The Partly Preserved Natural Phases in the Concatenative Speech Synthesis Based on the Harmonic/Noise Approach

ZBYNĚK TYCHTL

Department of Cybernetics
University of West Bohemia
Univerzitní 8, 306 14 Pilsen
CZECH REPUBLIC

Abstract: This paper describes our advances in the development of the Czech TTS system achieved mainly in the field of speech signal generation. We achieved very high quality of the synthesized signal with our time-domain TTS system, but the speech unit database needs tens of megabytes. This is uncongenial, when we aspire to implement the high quality synthesis system at low-end embedded devices (handhelds, phones etc.). We found the approaches for speech representation based on sinusoidal coding [1] or harmonic plus noise modeling [2] respectively very promising for our goal. It is mainly due to high compression possibility of the spectral representation of the speech. The major inconvenience is the necessity of natural phase components to reach quality naturally sounding synthesis. Since there is no known method for suitable phase representation, the methods for its substitution must be searched. In our experiments, we observed the phase coherence to be more important (from the view of naturalness) then the necessity of the strict usage of the original phase component in all instants (frames). We proceed from this experience and here we propose our method where only the one phase vector is needed for each voiced segment (continuous sequence of voiced frames) in every speech unit.

Key-Words: speech signal synthesis, harmonic/noise, phase components

1 Introduction
For years, we develop concatenative TTS speech synthesis system [3] with huge statistically prepared triphone-based speech unit database. For speech signal generation the time-domain concatenative approach is applied. We achieve very high quality of synthesized speech signal, but the unit database needs about tens of megabytes of the store. This is uncongenial, when we aspire to implement the high quality synthesis system at low-end embedded devices (handhelds, phones etc.). We found the sinusoidal coding [1] or harmonic plus noise modeling [2] techniques very promising in our goal of reaching relatively high quality of synthesized speech and simultaneously having possibility to compress well the speech unit database.

In our effort to build the high quality high-end (without restrictions to the computational power and the storage space) speech synthesis system, we also tried different approaches for speech signal generation. We tried approaches other than the time-domain e.g. LPC and residual excited LPC (called RELP). From model-based approaches, we have anticipated the capability to smooth spectral transitions between concatenated units via model parameters. We found all these methods producing number of artifacts, which degraded resulting synthesized speech to unacceptable level. On the other hand, all these model-based methods would be useful for efficient speech unit representation, which we would appreciate in development of the version of synthesis system for the embedded devices. Unfortunately, we did not found those approaches achieve satisfying quality.

In conjunction with our high-end time-domain system we also tried [4] the approach similar to MBROLA [5] where we off-line re-synthesized speech unit database to the constant preset pitch-frequency. From the utilization of the frequency-domain method for the re-synthesis with modification of the pitch-frequency, we expected achieving the high quality constant-pitch unit database free of artifacts that are usually obtained using time-domain pitch-modification. We performed several variations of this approach. For example, we tried to interpolate - beside other
common parameter like pitch and spectral amplitudes, the spectral phases. We tried to use zeroed phases, minimal phases, constant phases, partly randomized phases as well as some combinations of these approaches. Regardless of promising results of informal listening tests of re-synthesized speech we observed higher number of disruptive artifacts in a final synthesized speech using our time-domain system with so modified unit database.

Beside our push on high-end speech synthesis system we still go after the synthesis system suitable for low-end embedded devices. But we still aspire to reach the high quality naturally sounding synthetic speech. After our preliminary tests of HNM based approach [2], we found it to be capable to produce high quality synthetic speech as well. But it must be said that the level of reached quality is strongly constrained by the quality of the speech unit database. We found this method to be quite sensitive to accurate determination of the pitch-frequency and the placement of the phonetic unit boundaries. It is also necessary to ensure the coherence of phase components during synthesis stage that is not generally easy task. Stylianou in [2] offers the method based on the gravity of speech signals for the phase mismatches removal in a way of shifting them relatively around the center of gravity. It acts like a substitution of a demand of analyzing the signals synchronously with glottal closure instants. After all, due to our big effort continuously pursued to speech unit database development, we need neither pitch-frequency refinement nor phase correction by signal shifting. We have professionally recorded speech corpus with use of electroglottograph to also record the glottal signal in which we successfully detected the glottal closure instants (pitch-marks). Let it be mentioned, that for the unvoiced segment of the speech we defined pitch-mark-like instants equally spaced with rate of 6 ms to help us to process the speech units pitch-synchronously.

Then the speech unit database was created from the corpus employing the HMM-based automatic segmentation.

We can also use the module for the generation of the synthetic prosodic parameters.

2 The base-lines

Let us briefly summarize our initial conditions that we can build on due to extensive effort pursued in the development of our high-end time-domain Czech TTS synthesis system [3].

We have a high quality speech corpus recorded by a professional speaker. The speaker was asked to try to speak monotonously. Whole corpus was checked by listeners and disposed of insufficient records. Using the electroglottograph we recorded the glottal signal in which we successfully detected the glottal closure instants (pitch-marks). Let it be mentioned, that for the unvoiced segment of the speech we defined pitch-mark-like instants equally spaced with rate of 6 ms to help us to process the speech units pitch-synchronously.

Then the speech unit database was created from the corpus employing the HMM-based automatic segmentation.

We can also use the module for the generation of the synthetic prosodic parameters.

3 Analysis stage

By the term ‘analysis stage’ we denote the off-line process of yielding the parameters of the harmonic and/or noise parts of all speech units from the basic speech unit database.

It remains unsaid that we often use the term ‘speech unit database’ without explicit designation, which one is particularly meant. Let us mention here that we initially start with speech unit database built using automatic HMM-based approach for our time-domain high-end synthesis system. During the analysis stage another database is built just by consequent unit-by-unit analysis of the mentioned initial database for the purpose of yielding harmonic.
and noise features that are stored in the new database.

3.1 Unvoiced segments

By the unvoiced segments we denote uninterrupted sequences of frames in speech unit that are marked as unvoiced. We analyses such segments by the well-known LPC method. For the LPC analysis we use window of the length about 10 ms and it is shifted by the frame rate of 6 ms. We also estimate the speech signal variance every 2 ms in the frame to improve correctness in a modeling of the short noisy sounds like plosives. For every unvoiced frame we estimate 10 LPC coefficients and 3 (each for 2 ms of signal) variances.

3.2 Voiced segments

In [2], it is assumed that voiced speech segment $s$ can be modeled by the sum of two components. The first one models ‘voiced’ (harmonic) part of the signal and the other one models the ‘noise’ part:

$$s = s_h + s_n,$$  (1)

where $s_h$ denotes the harmonic part and $s_n$ denotes the noise part. Those two parts are also assumed to be separated in the frequency domain by the boundary in the frequency band. The boundary (and consequently the number $L$ of harmonics) can be well determined using the approach described in [2]. There it is determined in every analyzed frame separately. In each frame the maximal voiced frequency $F_{\text{max}}$ is determined and the frequency band up to this boundary is marked as a voiced part and the rest of whole frequency band is marked as unvoiced part.

In the context of the method and experiments proposed in this paper we considered the $F_{\text{max}}$ constant for all the frames in all units in the unit database. We did it just for the interim simplification of the implementation and simpler description.

3.2.1 Voiced parts of the voiced segments

The voiced part is modeled as sum of harmonics

$$s_h = \sum_{k=-L}^{L} A_k(t) e^{j\omega_h(t)t}.$$  (2)

where $L$ denotes the number of harmonics and $\omega_h$ denotes the fundamental frequency (pitch-frequency).

There were proposed several approaches, which differ in the way of estimation of amplitude factors $A_k$. In [2], three different models are mentioned. They differ in assumption that amplitudes in one frame have constant, linear or quadratic time dependence. It was declared, and we have experimentally confirmed it, that the simplest approach with constant amplitudes in the frame is sufficient.

In task of estimation of amplitudes we adopted method published in [1], that is computationally simpler then the one in [2]. It is based on a harmonic sampling of STFT (Short-Time Fourier Transform) of the analyzed speech frame. To obtain reasonable estimates of amplitudes using mentioned method, it is necessary to guarantee the quality of SFTF analysis by following several important rules.

The width and placement of the analysis window is very important. We confirm that the window needs be at least two local pitch-periods long. Rather a bit longer (but not too much) than shorter. Since we have well positioned pitch-marks in the speech units, we adaptively modify the actual analysis window width. Since the analysis window may not be long enough to offer higher frequency resolution in STFT, we use the FFT of quite higher length. We use 8192-point FFT, when we analyze speech sampled with $F_s=16$ kHz. It offers frequency resolution less that 2 Hz. Relative window placement in a frame is also driven using our pitch-marks. The window is always centered at the pitch-mark. Since we can rely on the correctness of our pit-marks, we have ensured (using pitch-synchronous window placement) the phase coherence in the voiced speech units. It is very important in concatenative speech synthesis to ensure the phase coherence in successive synthesized units. Let us mention, that in our approach this issue is not so much important. It is due to the fact, that in the synthesis stage we use just one phase component for each whole voiced segment of the synthesized speech. Regardless that, we confirm that it is still needful to position the analysis windows pitch-synchronously and centered at the pitch-marks to yield the suitable spectral estimate using FFT. Let it be mentioned, that we do this ‘substitution’ of phases by just one phase vector with intent to omit huge amount of phase data in the task of storing the speech unit database. Before description of phase vector construction, let us
formulate how we estimate the amplitude. The \( i \)-th element \( a_i \) of amplitude vector is obtained from STFT as

\[
a_i = \frac{2X \left( \sum_l w(l) \right)}{X},
\]

where \( X \) denotes STFT, \( \omega_0 \) is local fundamental frequency estimated from the local pitch-marks distance, \( w \) is the analysis weighting window, \( N \) is the number of bins of FFT and \( F_S \) is sampling frequency. The \( \text{round()} \) function rounds the argument to get the FFT bin nearest to the \( i \)-th \( \omega_0 \) harmonic. Due to the mentioned fact that we use quite long FFT, we reach the frequency error in the spectral sampling less than 2 Hz. It is certainly possible to use even longer FFT, but it would be useless since the spectral resolution would be much smaller that the error in a local \( F_0 \) estimation.

The phase components are also simply extracted from appropriate bin of the FFT output, but it has already been indicated that not all of them are considered to be stored in ‘small version’ of speech unit database. We propose, that just one phase component vector is stored for every voiced segment in the speech unit. It remains unanswered, how this vector is chosen and designed. This vector is not a simple copy of one of phase vectors yielded by FFT analysis. The reason for this is that, as we stated \( F_{\text{max}} \) be for simplicity constant, the number of elements of phase vectors vary dependently on the local fundamental frequency. It is certainly the same effect that occurs with vectors of amplitudes. The number of the vector elements that are obtain analyzing the \( k \)-th speech frame is

\[
L^k = \frac{F_{\text{max}}}{F_0^k}
\]

where \( F_0^k \) denotes the local fundamental frequency in the frame. We choose just one phase vector to be the basic representative of the phase component for the whole voiced segment in every speech unit. It makes sense to choose the frame for representative determination in the most spectrally stable area of the analyzed segment. For this purpose we evaluated a criterion giving a squared measure of inter-frame spectral differences in frequency band up to 2 kHz. We also tried other upper spectral boundaries but we found out that it is not necessary and that it is suitable just to pick up the frame right in the middle of the segment.

If one of phase vectors is chosen to be the basic representative of the phase component in speech segment it means it supplies just its \( L^k \) elements. If we would, later in synthesis stage, seek to use just this concrete phase vector, we could not synthesize the signal with lower fundamental frequency than

\[
F_{0}^{k} = \frac{F_{\text{max}}}{L^k}.
\]

So we need, by some way, to ‘extend’ the phase vector. For this purpose we perform following procedure. Starting at the frame where the basic representative was chosen we search frame-by-frame the voiced segment in the unit for lower fundamental frequencies. If the lower fundamental frequency is found in the consequent frame then the phase vector elements with higher indexes than \( L^k \) are appended to the basic representative. Such a way the procedure continues until the lowest fundamental frequency in voiced segment is found and phase vector representative is maximally extended. It is certainly not guarantied to found the frame in the unit with fundamental frequency low ‘enough’. In practice we define a global limit for the lowest fundamental frequency \( F_{0\text{min}} \) that can be synthesized. It is global for whole speech unit database and it simply constrains the prosody generation module. Now, it is clear that it is necessary to build every phase representative vector up to \( L_{\text{GLOB}} = \frac{F_{\text{max}}}{F_{0\text{min}}} \) vector elements. Since it is quite common that it is not satisfied searching just in the context of the originating frame of the representative, we extend the search to the other speech segments (unit-like) that were not included in speech unit database but they also represent the same phonetic unit. Let us mention here that since our synthesis system uses the triphones as the phonetic units, the speech units in a database are relatively short and they mostly contain only one voiced segment. So it is not complicated to find the corresponding voiced segment in the speech segment related to the particular phonetic unit. Even, regardless all these procedures, it happens in some cases, that we do not yield required number of phase vector elements. In those cases we have simply randomized those highest missing elements. To evaluate the influence of the randomization we forced synthesizer to produce speech with fundamental frequency lower than \( F_{0\text{min}} \) that was preset for unit database creation. Although we performed only subjective informal listening test we found out that it is difficult to identify whether the perceived unnaturalness in low-frequency parts are
caused by those several randomized phases. If we try to probe it by continuing in lowering $F_{0,\text{max}}$ in unit database creation we incline to synthesis system with random phases with expectable declination of the naturalness of synthesized speech.

Let it be mentioned, that using the described approach for phase representative vector construction by appending extra elements to its end we do not change the vector elements assignment to the particular frequencies. It also does not change $F_{\text{max}}$ by any way. In fact, there is no “fixed” assignment of the phase vector elements to frequency points. Simply said, every chosen number of phase vector elements is always assigned exactly to whole frequency band $0 \div F_{\text{max}}$.

### 3.2.2 Unvoiced part of the voiced segments

The analysis of the unvoiced part is performed practically the same way as in [2]. Although, we have the harmonic/noise boundary preset globally for all analyzed frame, we use it by the same way. From the vectors of amplitudes and phases the voiced part $s_h$ is synthesized. Then it is subtracted from the original speech signal to obtain the noise part $s_n$ and it is then LPC analyzed yielding LPC filter coefficients that are to be stored. In the noise part $s_n$ we also determine its energy time-evolution by measuring its variance every 2 ms by the same way as in the unvoiced segments.

### 4 Synthesis stage

The speech signal synthesis performed frame-by-frame using well-known pitch-synchronous approach. With the use of generated prosodic information ($F_0$ contour, durations and volume contour) all successive frames of the sizes of one local synthetic pitch-period are generated.

The unvoiced frames are generated by filtering a unit-variance white noise by normalized (in gain) LPC filter. The coefficients of the filter are changed with frame rate that was preset in the analysis stage to constant rate 6 ms in the unvoiced segments. The output of the filter is weighted by noise variances (see chapter 3). In the unvoiced units we do not perform any interpolation of the LPC coefficients. On the contrary we try to disengage the noise variance contour from discontinuities at concatenations by linear weighting. The tilt of the weighting is determined by the mean values of variance contours in left and right concatenated units.

The noise part $s_n$ is generated the same way also in voiced frames. It differs just by post-filtering with high-pass filter with cut-off frequency set to $F_{\text{max}}$.

#### 4.1 Amplitudes

To synthesize the voiced part of the voiced segments we employ straightly (2) where just instead of complex exponential we use sine-wave functions multiplied by synthetic amplitudes $a_i$ that are determined by simple re-sampling the spectral envelope formed by analytic amplitudes $a_i$ from (3). The amplitudes are the subjects of linear smoothing over the concatenation point of successive units.

#### 4.2 Phases

The phases for the sine-wave functions are obtained from the stored phase representative vector. At the start of generation of the voiced segment every element in the vector of amplitudes is coupled with the element from phase representative vector at same position in the vector.

In the successive frames ($k$) the phase $\phi^k_i$ of each harmonic component ($i$) (each sine-wave) is copied from preceding synthesized frame using following rule. If $F_{0S}^k < F_{0S}^{k-1}$ (synthetic $F_0$ decreases; S in subscript denotes ‘synthetic’) then for $\phi^k_i$ the phase of the nearest “higher” (at higher frequency) component from preceding frame is used. If $F_{0S}^k > F_{0S}^{k-1}$ (synthetic $F_0$ increase) then the phase of the nearest “lower” (at lower frequency) component from preceding frame is used.

Let’s follow the consequences on the example, where synthetic $F_{0S}$ slowly vary from 200 Hz at beginning of the voiced segment in synthesized unit to 100 Hz at its end. The synthetic phase component $\phi^k_i$ which was at the start assigned to the harmonic component at 200 Hz is at the end used at frequency 100 Hz. Generally the phase component $\phi^k_i$ initially assigned at beginning to component at frequency $F_i$ is at the end of segment assigned to component at frequency $F_i = \alpha F_j$. Coefficient $\alpha$ corresponds to variable prosodic parameter driving the requirement on synthetic $F_{0S}$ contour. So as the synthetic $F_{0S}$ varies during voiced segment synthesis, the assignment of the phase vector elements is accordingly shifting across the frequencies. If $F_{0S}$ increases during voiced segment synthesis, the phase vector elements seems being moving upward the frequency axis as are assigned to component at higher frequencies in consecutive
frames. So as it causes decrease of number of harmonic components being synthesized (only those less than $F_{\text{max}}$ are used) the number of phase vector elements used decreases. In an opposite case when $F_0S$ decreases during segment synthesis, the number of harmonic components being synthesized increases. It means that more phase vector elements is being used in the same constant frequency interval $0\div F_{\text{max}}$.

To avoid the absence of mandatory phase vector elements we perform the technique (described in chapter 3.2.1) that extends the phase representative vector and constrains the global minimal fundamental frequency.

5 Conclusions
In this paper we present our development towards the high quality speech synthesis based on harmonic/noise (or sinusoidal and noise) speech representation.

We offer the method that determines just one phase vector for every whole voiced segment in speech unit. So instead of storing one phase vector for every voiced frame (of the length of one pitch period) we store just one vector (called phase representative vector) for whole sequence of voiced frames in speech unit into the unit database. Since mostly the voiced speech unit (that represents phonetic unit – triphone) contains just one voiced sequence (segment) of voiced frames, we store to speech unit database just the number of phase vectors that is comparable to the number of units in database.

Informal subjective listening tests confirm that the keeping of the phase coherence across the concatenation of voiced units (under the conditions of changing prosodic parameters) is perceptually more important than the fact that phase component obtained from one phonetic unit is being used in the other (following) phonetic unit.

The use of this approach that uses natural phase component gives better results than using just zeroed, minimal or other completely artificial phases. Moreover, amount of data that must be kept in database is highly reduced.

6 Acknowledgements
This research was supported by the Grant Agency of Czech Republic, project No. GAČR 102/02/0124 and by the Ministry of Education of Czech Republic, project No. MSM 235200004.

7 References