TIME-DOMAIN SYNTHESIZER FOR PRESERVING MICROPROSODY

Reinhard Posmyk

Lehrstuhl für allgemeine Elektrotechnik und Akustik,
Ruhr-Universität, Postfach 10248, D-4630 Bochum 1, FRG

ABSTRACT

In this paper a new technique for coding speech signals for the use in high quality text-to-speech environment is presented. A prosodic control of the acoustic output is implemented merely by modifying the speech waveform during the resynthesis phase. Any degradation in sound quality is avoided, since no parametric representation of the speech signal is used for the storage of speech. This algorithm preserves the intrinsic features of the original speech signal (microprosody) while allowing for a prosodic control with only few values per time. The algorithm requires very small computational equipment for the synthesis.

INTRODUCTION

When looking at high performance text-to-speech systems - in terms of intelligibility and naturalness - two different principles for the production of acoustic speech output gained considerable commercial significance [1]: The first one is the formant synthesis, as it is used, e.g., in the DECtalk system (former Klattalk). For the generation of speech that sounds natural, it is necessary to control lots of parameters by an elaborated rule-system. Thus, very detailed knowledge about those cues in human speech that are important for perception as well as about the dynamic processes of human speech production is needed.

The second kind of synthesizers uses stored segments of natural speech and concatenates them to new phrases. The segments are usually diphones or demi-syllables that are stored in a parametric representation, e.g., LPC coefficients, to give the opportunity to resynthesize them with different prosodic values. Even so, LPC-based synthesizers need significant computational power to implement the synthesis filters. Additional synthesizers that implement an articulation model of the human vocal tract have also been developed. But these models even more suffer from a lack of knowledge about the physiology of speaking.

This review shows that the generation of high-quality speech requires a complex equipment for the synthesis. These synthesizers also need comprehensive control information for pitch contour, because even for those synthesizers that use speech segments from natural speech all inherent data for this parameter, known as microprosody, is lost in the preceding stage of coding and has to be recalculated from the phonetic representation of speech. And, although the quality of the best of these synthesizers is quite good for male voices, problems remain for the production of female voices, since all synthesizers use a parametric description of the speech signal, where the set of considered parameters seems to be incomplete, at least as far as naturalness is concerned.

Starting off from the fact that all perceptual relevant information is included in the time signal of a human utterance, it seems reasonable to build a speech synthesizer that uses the modified time signal of natural speech segments directly as elements for the synthesis. These speech segments can be diphones, demi-syllables or even larger units like complete words. To balance the requirements for a flexible and unrestricted vocabulary and for high naturalness probably a hybrid definition for the size of units is the best solution. The output of those modified and concatenated natural speech segments, of course, does not result in a high-quality sound, since the beginnings and endings of the concatenated segments do not fit exactly in fundamental frequency, in sound quality, and in signal energy. Differences in the spectral properties, however, can be minimized by a suitable choice of segments. After an alignment of the beginnings and the endings of the segments the envelope of the signal is modified, so that the signal energy of connectable segments is the same at the segment boundaries. But even if the problem of discontinuity of the fundamental frequency may easily be solved, such a synthesizer cannot unrestrictedly be used in a text-to-speech environment. The static elements for synthesis prevent the control of prosodic parameters like pitch patterns or sound durations.

PREPARATION OF SPEECH SEGMENTS

In 1981 Großmann introduced a time domain synthesizer that uses specially prepared diphone elements [2]. To allow for the control of the fundamental frequency several similar versions of each segment that only differ in fundamental frequency and that are derived from a standard segment by LPC analysis and synthesis have to be stored. By this method, only very coarse and insufficient prosodic control was achievable.
Provided that the human auditory system does not exactly analyse the phase relations of spectral components of a single speech signal but rather their absolute distribution, the waveform of the signal can be modified considerably without a loss of quality. With respect to a single period of a voiced sound – in the following text denoted as ‘frame’ – the waveform can be tuned in the way that most of the energy is concentrated in its time center. Mozer [3] introduced a technique that uses this property of time signals to build a speech synthesizer for a restricted vocabulary using large segments like words and syllables. The main objective was the reduction of data rates to achieve the integration of a synthesizer on a single chip. The Mozer coding algorithm consists of several modules that compress and encode natural speech data at rates of some hundred bits per second. The core of this technique is the fact that each period of the time signal is rebuilt from the Fourier coefficients of the original signal in the way that the phase angles of these components are either 0 or 180 degrees. In the case that the speech signal can be approximated as a stationary process, where the waveform is repeated infinitely, the new waveform still has the same magnitude spectrum in the frequency domain as the original one. The process of calculating an optimal set of phase angles tends to be extremely time-consuming, since the signal energy must be minimized at the borders in order to allow for a concatenation of neighbouring periods. For the Japanese language for its application in a text-to-speech system with unrestricted vocabulary, a similar technique has been developed by Yazu and Yamada [4]. It only uses spectral components with phase angles of 0 degrees. The centering of signal energy is done by simply applying a trapezoidal window to the resulting waveform.

An alternative synthetic time signal is used in this approach for a time-domain synthesizer. It is built from the Fourier components that have phase angles of +/- 90 degrees. By overlaying such components the task of constructing a set of optimal phase angles becomes much easier, because the signal energy is automatically minimized at the borders, not only the signal energy must be minimized. The waveform should also have a minimal slope, since the slope of the neighbouring period is completely unknown. This implies that the control of the fundamental frequency will be possible either by inserting zero sequences or by partially merging consecutive periods without the need to apply any windowing [5]. A set of optimal phase angles can be found according to the following properties: The time series of a periodic signal \( x(t) \) is built up from the Fourier coefficients \( X(k) \) by

\[
x(t) = \text{Re} \left\{ \sum_{k=0}^{N} X(k) e^{\frac{2\pi i k t}{N}} \right\}
\]  

(1)

In order to have a minimal slope for \( t=0 \) and \( t=\frac{N}{2} \), the first derivative of (1) must be minimized:

\[
x'(t) = 2\pi \sum_{k=0}^{N} X(k) s(k) \sin \frac{2\pi k t}{N}
\]

(2)

From \( \lim |x'(t)| = 0 \) and with \( s(k) \) being either +1 or -1 corresponding to a phase angle of -90 or +90 degrees for the Fourier component \( k \), it follows

\[
\sum_{k=0}^{N} k |X(k)| s(k) \leq \text{min.}
\]

(3)

This equation probably may be solved by several combinations of \( s(k) \), but even if there is only one optimal combination, it is difficult to find it. In a practical implementation equation (3) is solved in the following way: Starting off with the magnitude of the Fourier coefficient that corresponds to the lowest frequency \( X(1) \) – which gets a positive sign \( s(1) = +1 \) – the \( s(k) \) of the following Fourier coefficients are chosen, so that for every \( k \) the sum becomes minimal. The fact that the phase angles of the dominating Fourier component itself, does not change rapidly on consecutive periods is a very useful property of the resulting waveform; if that were not the case, it would lead to audible distortions.

This coding technique is applied to unvoiced sounds with some modifications. Since the waveform is not periodic, an arbitrary interval is chosen to meet the above mentioned definition. In contrast to voiced sounds, where rapid changes in phase angles must be avoided, it is desirable to have a random distribution of phase angles in successive frames for unvoiced sounds. Otherwise periodic components would result in a kind of voicing.

THE CONTROL OF THE FUNDAMENTAL FREQUENCY

As stated beforehand, the time axis cannot be modified in any way in order to preserve the original formant positions in the frequency spectrum. Thus, fundamental frequency control must be done by shifting successive frames in their relative position. If the fundamental frequency for reproduction is lower than that of the original, the period is lengthened by appending zero samples to the end. To increase fundamental frequency compared to the original, successive periods are shifted towards each other, and the samples in the common range are

\[
x'(t) = 2\pi \sum_{k=0}^{N} X(k) s(k) \sin \frac{2\pi k t}{N}
\]

(2)

\[
\sum_{k=0}^{N} k |X(k)| s(k) \leq \text{min.}
\]

(3)

This equation probably may be solved by several combinations of \( s(k) \), but even if there is only one optimal combination, it is difficult to find it. In a practical implementation equation (3) is solved in the following way: Starting off with the magnitude of the Fourier coefficient that corresponds to the lowest frequency \( X(1) \) – which gets a positive sign \( s(1) = +1 \) – the \( s(k) \) of the following Fourier coefficients are chosen, so that for every \( k \) the sum becomes minimal. The fact that the phase angles of the dominating Fourier component itself, does not change rapidly on consecutive periods is a very useful property of the resulting waveform; if that were not the case, it would lead to audible distortions.

This coding technique is applied to unvoiced sounds with some modifications. Since the waveform is not periodic, an arbitrary interval is chosen to meet the above mentioned definition. In contrast to voiced sounds, where rapid changes in phase angles must be avoided, it is desirable to have a random distribution of phase angles in successive frames for unvoiced sounds. Otherwise periodic components would result in a kind of voicing.

\[
x'(t) = 2\pi \sum_{k=0}^{N} X(k) s(k) \sin \frac{2\pi k t}{N}
\]

(2)

\[
\sum_{k=0}^{N} k |X(k)| s(k) \leq \text{min.}
\]

(3)

This equation probably may be solved by several combinations of \( s(k) \), but even if there is only one optimal combination, it is difficult to find it. In a practical implementation equation (3) is solved in the following way: Starting off with the magnitude of the Fourier coefficient that corresponds to the lowest frequency \( X(1) \) – which gets a positive sign \( s(1) = +1 \) – the \( s(k) \) of the following Fourier coefficients are chosen, so that for every \( k \) the sum becomes minimal. The fact that the phase angles of the dominating Fourier component itself, does not change rapidly on consecutive periods is a very useful property of the resulting waveform; if that were not the case, it would lead to audible distortions.

This coding technique is applied to unvoiced sounds with some modifications. Since the waveform is not periodic, an arbitrary interval is chosen to meet the above mentioned definition. In contrast to voiced sounds, where rapid changes in phase angles must be avoided, it is desirable to have a random distribution of phase angles in successive frames for unvoiced sounds. Otherwise periodic components would result in a kind of voicing.

\[
x'(t) = 2\pi \sum_{k=0}^{N} X(k) s(k) \sin \frac{2\pi k t}{N}
\]

(2)

\[
\sum_{k=0}^{N} k |X(k)| s(k) \leq \text{min.}
\]

(3)

This equation probably may be solved by several combinations of \( s(k) \), but even if there is only one optimal combination, it is difficult to find it. In a practical implementation equation (3) is solved in the following way: Starting off with the magnitude of the Fourier coefficient that corresponds to the lowest frequency \( X(1) \) – which gets a positive sign \( s(1) = +1 \) – the \( s(k) \) of the following Fourier coefficients are chosen, so that for every \( k \) the sum becomes minimal. The fact that the phase angles of the dominating Fourier component itself, does not change rapidly on consecutive periods is a very useful property of the resulting waveform; if that were not the case, it would lead to audible distortions.

This coding technique is applied to unvoiced sounds with some modifications. Since the waveform is not periodic, an arbitrary interval is chosen to meet the above mentioned definition. In contrast to voiced sounds, where rapid changes in phase angles must be avoided, it is desirable to have a random distribution of phase angles in successive frames for unvoiced sounds. Otherwise periodic components would result in a kind of voicing.

\[
x'(t) = 2\pi \sum_{k=0}^{N} X(k) s(k) \sin \frac{2\pi k t}{N}
\]

(2)

\[
\sum_{k=0}^{N} k |X(k)| s(k) \leq \text{min.}
\]

(3)

This equation probably may be solved by several combinations of \( s(k) \), but even if there is only one optimal combination, it is difficult to find it. In a practical implementation equation (3) is solved in the following way: Starting off with the magnitude of the Fourier coefficient that corresponds to the lowest frequency \( X(1) \) – which gets a positive sign \( s(1) = +1 \) – the \( s(k) \) of the following Fourier coefficients are chosen, so that for every \( k \) the sum becomes minimal. The fact that the phase angles of the dominating Fourier component itself, does not change rapidly on consecutive periods is a very useful property of the resulting waveform; if that were not the case, it would lead to audible distortions.
Fig. 2: Waveform after synthesis (f).

Fig. 3: Waveform after synthesis (0.7 f).

Fig. 4: Waveform after synthesis (1.4 f).

Fig. 5: Preparation of sound data

THE IMPLEMENTATION OF A SYNTHESIZER

The preparation of the data, as used for synthesis, is rather time-consuming. Fig. 5 shows this process. The corresponding speech segments are extracted from nonsense words, in which the segments appear in a neutral context. This approach avoids context specific coarticulations. Those segments which include a vowel are spoken both in a stressed and an unstressed position. In order to be able to also synthesize words that are not German, additional sounds, mainly English ones, are included as well. Next, frame markers are assigned to the segments. Each frame is transformed with a preemphasis in the frequency domain, and phase angles are aligned as described above. A spectral distance to the preceding frame - if it is available - is calculated to decide whether the frame can be skipped.
or not. Inverse FFT is applied with a frame size that has the same length as the original one. Finally, the resulting data are compressed and stored with additional information about the frame type (voiced/unvoiced) and the percentage of the total size of the segment. This information is used for synthesis to decide which frame is to be repeated in which way.

As well as for Moser coding for this synthesis it is sufficient to store the samples of one half of the modified signal for the reconstruction of the waveform. The second part of the waveform is here the antisymmetric equivalence of the first one. However, if the fundamental frequency of the original frame matches that one which is required for reproduction, the synthesis is simply done by reading the samples from the memory at first in the normal order and then in reverse order with an inversion of each sample. A further compression of the data is done by an 8-bit logarithmic quantization. An additional reduction for the storage requirement is achievable by more sophisticated coding techniques like ADPCM. During the whole process the segment inherent parameters for the frame size (pitch) is preserved.

Fig. 6: Architecture of the synthesizer

For the synthesis (Fig. 6) a phonetic description for the orthographic input text is calculated by the TTS system SYRUB [5] that produces coarse information for allophone code, duration, fundamental frequency, and sound intensity. Successive allophone codes are parsed to match the largest and best fitting segment in the speech data file. In this step, stressed sounds are distinguished from unstressed ones. The calculated prosodic parameters are treated as an offset to the values that are inherent in the sound data. The samples of each frame are read from the memory, expanded, modified in amplitude, then modified in amplitude, and shifted according to the calculated fundamental frequency. The data are converted by a DAC and integrated by an analog low-pass filter.

CONCLUSIONS

The algorithm for time-domain synthesis presented in this paper is able to produce speech with a good quality, especially with respect to intrinsic features of the original human speech sound that is used for synthesis. The advantage of this approach is the minimal hardware requirement with the elimination of special purpose chips for the synthesizer that only needs some more data storage than other synthesizers.