IDIAP RESEARCH REPORT

INCREMENTAL SYLLABLE-CONTEXT PHONETIC VOCODING

Milos Cernak Philip N. Garner

Alexandros Lazaridis[a] Petr Motlicek Xingyu Na[a]

Idiap-RR-05-2015

FEBRUARY 2015

[a]Idiap Research Institute

Centre du Parc, Rue Marconi 19, P.O. Box 592, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

Incremental Syllable-Context Phonetic Vocoding

Milos Cernak, Philip N. Garner, Alexandros Lazaridis, Petr Motlicek, Xingyu Na

February 24, 2015

Abstract

Current very low bit rate speech coders are, due to complexity limitations, designed to work off-line. This paper investigates incremental speech coding that operates real-time and incrementally (i.e., encoded speech depends only on already-uttered speech without the need of future speech information). Since human speech communication is asynchronous (i.e., different information flows being simultaneously processed), we hypothesised that such an incremental speech coder should also operate asynchronously. To accomplish this task, we describe speech coding that reflects the human cortical temporal sampling that packages information into units of different temporal granularity, such as phonemes and syllables, in parallel. More specifically, a phonetic vocoder — cascaded speech recognition and synthesis systems — extended with syllable-based information transmission mechanisms is investigated. There are two main aspects evaluated in this work, the synchronous and asynchronous coding. Synchronous coding refers to the case when the phonetic vocoder and speech generation process depend on the syllable boundaries during encoding and decoding respectively. On the other hand, asynchronous coding refers to the case when the phonetic encoding and speech generation processes are done independently of the syllable boundaries. Our experiments confirmed that the asynchronous incremental speech coding performs better, in terms of intelligibility and overall speech quality, mainly due to better alignment of the segmental and prosodic information. The proposed vocoding operates at an uncompressed bit rate of 213 bits/sec and achieves an average communication delay of 243 ms.

1 Introduction

Current very low bit rate (VLRB) speech coders that operate at bit-rates of the order of hundreds bits per second (bps) are designed with a different structure than conventional speech coders for communication. Because of communication delay and complexity limitations, the VLBR coders are currently used only to store large amounts of pre-recorded speech, such as audio books and electronic dictionaries. Indeed, no ITU-T standard has been yet created for bit rates below 4 kbps. The standardisation effort begun in 1994 [1], but it has been shown to be difficult to achieve toll-quality performance in all conditions, such as intelligibility, quality, speaker recognizability, communicability, language independence and complexity.

In this paper we elaborate on the condition on communicability of 200–300 bps VLBR systems. Our aim is to investigate speech coding that can be used as a conventional coder with an acceptable communication delay for real-time speech communication, with a view to being exploited in military and tactical communication systems. To our knowledge such a system is not available. To achieve such low bit rates, parametric speech coding paradigm has to be used. It has already been shown that automatic speech recognition (ASR) and text-to-speech (TTS) can be cascaded to form a vocoder [2, 3, 4, 5], where a set of transmitted parameters is derived from recognised and re-synthesized segments. However, ASR and TTS are both in principle designed to work off-line, with observed significant degradation if employed in an online mode, a basic requirement of a real-time communication.

Humans, by contrast, "encode" speech real-time and in an incremental fashion, i.e., encoded speech depends only on current and past/already-uttered speech and not future/to-be-uttered speech (similar to causality in digital signal processing theory) [6]. Recent neuroimaging studies of Giraud et al. [7] have shown that, during speech processing, the brain generates cortical oscillation in the θ-range (4-7 Hz) that may correspond to the syllable rate, and faster γ-range oscillations (25-40 Hz) that correspond to the phonetic scale. This cortical temporal sampling, i.e., packaging information into units of different temporal granularity such as the phonetic level structural units (phonemes) and the syllabic level structural units (syllables), is thought to play a key role in human speech processing. Further, auditory cortical oscillations show hierarchical phase-nesting or synchronisation across different temporal granularity [8]. Speech communication is also known to be an asynchronous process due to these different information flows embedded in the speech [2].

We can learn from the human speech processing that it has explicit simultaneous phonetic and syllabic components, which are synchronously or asynchronously related. Driven by this, we hypothesise that speech coding inspired by the human speech signal processing can target the communicability requirement of low bit rate coding, and should work asynchronously as well. For that purpose we evaluate synchronous and asynchronous versions of both speech encoder and decoder. In brief, synchronous coding refers to the case where the phonetic vocoder and speech generation process depend on the syllable boundaries during encoding and decoding respectively. On the other hand, asynchronous coding refers to the case where the phonetic encoding and speech generation process perform independently to the syllable boundaries.

The recent work of Flanagan [9], also inspired by human speech processing, proposes parametric speech coding based on an articulatory representation. The drawback of this approach is that it is computationally very expensive (around 100 times real-time only to compute solutions of the Navier-Stokes fluid flow equations). However, since our goal was to design an incremental real-time codec, we have chosen a minimalist approach using a phonetic vocoder. Rather than using a segmental vocoder [10] as the syllabic component observed in the human speech processing, we propose to extend phonetic vocoding with syllable-based (hereinafter called *syllabic*) information transmission mechanisms (e.g., stress, accent, pitch information). The design of the syllable-context phonetic coding and understanding of the synchronisation between phonetic and syllabic information in encoded speech constitute the main contributions of this paper.

The paper is structured in the following way. Section 2 introduces incremental phonetic and syllabic information coding. Section 3 describes in more details the proposed speech coding. Section 4 presents the experiments and results, and finally Section 5 concludes the paper with discussion and future work outline.

2 Incremental Phonetic and Syllabic Information Coding

2.1 Background

The coding process can be abstracted as a copy-synthesis task, i.e., trying to copy as well as possible the audio from the speaker at the transmitter side and synthesize it at the receiver side. From this point of view we can classify parametric VLBR copy-synthesis approaches that differ in the temporal granularity of the re-synthesis method as: (i) frame-based (e.g., formant [11], articulatory [12, 13], MELP synthesis [14]), (ii) phoneme-based (corpus-based [5] or HMM-based [2, 3]), and (iii) segment-based [10, 15, 16]. The trend in lowering the bit rate while preserving high speech quality is to increase temporal granularity of the copy-synthesis, i.e., additionally increasing communication delay. For example, the enhanced MELPe 1.2 kbps coder that employs a context of three consecutive 22.5 ms frames performs nearly as well as 2.4 kbps MELP coder [14]. The temporal granularity can be phonologically interpreted as (sub)phonetic and syllabic information.

2.1.1 Phonetic information

The condition of communicability of the speech encoder leads to the incremental speech processing system, i.e., encoded speech is generated immediately from received current and past speech parameters. If the coder is composed of cascaded ASR and TTS systems, both need to operate incrementally. The phonetic vocoder transmits phonetic information about phoneme segments (their identity along with their durations). We try to investigate a phonetic vocoder that would also allow effective transmission of important information beyond the phonemes.

2.1.2 Syllabic information

Syllables are important supra-segmental units because the span covers the fundamental pitch variant for prosody events as shown in the linguistic research of Xu et al. [17]. Syllabic information consists of accent and stress of the syllable, its length in terms of the number of phonemes, and the vowel name that forms the nucleus of the syllable. Syllables are considered as language-independent and therefore are suitable also for speech coding. The G.114 recommendation regarding mouth-to-ear latency indicates that most users are “very satisfied” as long as latency does not exceed 200 ms [18]. This duration is the average duration of syllables (valid for our English and Mandarin speech data).

2.2 Syllable-context phonetic coding

Rather than using the segmental vocoder as the syllabic component observed in the human speech processing, we propose to extend phonetic vocoding — phoneme ASR — with syllabic information transmission mechanisms, based on syllable context parametric TTS, aiming to unify the transmitted information on the syllable context level. We hypothesise that unifying context across all levels of transmitted information, i.e., phoneme and syllable levels, may decrease the overall bit rate of the speech coder, while allowing acceptable communication delay.

We parametrize the original pitch of each syllable based on the syllable boundaries given by a Syllable Onset Detection (SOD) and by using the discrete (Legendre) orthogonal polynomial (DLOP). DLOP has successfully been used before in this task [19, 20]. The syllabic pitch contour $f\left(\frac{i}{N}\right)$, with the length of $N+1$, is approximated using polynomials as

$$\hat{f}\left(\frac{i}{N}\right) = \sum_{j=0}^{J-1} a_j \cdot \phi_j\left(\frac{i}{N}\right), \quad 0 \leqslant i \leqslant N \tag{1}$$

where the parameters are

$$a_j = \frac{1}{N+1} \sum_{i=0}^{N} f\left(\frac{i}{N}\right) \cdot \phi_j\left(\frac{i}{N}\right), \quad 0 \leqslant j \leqslant J-1. \tag{2}$$

J represents the order of approximation and $\phi_j\left(\frac{i}{N}\right)$ is the j-th DLOP polynomial. Using the transformation described by Eq. 2, a pitch contour is encoded into a parameter vector $(a_0, a_1, \ldots, a_{J-1})$. Considering that the pitch of unvoiced segments is undefined, only the longest contour within a syllable is parameterized. More details and evaluation results of this approach can be found in our previous work [16].

The pitch parametrisation is extended with quantized syllabic accent and stress parametrisation. We use syllable-based accent and stress encoding as described in detail in our previous work [21]. We showed how to infer accent and stress features from the speech signal (hereinafter called also signal-based labels), using syllabic quantized F0 and energy acoustic measures.

We constrain the TTS contextual factors to the syllable level in the decoding module. We have already showed that the context above syllable, i.e., information related to words, phrases or utterances, is less important for the VLBR coding [22].

2.2.1 Synchronisation of incremental encoding

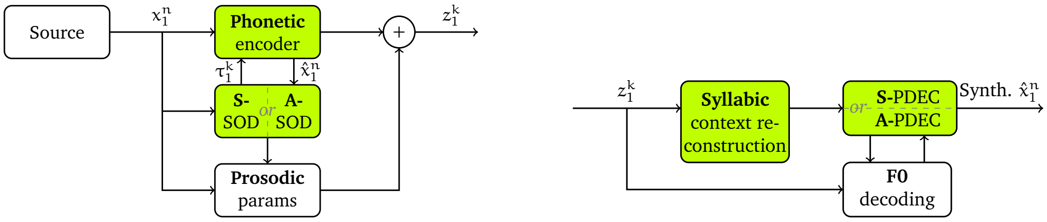
Incremental encoding can be performed using incremental phoneme ASR and incremental syllabification. Conventionally, syllabification is performed on the sequence of phonemes [23], or directly from the speech signal [24]. However, most of the published techniques do not satisfy causality. To combine phoneme ASR and syllabification, satisfying in the same time the causality of the system, we investigated two incremental encoding approaches:

- **Asynchronous encoding**: The phoneme ASR generates the sequence of phonemes from the best partial hypotheses within a fixed time delay τ, usually in the range of 80–200 ms. Asynchronously to the τ, syllable onsets are determined based on the sonority sequencing principle approach [25].
- **Synchronous encoding**: Synchronous encoding is driven by speech signal-based SOD. All the partially recognised phonemes between the syllable onset times, determined by the SOD, then compose the syllables. In this case phoneme ASR and syllabification are synchronous.

2.2.2 Synchronisation of incremental decoding

In the incremental decoding part, the reconstructed syllable context symbolic representation of transmitted speech is used for speech re-synthesis performed by the phonetic vocoder. Similar to the speech encoding, we identified two approaches to re-synthesize the speech incrementally:

- **Asynchronous decoding**: The phonetic vocoder always uses the same previous context (e.g., last two phonemes), asynchronously to the detected syllable boundaries.
- **Synchronous decoding**: The phonetic vocoder uses current syllable determined by the SOD, to synthesize the speech parameters. Similarly to synchronous encoding, the buffer to smoothing speech parameters is synchronous to the syllable boundaries determined by syllable onsets.



(a) Encoder (b) Decoder

Figure 1: The composition of the functional components of the speech coder. Phonetic vocoder consists of a phonetic encoder and two versions of phonetic decoders (PDEC). The encoder (a) can work with a synchronous Syllable Onset Detector (S-SOD) that triggers the phonetic encoder on syllable onset times τ_1^k, or with an asynchronous detector (A-SOD) that depends on the phonetic encoder. On the decoder side (b), the reconstructed syllable-context symbolic representation is used for speech re-synthesis performed by the phonetic decoder that also may work synchronously (S-PDEC) or asynchronously (A-PDEC) with syllable boundaries.

3 Description of the coder

The speech coder is designed as a real-time incremental system operating with an algorithmic (communication) delay of approximately the duration of a syllable. It consists of three basic modules: (A) syllabic speech segmentation, applied in (B) speech encoding and (C) speech decoding.

3.1 Syllabic speech segmentation

Speech segmentation has already been identified as an important property of low bit rate coders. For example, the scalable phonetic vocoder using MELP analysis/synthesis [26] can benefit from the fixed segmentation [27], i.e., segmenting periodically in fixed time intervals. However, we hypothesise that variable speech segmentation, i.e., based on syllable boundaries, can further optimise the speech transmission process. In our proposal, the syllabic component performs such a variable speech segmentation by using the SOD component operating either in asynchronous or synchronous mode.

3.1.1 Incremental encoding

Figure 1a shows the design of the proposed encoder. The phonetic component is based on a phonetic encoder — phoneme ASR — while the syllabic component is based on syllable-context phonetic coding introduced in Section 2. The encoder with the SOD, that plays a role of 'a system clock', outputs syllable tokens $z \in z_1^k$ that each consist of the syllabic symbolic representation and prosody parametrisation of the encoded speech. The pitch parametrisation is complemented with quantized syllabic accent and stress parameters.

3.1.2 Incremental decoding

Figure 1b shows the design of the decoder. It has been shown that missing syllabic-rate information degrades participants ability to identify consonants and to understand sentences [28]. Therefore, analogously to human perception, syllabic context re-construction is processed first. The segmental details are reconstructed using either synchronous (S-PDEC) or asynchronous (A-PDEC) re-synthesis of the phonetic vocoder, and the pitch is decoded from pitch parameters.

3.2 Encoder

3.2.1 Phonetic vocoder analysis

A real-time ASR system is composed of a feature extraction and an incremental search module. We use tracter[1], a data-flow framework, as an interface between the ASR's pull architecture and the Analogue to Digital Converter's push architecture [29]. Figure 2 shows the directed graph of components.

[1] https://github.com/mcernak/tracter

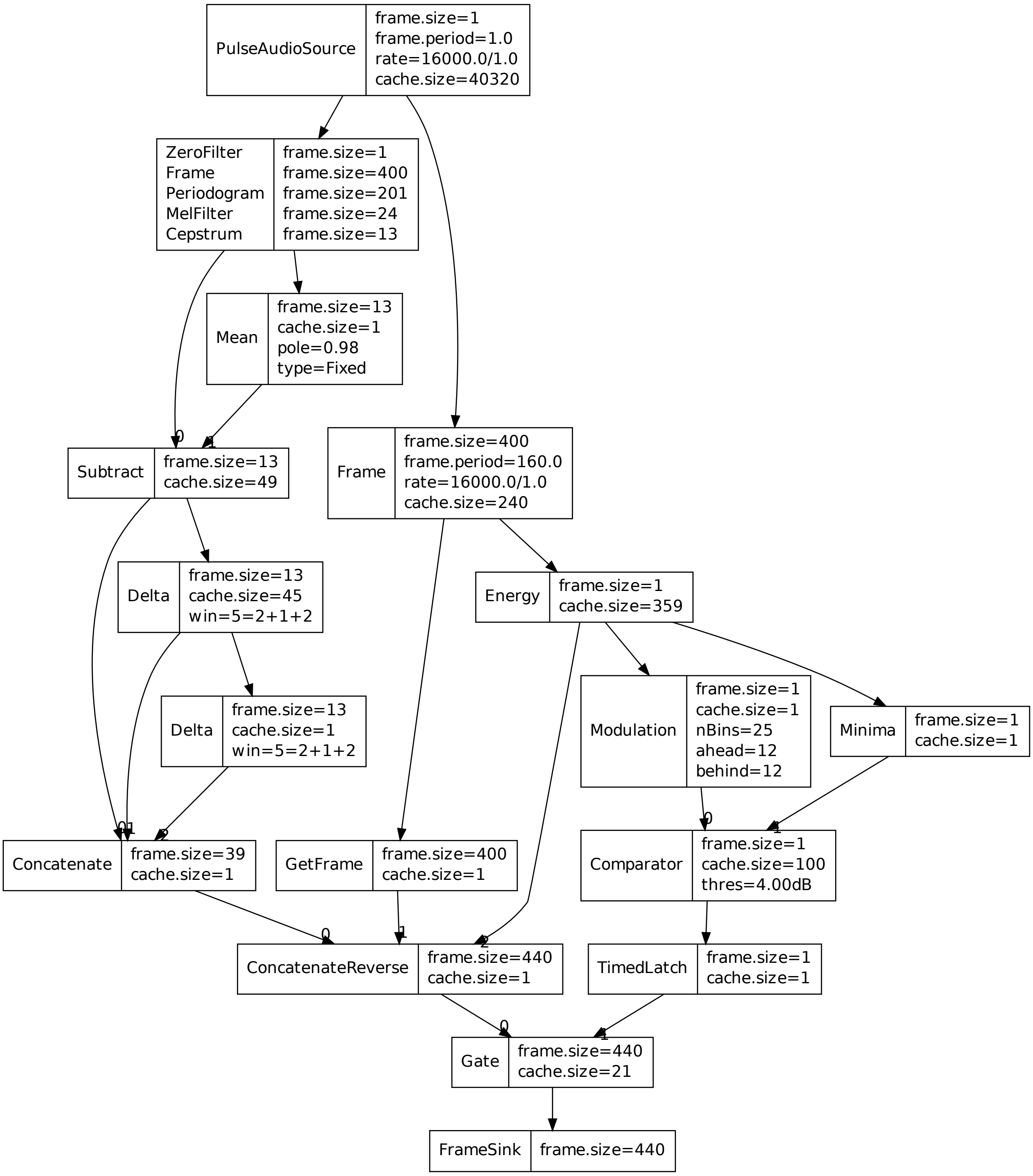


Figure 2: Front-end of the encoder. The source component can be selected from microphone, file or TCP/IP socket inputs. The gate component is controlled using minima-based voice activity detection. The front-end outputs 39 dimensional mel-cepstral features plus an energy feature, and passes all speech samples for syllabic pitch parametrisation calculated directly in the encoder.

The incremental encoding is triggered by a minima-based voice activity detection [30, 31]. In addition to conventional mel-frequency cepstral features with adaptive cepstral mean normalisation, the feature extractor also provides energy features. As the feature extractor is not aware of syllable boundaries, it also pushes all speech samples to the encoder for syllabic pitch parametrization that is performed directly by speech encoder.

The speech encoder is based on an incremental phonetic ASR system[2], a Weighted Finite State Transducer (WFST) based recognition system. The WFST graph is basically composed of an n-gram phoneme

[2] https://github.com/mcernak/juicer

Table 1: A sonority scale used in the asynchronous incremental syllabification.

Sounds	Level
Vowels	4
Glides (w, j)	3
Liquids (l, ɾ)	2
Nasals (m, n, ŋ)	1
Obstruents (all other consonants)	0

LM. The ASR pulls features frame-by-frame from tracter, and implements a partial decoding mechanism that traces active paths and returns the path of the best token with each processed frame. The encoded speech is segmented into the syllables using synchronous or asynchronous SODs. The encoder then integrates the TEMPO method [32] for syllabic pitch extraction using 10 ms frame shift. Optionally, syllable-based Kalman smoothing of the raw pitch values is used before pitch curve fit encoding. Finally, the syllable tokens are formed for transmission.

3.2.2 Syllabic prosody packaging

In asynchronous incremental syllabification (A-SOD in Figure 1a), based on the phoneme identity, a sonority sequencing principle with onset maximisation is applied to determine the syllable onsets. We use a simplistic approach with the sonority distance parameter set to 2. Table 1 shows the sonority scale that was used.

In synchronous incremental syllabification (S-SOD in Figure 1a), we use text-based pre-processed syllable boundaries, as an oracle syllable onset detector. We hypothesise that any robust speech-based incremental syllable onset detector, which to our knowledge is not available now, can only reach (and not overcome) performance of the text-based syllabification. Pre-processed syllable boundaries were generated from text by finding the minimum sonorant position between vowels [25], as applied in the Festival system [33].

Stress, accent and pitch information is encoded using syllable-based techniques introduced in the previous Section 2.2.

3.3 Decoder

We employ the HMM-based speech synthesis system (HTS) [34] with STRAIGHT re-synthesis [32] at the receiver site. By default, HTS synthesizers need a contextual symbolic speech representation (hereinafter called *labels*), where each phonetic symbol to be synthesised depends on all words and phrases in the synthesised utterance. As stated in the previous Section 2.2, we use only syllable context phonetic labels.

3.3.1 Syllable-context reconstruction

For building the syllable-context phonetic decoder, the contextual factors used in the decoding are reduced to the ones available from the incremental ASR with applicable syllabifier. Thus, word, phrase and utterance related factors are removed from the training labels. The conventional text-based labels of stress and accent are replaced by the syllabic signal-based labels. The decoder then reconstructs all the 14 contextual factors listed by Table 2 from input syllable tokens z_1^k.

3.3.2 Phonetic vocoder re-synthesis

In the task of real-time speech synthesis, it is required that the processing time of the current speech unit is shorter than the duration of the previous synthesized unit. Aiming at constructing a real-time incremental speech coding system, we propose real-time speech synthesis from two aspects. We investigate both synchronous and asynchronous parameter generation methods, S-PDEC and A-PDEC respectively. The decoder is designed and implemented in such a way that for syllabic contexts,

1. real-time parameter generation methods do not distort the generated parameter trajectories,
2. and online STRAIGHT re-synthesis does not degrade output voice quality.

Table 2: Contextual factors used in the syllable-context decoder. Syllable-context factors $PS$ and $CS$ stand for the previous and current syllable, respectively. Position starts from 0 and number starts from 1.

No.	Contextual factors
1.	the phoneme before the previous phoneme
2.	the previous phoneme
3.	the current phoneme
4.	the next phoneme
5.	the phoneme after the next phoneme
6.	forward position of the current phoneme in $PS$
7.	backward position of the current phoneme in $CS$
8.	the number of phonemes in $PS$
9.	the number of phonemes in $CS$
10.	name of the vowel in $CS$
11.	the quantized F0 p_i label of the $PS$
12.	the quantized energy e_i label of the $PS$
13.	the quantized F0 p_i label of the $CS$
14.	the quantized energy e_i label of the $CS$

To implement asynchronous decoding (A-PDEC on Figure 1b), we use the performative HTS-based speech synthesis system. In the streaming/performative-HTS framework [35], it is assumed that a real-time system may require some knowledge of the future of its input to produce the current output. Thus, its requirement for time lookahead is implemented in a buffer. This assumption is true in term of performative real-time, as well as in the real-time speech coding task. In the present work, the incremental asynchronous decoding is implemented as performative-HTS speech parameter generation with 2 previous phoneme smoothing [36]. Although the algorithmic delay varies according to the phoneme duration, the synthesizer operates real-time using the current setup of the speech coding system. Cepstral instead of mel-cepstral features are used, as re-synthesis without mel-warping was almost two times faster.

To implement synchronous decoding (S-PDEC in Figure 1b), we use a low latency parameter generation method called MLPG-b proposed for HMM-based real-time speech synthesis [37]. In the conventional HTS systems, speech parameters are generated by solving

$$\mathbf{R}\mathbf{c} = \mathbf{r}, \tag{3}$$

where $\mathbf{c}$ represents the acoustic parameter sequence to be used for synthesising the speech signal. To solve Equation 3, matrices $\mathbf{R}$ and $\mathbf{r}$ are calculated given the HTS models. $\mathbf{R}$ is determined by the variance matrix, and $\mathbf{r}$ is determined by the mean vectors, which describe the sequence of Gaussian-based acoustic models at the sentence level. However, MLPG-b was proposed by generating the phoneme sequence and the acoustic parameter trajectories as context-dependent semantic blocks. By segmenting the input labels into context-dependent blocks – syllables, Eq. (3) is approximated by

$$\{\mathbf{R}_y \mathbf{c}_y = \mathbf{r}_y\}_{y=1}^{Y}, \tag{4}$$

where $\mathbf{c}_y$ represents the acoustic parameter of the yth block. $\mathbf{R}_y$ and $\mathbf{r}_y$ denote the corresponding blocked matrices. By solving a batch of Y equations consecutively, the whole sentence of parameters is generated. The advantage of MLPG-b is that the rendering of speech can be started shortly after the synthesis phase begins, without necessity to wait for the completion of the whole sentence. The evaluation results indicate that the generated acoustic parameters by MLPG-b system are not significantly distorted compared with the HTS system, and it is faster than real-time even in an extremely limited computational environment.

Unlike the asynchronous decoding system, the synchronous system does not apply a lookahead and smoothing strategy. According to the theoretical analysis of the matrices of Eq. (4), the generated parameters given high-level contextual blocking boundaries, which in this case are the syllabic boundaries, promise an acceptable continuity of the synthesised speech [37].

Once the parameters have been generated using synchronous or asynchronous systems, speech samples are eventually re-synthesized using a real-time incremental (online) version of the STRAIGHT re-synthesis. In this way speech samples are generated frame-by-frame, either with each processed phoneme with the asynchronous system, or with each syllable in the synchronous system. Comparing the offline and online re-synthesis, the difference in the quality of the generated speech is minimal. In the offline version, the bias used in smoothed spectrum to cepstrum calculation is a scaled minimal value of whole spectrum of synthesized speech, while in the online version this bias is estimated only from the current frame of generated speech. This local bias estimation finally results in slightly suppressed spectral amplitudes over the whole frequency range, nevertheless it is perceptually negligible.

4 Experiments

We first evaluate the proposed decoder with oracle encoder. This includes evaluation of syllabic accent and stress parameters, used during the decoding. Afterwords we evaluate the whole coding system with an integrated encoder. Thus, we evaluate:

1. **Decoder with oracle encoder**, measuring qualities of log quantized syllabic pitch and energy parameters, and focusing on synchronous vs. asynchronous incremental speech signal re-synthesis,

2. **Integrated encoder and decoder**, focusing on the impact of the latency of incremental ASR on speech encoding, and evaluating synchronous vs. asynchronous incremental syllabification.

First, we describe the training of the acoustic models used for speech encoder and decoder. Then we continue with the performed experiments and discussion of the results.

4.1 Training

Two American English acoustic models, first for the encoder (HMM/GMMs system) and second for the decoder (HTS system) were developed, both using the CMU pronunciation dictionary. The dictionary consisted of 40 unique phonemes including silence.

4.1.1 Phonetic encoding

We trained HMM/GMM systems for phonetic encoding using the Wall Street Journal WSJ0 and WSJ1 continuous speech recognition corpora [38]. Three-state, cross-word triphone models were trained with the HTS variant [34] of the HTK toolkit on the *si_tr_s_284* set of 37,514 utterances. We tied triphone models with decision tree state clustering based on the minimum description length (MDL) criterion [39]. The MDL criterion allows an unsupervised determination of the number of states. In this study, we obtained 12,685 states each modelled with a GMM consisting of 16 Gaussians.

The phoneme language model was trained from mono-phone transcription of the acoustic model training set, using Witten-Bell discounting for N-grams of order 3.

Finally, WFST models were composed using the Juicer tools and the AT&T FSM library [40] into the final $C \circ L \circ G$ transducer.

4.1.2 Phonetic decoding

For the re-synthesis, the HTS system trained using the CMU-ARCTIC database was used.

In order to justify an application of the syllabic signal-based labels, the Mutual Information (MI) measure between the text-based and signal-based labels was applied. MI is a measure for evaluating the statistical information shared between two quantities [41]. The proposed syllabic quantized F0 p_i and energy e_i features were calculated from the speech signal measures, and the labels $M_i \in \{0, 1, \dots, 7\}$ were assigned also to the current and previous syllable (8 labels resulted from use of 3-bit code-books). To combine the p_i and e_i features (e.g., to capture higher pitch and lower energy) into a single label, we constrained $M_i = p_i = e_i$, where the value of the label in question was the same for both acoustic measures. It simplified the construction of the question set for the context clustering as well.

The MI as a measure of the conventional text-based *stress* and *accent* labels C and the signal-based M_i labels can be defined as:

$$I(C; M_i) = \sum_{c} \sum_{m \in M_i} p(c, m) \log_2 \left(\frac{p(c, m)}{p(c)p(m)} \right), \tag{5}$$

where $p(c, m)$ is the joint probability of C and M_i, and $p(c)$ and $p(m)$ are marginal probabilities. The MI is a measure of information in bits conveyed by M_i about C, and it is normalised by the mutual information measure with the entropy of C defined as:

$$H(C) = -\sum_{c} p(c) \log_2 p(c). \tag{6}$$

We evaluated the normalised measure $\frac{I(C;M_i)}{H(C)}$ for the following classes (options) of C:

1. C_a, where $C_a =$ *accent*, and is a measure of information in bits that conveys M_i about conventional *accent* labels,
2. C_s, where $C_s =$ *stress*, and is a measure of information in bits that conveys M_i about conventional *stress* labels,
3. $C_{s \wedge a}$, where $C_{s \wedge a} =$ *stress* $\wedge$ *accent*, and is a measure of information in bits that conveys M_i about the intersection of conventional *stress* and *accent* labels, i.e., $C_a \cap C_s = \{c : c \in C_a \wedge c \in C_s\}$.
4. $C_{s|a}$, where $C_{s|a} =$ *stress|accent*, and is a measure of information in bits that conveys M_i about the union of conventional *stress* and *accent* labels, i.e., $C_a \cup C_s = \{c : c \in C_a \text{ or } c \in C_s\}$,

Table 3 shows the normalised MI values calculated for the male `bdl` and female `slt` testing speakers from the CMU-ARCTIC speech database [42].

Table 3: *Normalised MI values of* `bdl` *and* `slt` *voices.*

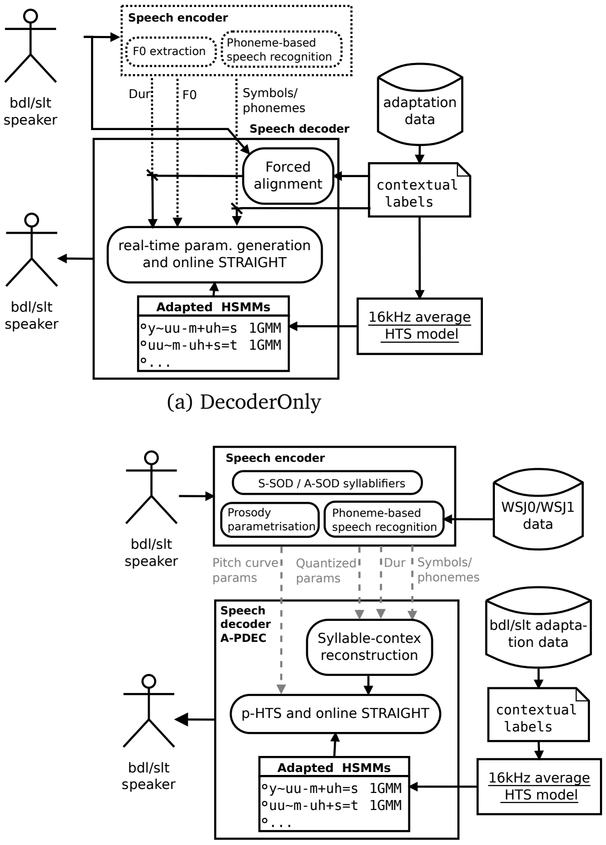
$\frac{I(C;M_i)}{H(C)}$	`bdl`			`slt`		
	p_i	e_i	p_i, e_i	p_i	e_i	p_i, e_i
$C = C_a$	10.1	8.2	16.7	10.8	8.6	16.7
$C = C_s$	10.9	7.4	18.3	11.8	8.7	18.3
$C = C_{a \wedge s}$	11.4	9.6	17.9	11.8	9.7	17.9
$C = C_{a\|s}$	12.4	8.5	**20.6**	13.7	10.0	**20.6**

From the normalised MI values we see that (a) p_i is more informative than e_i which is more evident in the female speaker, and (b) individual p_i and e_i values are less predictive than their combination. The best predicted syllables using the signal-based labels are those which are either accented or stressed, i.e., those represented by the conventional *stress* or *accent* label class $C_{a|s}$. From this analysis we can conclude that the proposed syllabic M_i labels have enough predictability power to replace the conventional text-based stress and accent labels. In that way we do not explicitly model stress and accent. Rather, we hypothesise that the context clustering does the job in a data-driven manner, and selected contextual questions about M_i values will reflect the information about actual conventional stress and accent features.

The signal-based labels were generated for all the training speech data as described above. All log F0 measurements of the training set were extracted using the TEMPO method exploiting a 5 ms frame shift, and the log energy values were calculated using the tracter signal-processing tool. The syllable boundaries were extracted from the contextual labels provided by the CMU-ARCTIC database; average values were calculated per syllable. The log F0 quantization code-books, created per speaker, were linearly spaced between the $\mu - 3\sigma$ and $\mu + 3\sigma$ boundaries, where μ is the mean and σ is the standard deviation of all the measurements belonging to the training data of a particular speaker. The order of energies of accented/stressed syllables was ten. In order to compress the range of less important lower energies for estimation of quantization boundaries, we considered only energies above 1. This resulted in more values of log energies slightly above 0, and compressing the left quantization boundary from $\mu - 3\sigma$ to $\mu - 2\sigma$.

An average model was built using five speakers, including 3 males (`bdl`, `jmk` and `rms`) and 2 females (`clb` and `slt`). Each utterance with unique `id` was assign to:

- the training set, if $0 \leqslant \mathtt{id} \leqslant 450$,
- the test set, if $450 < \mathtt{id} \leqslant 500$, and
- the adaptation set, if $\mathtt{id} > 500$.



(a) DecoderOnly

(b) EncoderAndDecoder

Figure 3: Speech coding experimental setups: (a) abstracting the encoder for evaluation of incremental speech decoding and the syllabic prosody parameters, and (b) integrated speech encoder and A-PDEC decoder for evaluation of overall speech quality.

In this way, the training set of the average model comprised of 4493 sentences (some corrupted utterances were excluded from the training). The `bdl` and `slt` speakers were selected as testing speakers. The `bdl` and `slt` adaptation sets of 131 sentences were used for a constrained structural maximum a posteriori linear regression (CSMAPLR) adaptation [43] of the average model. Finally the `bdl` and `slt` test sets of 200 sentences were used for evaluating the two systems.

4.2 Decoder with oracle encoder

Figure 3a shows an experimental setup to evaluate the decoder. We first evaluate incremental speech decoding with HTS models trained only with the first 10 contextual factors, listed in Table 2. Then we continue with evaluation of the last 4 contextual factors of the table – the syllabic accent and stress parameters.

We abstracted here the encoder side, and used the true input to the decoder, i.e., the phoneme sequence from the syllable context labels. The phoneme labels of the evaluation set were aligned with the speech samples, and the original pitch was transmitted to the decoder.

4.2.1 Incremental speech decoding

The overall quality of the re-synthesised speech was evaluated subjectively using the Degradation Category Rating (DCR) procedure [44] quantifying the Degradation Mean Opinion Score (DMOS). This method provides a quality scale of high resolution, due to comparison of a distorted (synthesized) signal with a (natural/original) reference. The aim was to capture speech encoding quality variations based on the different speech parameter generation and re-synthesis methods.

The test consisted of 6 sentences (3 from male `bdl` and 3 from female `slt`) randomly chosen from the ARCTIC database, with length of at least 2 seconds. Listeners rated the following 4 versions of the synthesis systems:

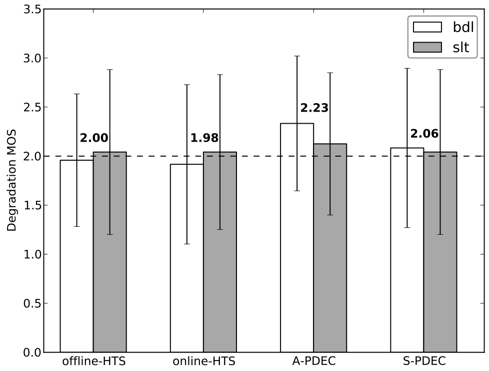


Figure 4: Subjective evaluation results for different HTS re-synthesis systems. The numbers above the bars are the averages for both speakers together.

1. **offline-HTS**: HTS v.2.1 speech parameter generation – from transmitted parameters to the cepstral coefficients, using the `hts_engine` API v.1.06, with offline STRAIGHT re-synthesis – from cepstral coefficients to the speech samples.

2. **online-HTS**: HTS v.2.1 speech parameter generation with online STRAIGHT re-synthesis.

3. **A-PDEC**: asynchronous incremental phoneme-based speech parameter generation using the performative HTS system with online STRAIGHT.

4. **S-PDEC**: synchronous incremental syllabic speech parameter generation with online STRAIGHT.

Eight listeners were asked to rate the degradation of re-synthesised signals compared with reference signals based on their overall perception. According to the DCR procedure, it is not fair to build a pair associating two synthetic signals since it would have implied that the first synthetic signal outclasses perception of the second one. Therefore natural speech was selected as a reference signal in the test. Listeners had to describe degradation within the following five categories:

1. Very annoying,

2. Annoying,

3. Slightly annoying,

4. Audible but not annoying,

5. Inaudible.

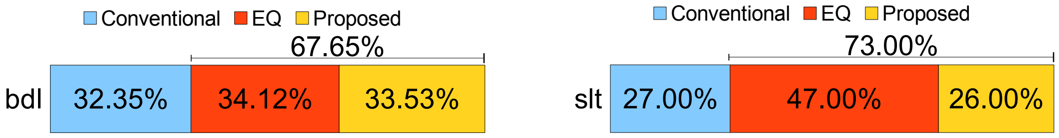
We evaluated two tests based on the two hypotheses presented above: (a) whether online STRAIGHT degrades the generated speech compared to the offline STRAIGHT (i.e., comparing the performance of the first and second systems), and (b) whether real-time speech parameter generation further degrades the speech quality (i.e., comparing the second and last two evaluated systems).

Figure 4 shows the subjective evaluation results. A t-test confirmed that the differences of the first test (a) is not statistically significant ($p > 0.05$), so we can conclude that online STRAIGHT re-synthesis works reasonably well. In the second test (b), the difference between the online-HTS system and asynchronous A-PDEC is significant ($p < 0.05$), and the difference between the two incremental systems is statistically insignificant. We chose the A-PDEC decoding for integrated encoder and decoder experiments.

4.2.2 Syllabic accent and stress parameters

This evaluation follows our previous work [21]. Here, we extended the testing set that now consists of one male and one female speaker. We trained two HTS systems, (i) **proposed** — with all 14 contextual factors as listed in Table 2, and (ii) **conventional** — where we replaced last 4 contextual factors by stress and accent features inferred from the text.

To evaluate syllabic accent and stress encoding, an ABX test was conducted. The motivation for the ABX test was to see whether the use of stress and accent information based on the speech signal on the



(a) BDL

(b) SLT

Figure 5: ABX subjective evaluation test results (in percentages) for the comparison between the conventional and proposed systems, for (a) `bdl` and (b) `slt` speakers.

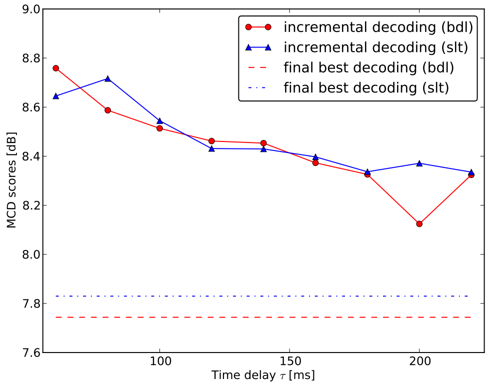


Figure 6: Impact of incremental ASR with time delay τ and A-SOD syllabification on cepstral distortion of encoded speech, for male `bdl` and female `slt` test speakers.

encoder's part (rather than based on textual labels), will affect the overall quality of the synthetic speech on the decoder's side.

According to [45], the ABX test is suitable for rating small degradation using a continuous impairment scale, and expert (trained) listeners should be used. The listeners were asked to choose between speech samples produced from the conventional and the proposed systems. 17 listeners for evaluation of the `bdl` samples and 10 listeners for evaluation of those of `slt` participated in the listening tests. In each test, the listeners were asked to listen for each pair of sentences to the two samples (as many times as they wanted), and choose between the two samples in terms of the overall quality. Additionally, the listeners could choose a third option, “both samples sound the same”, if they had no preference between them. For both tests, the same 10 sentences from the test set were used.

Figure 5 presents the results of the listening test. The proposed system, i.e., the system with the syllabic stress and accent speech signal-based labels, achieves similar performance to the conventional system, i.e., the system with the conventional stress and accent text-based contextual factors. These results clearly validate our hypothesis that speech signal-based stress and accent features can perform as well as text-based contextual factors.

4.3 Integrated encoder and decoder

Having evaluated the HTS re-synthesis and syllabic prosody transmission components, we integrated all the speech coding modules to a full speech coding system. Figure 3b shows the experimental setup.

First, we objectively evaluated the impact of incremental ASR on encoded speech. The incremental phoneme ASR was done by parsing the best partial hypothesis on regular time intervals τ. Asynchronous syllabification (A-SOD system) is then done incrementally with each ASR label, based on the sonority sequencing principle and the syllable onset maximisation [23]. Smaller τ impacts intelligibility of the encoded speech, while larger τ increases the encoding delay.

Mel Cepstral Distortion (MCD) [46] on the test set is used as an objective metric for evaluating the impact of incremental speech recognition on quality of speech coding. Figure 6 shows MCD scores on mel-cepstral vectors of original and asynchronously incrementally encoded speech for different $\tau = 60, 80, 100, \ldots, 220$ ms. In the synchronous S-SOD system, $\tau = \tau_1^k$, with an average syllable duration about 200ms. The encoding delay $\tau = 200ms$ was selected for further stimuli generation in subjective evaluation of quality.

Higher MCD scores are caused by phonetic misalignment as the speech is re-synthesized from phonetic

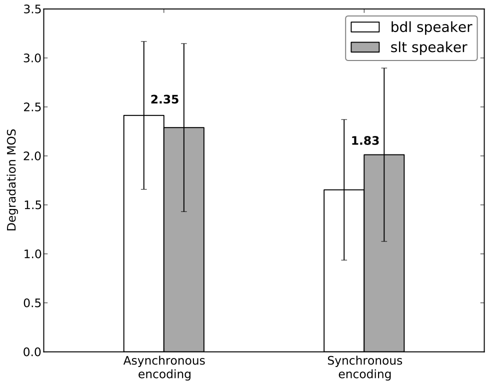


Figure 7: Subjective evaluation of the encoded speech quality for A-PDEC speech decoder. Asynchronous encoding uses the A-SOD system while synchronous encoding uses the S-SOD system.

ASR labels. The phonetic ASR was the WFST system, where due to the label pushing algorithm during WFST composition and minimisation, phoneme boundaries are estimated less accurately. However, this does not have a significant impact on perceived speech quality, as we show in the next evaluation test. This test was performed again as a DMOS subjective test to quantify overall speech degradation. The aim was to capture overall speech encoding quality variations based on the different incremental syllable onset detection methods.

11 listeners were asked to rate the degradation of encoded signals compared with reference signals based on their overall perception. The test consisted of 17 sentences (10 from male `bdl` and 7 from female `slt`) randomly chosen from the ARCTIC database, with length of at least 2 seconds. Figure 7 shows the scores. A relative improvement of 22% was obtained by asynchronous incremental coding. In addition, we performed also Perceptual Evaluation of Speech Quality test [47], where we obtained an average 1.45 PESQ MOS for the asynchronous coding and an average 0.88 PESQ MOS for the synchronous coding, with the relative improvement of 40% for the asynchronous coding. The lower PESQ absolute values can be explained by the implicit quality expectations of the PESQ method. The PESQ MOS corresponds to absolute category ratings like “good”, “fair”, etc. (defined in ITU-T Rec. P.800, sec. B.4.5), so the meaning of what is considered “good” depends on the expectation of the listener. The vast majority of databases with which PESQ were trained contain speech that was processed with commercial codecs, which operate at much higher bit rates. So the quality expectation is completely different from that of a user of military communication equipment we work on. In other words, the PESQ MOS reflects the opinion of an average listener who is used to commercial telephony services, and such a user would surely think that a VLBR codec sounds “poor” or even “bad” compared to what he or she is used to.

A t-test of both subjective and objective tests confirmed that the differences between the synchronous and asynchronous incremental encoding are significant ($p < 0.01$).

To better understand differences in the overall speech coding quality, we conducted an intelligibility test based on the objective Speech Intelligibility Index test [48], with no background noise. All recordings from the listening test were analyzed by a one-third-octave filterbank with band center frequencies given in Hertz [160, 200, 250, 315, 400, 500, 630, 800, 1000, 1250, 1600, 2000, 2500, 3150, 4000, 5000, 6300, 8000]. For the range from 160 Hz to 1250 Hz, a multirate filter implementation was used. Then we normalised power to 83 dB SPL, and calculated speech intelligibility indexes 0.90, 0.91 and 0.92, for the synchronous encoding setup, the asynchronous encoding setup, and original human recordings, respectively. Though the differences are relatively small, they are statistical significant (two-tailed t-test, $p < 0.05$), and partially explain lower degradation and PESQ MOS of the synchronous encoding setup. Indeed, we observed in synchronous encoding more mismatches between the phonetic and prosodic information, for example in the form of malformed syllables. That impacts intelligibility and also pitch and accent transmission, which are based on syllable-boundaries.

The asynchronous system that combined the A-SOD and A-PDEC modules performed significantly better in terms of overall speech quality degradation than the synchronous one that combined the S-SOD and S-PDEC modules. By comparing encoded speech quality with oracle transmitted parameters in Figure 4 and asynchronously encoded speech quality in Figure 7, we notice an improvement from 2.23 DMOS to 2.35. This might be an impact of quantized prosody transmission that was included in the second system. This also confirms that the phonetic ASR classified broad phonetic classes correctly and intelligibility was maintained.

Table 4: Asynchronous syllable-context phonetic vocoder bit allocation.

Parameter	Bits/unit	Unit	bps
Phoneme	6	Phoneme	68.6
Duration	5	Phoneme	57.2
Energy	3	Syllable	12.3
Average pitch	3	Syllable	12.3
DLOP $a_i(,a_2)$	8(16)	Syllable	62.4(124.8)
Total			**212.8(275.2)**

4.4 Bit rate and algorithmic delay

The test set contained 2437 syllables with 6759 phonemes in 592 seconds of speech including silences. On average, there were 11.43 phonemes/sec and 4.11 of syllables/sec. Encoding of 40 unique phonemes required 6 bits/phoneme, and their duration encoded as a number of 10ms frames required 5 bits/phoneme (setting 320ms as maximal duration of a phoneme was sufficient). The transmission of the syllabic context required 6 bits/syllable for accent and stress parameters, and 2 bytes/syllable for pitch parameters. The first Legendre parameter a_0 represents the mean of the contour since $\phi_0 = 1$. Therefore a_0 does not have to be transmitted as it can be reconstructed from the syllabic pitch parameter p_i.

The bit allocation for the asynchronous syllable-context phonetic vocoder is shown in Table 4. We showed in [16] that by using the second order DLOP instead of the third order, parametrisation does not impact the quality of the encoded speech. This option is shown in the table as well.

The average syllable duration, including leading, trailing and short pause silences, was 243 ms. As both speech encoding and decoding processing were faster than real-time, we consider the average syllable duration as an algorithmic latency of 243 ms of the proposed coder. According to the G.114, the users are “very satisfied” as long as latency does not exceed 200 ms, and “satisfied” as long as latency does not exceed 280 ms [18].

5 Conclusion

We have shown that phonetic vocoding can be extended with syllabic prosody transmission. We have investigated how the communicability requirement for speech coding affects the design and implementation of a coder that is based on cascaded ASR and TTS. We have thus studied the incremental versions of both modules, which leads to synchronisation issues of phonetic and syllabic information transmission.

We found that both incremental synchronous modules performed worse than their asynchronous versions. Synchronous incremental speech decoding (re-synthesis), performed worse than asynchronous decoding. This is because inter-syllable context, that was beyond transmitted information in the synchronous system, but still recovered from phonetic information in the asynchronous system, seems to be important for smoothed speech re-synthesis. We found that synchronous incremental speech encoding also performed significantly worse than the asynchronous one in terms of intelligibility and speech quality degradation.

Our experiments confirmed that the asynchronous syllable-context phonetic vocoder that combined both asynchronous A-SOD and A-PDEC systems achieved the best overall intelligibility and speech quality. We believe that it may be related to the fact that human speech communication is asynchronous as well. Our results support findings that syllable-rate information is less synchronised to phoneme-rate information, as observed in infant and adult-oriented speech [49]. We trust that the presented speech coding framework could partially contribute to better understanding of hierarchical phase-nesting of auditory cortical oscillations by allowing further simulations.

According to the evaluation of the proposed speech coder, we can conclude that it is (i) effective in terms of very low bit rates, and (ii) incremental encoding speech with an acceptable communication delay. The codec uses an old-fashioned ASR front-end; its achieving reasonably good quality may provide us some proof of its usability for incremental low bit rate speech coding. We also evaluated syllabic prosody packaging/transmission within a full speech codec experimental setup, and confirmed its usability. Quantization was speaker-dependent in this work; we currently work on speaker-independent quantization schemes. Developing a more robust and language independent phonetic vocoder is also one aspect of our future work.

The codec is prototyped in C++, and demonstrated at http://www.idiap.ch/project/recod/demo/incremental-coding.

Acknowledgment

This research was partly supported under the RECOD project by Armasuisse, the Procurement and Technology Center of the Swiss Federal Department of Defence, Civil Protection and Sport.

References

[1] S. Dimolitsas, C. Ravishankar, and G. Schroder, “Current objectives in 4-kb/s wireline-quality speech coding standardization,” *IEEE Signal Processing Letters*, vol. 1, pp. 157–159, Nov. 1994.

[2] J. Picone and G. R. Doddington, “A phonetic vocoder,” in *Proc. of ICASSP*, pp. 580–583 vol.1, IEEE, May 1989.

[3] K. Tokuda, T. Masuko, J. Hiroi, T. Kobayashi, and T. Kitamura, “A very low bit rate speech coder using HMM-based speech recognition/synthesis techniques,” in *Proc. of ICASSP*, vol. 2, pp. 609–612 vol.2, IEEE, May 1998.

[4] K.-S. Lee and R. Cox, “A very low bit rate speech coder based on a recognition/synthesis paradigm,” *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 9, pp. 482–491, Jul 2001.

[5] G. V. Baudoin and F. El Chami, “Corpus based very low bit rate speech coding,” in *Proc. of ICASSP*, vol. 1, pp. I–792–I–795 vol.1, IEEE, Apr. 2003.

[6] W. J. M. Levelt, *Speaking: From Intention to Articulation (ACL-MIT Series in Natural Language Processing)*. A Bradford Book, Aug. 1993.

[7] A.-L. L. Giraud and D. Poeppel, “Cortical oscillations and speech processing: emerging computational principles and operations.,” *Nature neuroscience*, vol. 15, pp. 511–517, Apr. 2012.

[8] P. Lakatos, A. S. Shah, K. H. Knuth, I. Ulbert, G. Karmos, and C. E. Schroeder, “An oscillatory hierarchy controlling neuronal excitability and stimulus processing in the auditory cortex.,” *Journal of neurophysiology*, vol. 94, pp. 1904–1911, Sept. 2005.

[9] J. Flanagan, “Parametric representation of speech signals [dsp history],” *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 141–145, 2010.

[10] J. Cernocky, G. Baudoin, and G. Chollet, “Segmental vocoder-going beyond the phonetic approach,” in *Proc. of ICASSP*, vol. 2, pp. 605–608 vol.2, IEEE, May 1998.

[11] J. Holmes, “The influence of glottal waveform on the naturalness of speech from a parallel formant synthesizer,” *Audio and Electroacoustics, IEEE Transactions on*, vol. 21, pp. 298–305, Jun 1973.

[12] C. Goodyear and D. Wei, “Articulatory copy synthesis using a nine-parameter vocal tract model,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 385–388 vol. 1, May 1996.

[13] Y. Laprie, M. Loosvelt, S. Maeda, R. Sock, and F. Hirsch, “Articulatory copy synthesis from cine x-ray films,” in *Proceedings of Interspeech*, pp. 2024–2028, 2013.

[14] A. E. Ertan and T. P. Barnwell, “Improving the 2.4 Kb/s Military Standard MELP (MS-MELP) Coder Using Pitch-Synchronous Analysis and Synthesis Techniques,” in *Proc. of ICASSP*, vol. 1, pp. 761–764, IEEE, Mar. 2005.

[15] A. Pradhan, S. Chevireddy, K. Veezhinathan, and H. Murthy, “A low-bit rate segment vocoder using minimum residual energy criteria,” in *Proc. of NCC*, pp. 1–4, IEEE, Jan. 2010.

[16] M. Cernak, X. Na, and P. N. Garner, “Syllable-Based Pitch Encoding for Low Bit Rate Speech Coding with Recognition/Synthesis Architecture,” in *Proc. of Interspeech*, pp. 3449–3452, Aug. 2013.

[17] Y. Xu and Q. E. Wang, "Pitch targets and their realization: Evidence from Mandarin Chinese," *Speech Communication*, vol. 33, pp. 319–337, March 2001.

[18] ITU-T Rec. G.114, "One-way transmission time ," (Geneva, Switzerland) 2003.

[19] C.-C. Hsia, C.-H. Wu, and J.-Y. Wu, "Exploiting prosody hierarchy and dynamic features for pitch modeling and generation in hmm-based speech synthesis," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, pp. 1994–2003, Nov 2010.

[20] N. Dehak, P. Dumouchel, and P. Kenny, "Modeling prosodic features with joint factor analysis for speaker verification," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 15, no. 7, pp. 2095–2103, 2007.

[21] M. Cernak, A. Lazaridis, P. N. Garner, and P. Motlicek, "Stress and Accent Transmission In HMM-Based Syllable-Context Very Low Bit Rate Speech Coding," in *Proc. of Interspeech*, pp. 2799–2803, Sept. 2014.

[22] M. Cernak, P. Motlicek, and P. N. Garner, "On the (UN)importance of the contextual factors in HMM-based speech synthesis and coding," in *Proc. of ICASSP*, pp. 8140–8143, IEEE, May 2013.

[23] S. Bartlett, G. Kondrak, and C. Cherry, "On the Syllabification of Phonemes," in *Proceedings of NAACL*, (Stroudsburg, PA, USA), pp. 308–316, Association for Computational Linguistics, 2009.

[24] P. Mermelstein, "Automatic segmentation of speech into syllabic units," *J. Acoust. Soc. Am.*, vol. 4, no. 58, pp. 880–883, 1975.

[25] H. J. Giegerich, *English Phonology Introduction*. Cambridge University Press, 1992.

[26] A. McCree, "A Scalable Phonetic Vocoder Framework Using Joint Predictive Vector Quantization of Melp Parameters," in *Proc. of ICASSP*, vol. 1, p. I, IEEE, May 2006.

[27] A. McCree, K. Brady, and T. F. Quatieri, "Multisensor very lowbit rate speech coding using segment quantization," in *Proc. of ICASSP*, pp. 3997–4000, IEEE, Mar. 2008.

[28] R. Drullman, J. M. Festen, and R. Plomp, "Effect of temporal envelope smearing on speech reception.," *J. Acoust. Soc. Am.*, vol. 95, pp. 1053–1064, Feb. 1994.

[29] P. N. Garner and J. Dines, "Tracter: a lightweight dataflow framework," in *Proc. of Interspeech*, (Makuhari, Japan), pp. 1894–1897, 2010.

[30] C. Ris and S. Dupont, "Assessing local noise level estimation methods: Application to noise robust ASR," *Speech Communication*, vol. 34, pp. 141–158, April 2001.

[31] R. Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics," *IEEE Trans. on Speech and Audio Processing*, vol. 9, no. 5, pp. 504–512, 2001.

[32] H. Kawahara, H. Katayose, A. de Cheveigné, and R. D. Patterson, "Fixed point analysis of frequency to instantaneous frequency mapping for accurate estimation of F0 and periodicity," in *Proc. of Eurospeech*, (Budapest, Hungary), 1999.

[33] A. Black, P. Taylor, and R. Caley, "The Festival Speech Synthesis System," technical report, Human Communication Research Centre, University of Edinburgh, 1997.

[34] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. Black, and K. Tokuda, "The HMM-based Speech Synthesis System Version 2.0," in *Proc. of ISCA SSW6*, pp. 131–136, 2007.

[35] M. Astrinaki, O. Babacan, N. d'Alessandro, and T. Dutoit, "sHTS : A streaming architecture for statistical parametric speech synthesis," in *International Workshop on Performative Speech and Singing Synthesis*, 2011.

[36] M. Astrinaki, N. d'Alessandro, B. Picart, T. Drugman, and T. Dutoit, "Reactive and continuous control of HMM-based speech synthesis," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*, pp. 252–257, IEEE, Dec. 2012.

[37] X. Na, X. Xie, and J. Kuang, "Low latency parameter generation for real-time speech synthesis system," in *Proc. of ICME*, pp. 1–6, July 2014.

[38] D. B. Paul and J. M. Baker, “The design for the wall street journal-based CSR corpus,” in *Proceedings of the workshop on Speech and Natural Language*, HLT ’91, (Stroudsburg, PA, USA), pp. 357–362, Association for Computational Linguistics, 1992.

[39] K. Shinoda and T. Watanabe, “Acoustic modeling based on the MDL principle for speech recognition,” in *Proc. of Eurospeech*, pp. I –99–102, 1997.

[40] M. Mohri, F. Pereira, and M. Riley, “A rational design for a weighted finite-state transducer library,” in *Automata Implementation* (D. Wood and S. Yu, eds.), vol. 1436 of *Lecture Notes in Computer Science*, pp. 144–158, Springer Berlin Heidelberg, 1998.

[41] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY, USA: Wiley-Interscience, 1991.

[42] J. Kominek and A. Black, “The CMU Arctic speech databases,” in *Proc. of 5th ISCA Speech Synthesis Workshop*, pp. 223 – 224, 2004.

[43] J. Yamagishi, T. Kobayashi, Y. Nakano, K. Ogata, and J. Isogai, “Analysis of Speaker Adaptation Algorithms for HMM-Based Speech Synthesis and a Constrained SMAPLR Adaptation Algorithm,” *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 17, pp. 66–83, Jan. 2009.

[44] ITU-T Rec. P.800, “Methods for subjective determination of transmission quality,” (Geneva, Switzerland) 1996.

[45] V. Grancharov and W. . B. Kleijn, “Speech Quality Assessment,” in *Springer Handbook of Speech Processing* (J. Benesty, Sondhi, and Y. Huang, eds.), pp. 83–100, Springer Berlin Heidelberg, 2008.

[46] R. F. Kubichek, “Mel-cepstral distance measure for objective speech quality assessment,” in *Proc. of ICASSP*, vol. 1, pp. 125–128 vol.1, IEEE, May 1993.

[47] ITU-T Rec. P.862, “Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs,” (Geneva, Switzerland) 2005.

[48] ANSI, “Methods for Calculation of the Speech Intelligibility Index,” 1997.

[49] V. Leong, M. Kalashnikova, D. Burnham, and U. Goswami, “Infant-Directed Speech Enhances Temporal Rhythmic Structure in the Envelope,” in *Proc. of Interspeech*, pp. 2563–2567, Sept. 2014.