k, c_{q_k})$ is a string-edit distance type of function that optimally warps c_{q_k} to w_k to yield the minimum alignment cost. While (K, B, Q) is any such arbitrary segmentaion and labeling of O in terms of the entries of $\mathcal{P}$, the optimal decoding is given by,

$$(K^*, B^*, Q^*) = \arg \min_{K,B,Q} D(O, Q) \tag{3}$$

with the corresponding optimal score $D(O, Q^*)$ being the Viterbi-score for the entire O; this can yield a per-letter Viterbi-score (or distortion) $v_i = D(O_i, Q^*)/T_i, i = 1, \ldots, L$, for L input unit sequences $O_i, i = 1, \ldots, L$, with corresponding number of monophones as T_i. We will use this later to show the rate-distortion performance of such a 'segment' decoding operation, which in turn impacts the final unit-selection solution.

2) Prosody profile selection by segment Viterbi: Subsequent to deriving $Q^* = (q_1^*, q_2^*, \ldots, q_k^*, \ldots, q_K^*)$ as above, the 'segment Viterbi' takes Q^* as input, and sets up a forced alignment between the segments from O, i.e., $(w_1, w_2, \ldots, w_k, \ldots, w_K)$ and the corresponding segmental unit groups $s_{q_k^*} \in \mathcal{S}$, to derive a sequence of optimal segmental unit indices $\phi^* = (\phi_1^*, \phi_1^*, \ldots, \phi_k^*, \ldots, \phi_K^*)$, where $s_{\phi_k^*} \in S_{q_k^*}$ and points to a segmental unit in the unit database $\mathcal{U}$. ϕ^* is defined and obtained as,

$$\begin{aligned} \phi^* &= \arg\min_{\phi} D(O, \phi) \\ D(O, \phi) &= [(1-\alpha) \sum_{k=1}^{K} d_u(w_k, s_{\phi_k}) \\ &\quad + \alpha \sum_{k=2}^{K} d_c(s_{\phi_{k-1}}, s_{\phi_k})] \end{aligned} \tag{4}$$

where, $d_u(w_k, s_{\phi_k})$ is the unit-cost between the input unit sequence segment w_k and unit database segment s_{ϕ_k}, in terms of optimally aligned string-edit distance, similar to the $d_{seg}(w_k, c_{q_k})$. $d_c(s_{\phi_{k-1}}, s_{\phi_k}) = \beta d(s_{\phi_{k-1}}(l_{\phi_{k-1}}), s_{\phi_k}(1))$, with $\beta = 0$ if $\phi_k = \phi_{k-1} + 1$, and $\beta = 1$, otherwise; here, $d(s_{\phi_{k-1}}(l_{\phi_{k-1}}), s_{\phi_k}(1))$ is the join cost between $s_{\phi_{k-1}}$ and s_{ϕ_k} in terms of 3 constituent costs, namely, the spectral join cost, pitch join cost and gain join cost between the last frame of $s_{\phi_{k-1}}$ and the first frame of s_{ϕ_k}, weighted appropriately to reflect the importance given to each of these joins.

Eqn. (4) is solved by a forced-alignment Viterbi, involving choice of $s_{\phi_k^*}$ from $S_{q_k^*}$ for $k = 1, \ldots, K$ in such a way that the overall combined unit-cost and join-costs are minimized.

Once ϕ^* is obtained as above, this yields the final 'segmental' units $s_{\phi_k}, k = 1, \ldots, K$ which have the best unit-cost with the corresponding segments $w_1, w_2, \ldots, w_k, \ldots, w_K$ of O in terms of symbolic string-edit distance, and the best join-cost between each other in terms of acoustic based (spectral, pitch and gain) join costs, thereby yielding the 'prosody profiles' associated with these $\phi^* = (\phi_1^*, \phi_1^*, \ldots, \phi_k^*, \ldots, \phi_K^*)$ segmental units, that can be transferred to the 1st stage unit sequence for prosody transplantation.

We will summarize the above 2-step procedure (i.e., 'segment decoding' using $\mathcal{P}$, followed by 'prosody profile selection' by 'segment Viterbi' using $\mathcal{S}$ as follows: The input unit sequence O is first decoded (by 'segment decoding') into an optimal sequence of phone-strings indexed as $Q^* = (q_1^*, q_2^*, \ldots, q_k^*, \ldots, q_K^*)$, which are in turn used to retrieve (by 'segment Viteribi') the best segmental units indexed by $\phi^* = (\phi_1^*, \phi_1^*, \ldots, \phi_k^*, \ldots, \phi_K^*)$ with the associated 'prosody-profile' serving the role of 'phonetically conditioned' prosody that matches with the given input text, and which can be transplanted on to the spectral information corresponding to the input unit sequence O. In essence, the two codebooks $\mathcal{P}$ and $\mathcal{S}$ together have learned and represent the phone-string to prosody-profile mapping, via the SKM algorithm and which is in turn retrieved by the 2-step Viterbi procedure.

IV. Experiments and Results

We present results on an Indian language 'Kannada' (one of the 22 scheduled languages of India) with the target prosody as 'story-telling' style. In our experimental framework, the 1st stage unit-selection system uses the mixed-Viterbi formalism proposed by us recently [13], and the 2nd stage unit-selection is as described here ('phone-level' and 'segmental' unit-selection). The 1st stage unit-selection uses a neutral prosody-style database of 1268 sentences ($\sim$ 4 hrs) and the 2nd stage prosodic-database is of 'story-telling' style with 300 sentences ($\sim$ 45 min). Of these 300, as shown in Fig. 2, 280 sentences are used as the 2nd stage unit-database, and the remaining 20 are used as 'out-of-database' (out-of-db) sentences, so that these provide the ground-truth prosody against which to compare the synthetic prosodic contours, i.e., pitch (p), gain (g) and duration (d) profiles, to yield the respective errors e^p, e^g and e^d computed between time-normalized synthesized and ground-truth prosody contours. 20 sentences from within the 280 sentence prosody-database are used as 'in-database' (in-db) sentences, to be able to calibrate the system, and provide the baseline performance for the out-of-db sentences.

We first show the rate-distortion characteristics of the 1st-step 'segment decoding' using the primary codebook $\mathcal{P}$ which behaves like a variable-length segment quantizer (in the original setting of spectral quantization scenario for which such quantizers were developed), but now looked upon as 'quantizing' or 'approximating' the input unit sequence O optimally into the segments $w_1, w_2, \ldots, w_k, \ldots, w_K$. For this, we plot the single-letter Viterbi score $\hat{v}$ (due to the 1st-step 'segment decoding' using $\mathcal{P}$ for a given specification $(P, \bar{l})$) vs the average number of bits per leter b (as a measure of effective bit-rate in representing the variable length entries c_p in $\mathcal{P}$ or in representing the decoded sequence $w_1, w_2, \ldots, w_k, \ldots, w_K$. This is shown in Fig. 4, which shows the rate-distortion ($\hat{v}$ vs b) for various $P = 100, 200, 300, 400, 500$ for two $\bar{l}$ = 5, 8 and 10. It can be seen that use of longer phone-strings in $\mathcal{P}$ moves the rate-distortion curve to the left (e.g. 10 over 8 and 5) indicating the better approximation power of such a primary codebook with $\bar{l}$ = 10 over $\bar{l}$ = 8 or 5. This in turn can be expected to reflect in the more accurate 2nd-step segment-Viterbi choice of units, since the decoding done at the 1st-step is now a better approximation of the input phonetic content.

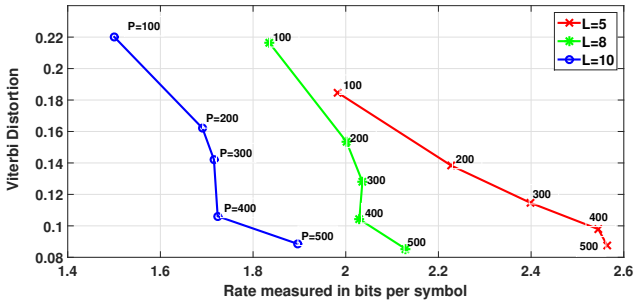

Fig. 4. *Rate-distortion characteristics of the 1st-step 'segment decoding' with the primary codebook for $\bar{l}$ = 5 and 10*

We now show the overall performance by the proposed 'segmental' unit-selection (for various primary codebook sizes $P = 100$ to 500, and $\bar{l} = 10$) in terms of the error measures e^p, e^g and e^d for 20 in-db and out-of-db sentences, and compare it with the errors for the 'phone-level' unit-selection. Fig. 5 shows this (left-panel: in-db and right-panel: out-of-db). It can be noted that 'phone-level' unit-selection consistently has lower errors, owing to the fact they do not perform any approximation other than what is dictated by any choice of α. Unlike this, 'segmental' unit-selection starts with an approximation in the 1st-step using $\mathcal{P}$ and this in turn propagates to determine the overall error. However, it should be noted that the difference in performance is marginal, and 'segmental' unit-selection is competitive to 'phone-level' unit-selection, particularly considering it can optimize the choice of prosody-profiles over a long segmental unit, and also enjoys extremely lower complexity than the full-scale unit-selection.

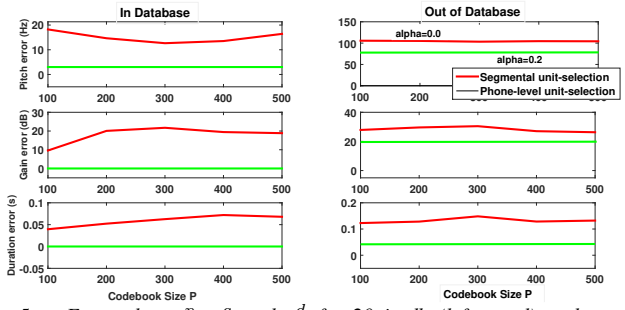

Fig. 5. *Error plots e^p, e^g and e^d for 20 in-db (left panel) and out-of-db Kannada sentences for both 'phone-level' unit-selection and 'segmental' unit-selection*

Fig. 6 compares the pitch contour plots for one out-of-

db Kannada sentence for 'phone-level' and 'segmental' unit-selection along with ground-truth. The 'phone-level' unit-selection has better match at all details, while 'segmental' unit-selection matches well in parts, while also having pitch shifts or distributed pitch errors, which comes by the method committing to a segmental unit based 'prosody-profile' selection, which when incorrect, spreads the error over the entire unit. However, both offer good match, and listening tests bear out the closeness of the two with ground truth. In AB-type listening tests (10 sentences, 5 listeners, native Kannada, A-1st stage prosody retained as is, B-2nd stage prosody due to either of the proposed methods), 92% trials were identified correctly as story-telling style for 'phone-level' unit-selection and 90% trials were identified correctly as story-telling for 'segmental' unit-selection, validating the consistency of the transferred prosody in invoking the perception of the correct prosody-style. Moreover, in AB-type listening tets comparing the two methods (A - phone-level unit-selection, B - segmental unit-selection), 88% trials were in favor of 'phone-level' unit-selection (as being closer to story-telling style categorically), validating the objective measures shown in Figs. 5 and 6.

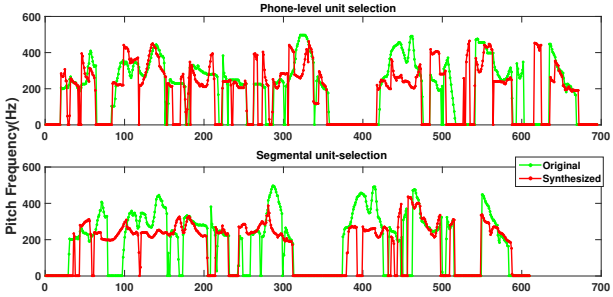

Fig. 6. *Pitch contour plots for one out-of-db Kannada sentences for both 'phone-level' unit-selection and 'segmental' unit-selection compared with ground-truth pitch contour*

An important difference between the 2-step 'segmental' unit-selection and the 'phone-level' unit-selection, is the complexity. Fig. 7 shows the relative complexity of the two unit-selections including symbolc and numeric costs. The segmenal unit-selection is a factor of 25 lower in complexity than the phone-level unit-selection for typical $P, \bar{l}$ sizes of the primary codebook $\mathcal{P}$.

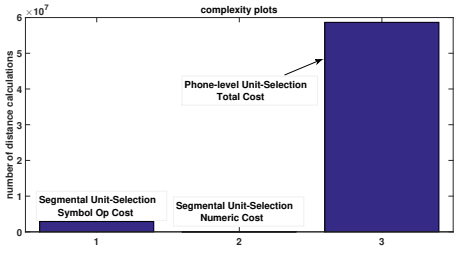

Fig. 7. *Complexity of 'segmental' and 'phone-level' US*

V. Conclusions

We have proposed a unsupervised learning of phonetic-to-prosodic mapping and a segmental unit-selection framework for prosody generation for TTS with advantages of longer phonetic conditioning and low complexity. We have characterized the performance of the proposed framework using various objective measures, comparing it with the phone-level framework and show its practical viability.

VI. Acknowledgements

The authors would like to thank the Department of Information Technology, Ministry of Communication and Technology, Government of India, for funding the TTS Consortium project, "Development of Text-to-Speech synthesis for Indian Languages Phase II", Ref. no. 11(7)/2011HCC(TDIL).

References

[1] M. Bulut, S. Narayanan and L. Johnson. Synthesizing expressive speech overview: challenges, and open questions. Ch. 9, in Text to Speech Synthesis: New Paradigms and Advances, ed. By S. Narayanan, A. Alwan (Pearson Education, Prentice Hall, pp. 197-223, 2005.

[2] E. Eide, R. Bakis, W. Hamza and J. F. Pitrelli. Toward expressive synthetic speech. Ch. 11, in Text to Speech Synthesis: New Paradigms and Advances, ed. By S. Narayanan, A. Alwan (Pearson Education, Prentice Hall, pp. 241-272, 2005.

[3] K. Sreenivasa Rao Predicting Prosody from Text for Text-to-Speech Synthesis. Springer Brief, 2012.

[4] K. S. Lee and R. V. Cox. A very low bit rate speech coder based on a recognition/synthesis paradigm. *IEEE Trans. on Speech and Audio Proc.*, 9(5):482–491, Jul 2001.

[5] K. S. Lee and R. V. Cox. A segmental speech coder based on a concatenative TTS. *Speech Commun.*, 38:89–100, 2002.

[6] V. Ramasubramanian and D. Harish. An unified unit-selection framework for ultra low bit-rate speech coding. *Proc. Interspeech '06, ICSLP*, pp. 217-220, Pittsburgh, 2006.

[7] V. Ramasubramanian and D. Harish. An optimal unit-selection algorithm for ultra low bit-rate speech coding. In ICASSP '07, Hawaii, 2007.

[8] D. Harish and V. Ramasubramanian. Comparison of segment quantizers: VQ, MQ, VLSQ and unit-selection algorithms for ultra low bit-rate speech coding. In *ICASSP '08*, pp. 4773-4776, Las Vegas, 2008.

[9] V. Ramasubramanian and D. Harish. Low complexity near-optimal unit-selection algorithm for ultra low bit-rate speech coding based on N-best lattice and Viterbi search. *Interspeech '08*, pp. 44, Brisbane, 2008.

[10] V. Ramasubramanian and D. Harish. Ultra low bit-rate speech coding based on unit-selection with joint spectral-residual quantization: No transmission of any residual information. *Proc. Interspeech '09*, pp. 2615-2618, Brighton, UK, 2009.

[11] V. Ramasubramanian. Ultra low bit-rate speech coding: An overview and recent results. In Proc. IEEE International Conference on Signal Processing and Communication (SPCOM), Bangalore, 2012.

[12] V. Ramasubramanian and Harish Doddala. Ultra low bit-rate speech coding. Springer-Briefs in Speech Technology, Springer Verlag NY, 2015.

[13] Mythri Thippareddy, C. Mahima, S. Adithya, Sunil Rao and V. Ramasubramanian. G2P-free grapheme-to-speech synthesis: UTF-8 based automatic unit-database annotation and mixed Viterbi unit-selection. Proc. O-COCOSDA '15, Shanghai, China, Oct. 2015.

[14] Mythri Thippareddy and V. Ramasubramanian. Prosody transplantation using unit-selection: Principles and early results. IEEE CONECCT 15, Bangalore, Jul 2015.

[15] Mythri Thippareddy, Noor Fathima, D. N. Krishna, Sricharan, V. Ramasubramanian. Phonetically conditioned prosody transplantation for TTS: 2-stage phone-level unit-selection framework. Accepted in Speech Prosody 2016, May-June 2016, Boston.

[16] M. G. Khanum Noor Fathima, Mythri Thippareddy, M. Arunakumari, H. C. Mamatha, H. N. Supriya, A. Sricharan, V. Ramasubramanian. Phonetically conditioned prosody transplantation for TTS: Unit granularity, context and prosody styles. Submitted to O-COCOSDA 2016, Bali, Indonesia, Oct. 2016.

[17] Y. Shiraki and M. Honda. LPC speech coding based on variable-length segment quantization. *IEEE Trans. on Acoust., Speech and Signal Proc.*, 36(9):1437–1444, Sept. 1988.

[18] M. Honda and Y. Shiraki. Very low-bit-rate speech coding. *In 'Advances in speech signal processing'*, eds. S. Furui and M. M. Sondhi, Marcel Dekker Inc., pp. 209-230. 1992.