

# ON THE IMPLEMENTATION OF THE HARMONIC PLUS NOISE MODEL FOR CONCATENATIVE SPEECH SYNTHESIS

Yannis Stylianou

AT&T Labs-Research, SIPS, Shannon Laboratories, 180 Park Avenue, Florham Park, NJ 07932  
email : yannis@research.att.com

## ABSTRACT

In concatenative speech synthesis systems, speech models are usually used to represent the speech signal. Recently, the Harmonic plus Noise Model, HNM, has been proposed for concatenative speech synthesis with promising results. One main drawback of HNM is its complexity. In this paper, we review four different methods of reducing the complexity of HNM. These include, straight-forward synthesis, synthesis using inverse Fast Fourier Transform, synthesis using Recurrence Relations for trigonometric functions, RR, and synthesis based on Delayed Multi-Resampled Cosine functions, DMRC. DMRC was shown to outperform all the other techniques reducing the complexity of HNM synthesizer by 95% compared to the current version of the HNM which is based on the SF method. Informal listening tests showed that the version of HNM based on the DMRC method provides higher quality of speech synthesis than the version based on SF.

## 1. INTRODUCTION

In the context of speech synthesis based on concatenation of acoustic units, speech signals may be encoded by speech models. These models are required to compress the speech database and to perform prosodic modifications where necessary and, finally, to ensure that the concatenation of selected acoustic units results in a smooth transition from one acoustic unit to the next.

There are various methods for concatenating acoustic units. LPC-based methods such as impulse driven LPC and Residual Excited LP (RELPC) have been proposed for speech coding as well as for speech synthesis [1]. TD-PSOLA [2] is one of the most popular concatenation methods. It performs a pitch-synchronous “analysis” and synthesis of speech. MBROLA [3] is based on the MBE speech coder [4] and it may be viewed as a modified TD-PSOLA method. In MBROLA, the voiced parts of the speech database are resynthesized with constant phase and constant pitch avoiding, therefore, concatenation problems during synthesis. Sinusoidal models have been proposed also for synthesis [5] [6]. The Harmonic plus Noise Model, HNM [7], is part of the family of sinusoidal models proposed for speech synthesis. In the context of HNM, speech signals are represented as a time-varying harmonic component plus a modulated noise component. The decomposition of the speech signal into these two components allows for more natural-sounding modifications (e.g., source and filter modifications) of the signal [8]. Also, the parametric representation of speech using HNM provides a straightforward way of smoothing discontinuities of acoustic units around concatenation points. Formal listening tests have shown that HNM provides high-quality speech synthesis while outperforming other models for synthe-

sis (e.g., TD-PSOLA) in intelligibility, naturalness and pleasantness [9].

Although HNM was found to be a very good candidate for the Next Generation TTS of AT&T, the main drawback of HNM remains its complexity. High complexity is an important disadvantage in real applications of a TTS system where we need to run as many channels as possible on commonly available hardware. The main task of the HNM module during synthesis is to load HNM parameters from the HNM database, modify these parameters when necessary (pitch scale modification requires re-estimation of harmonic amplitudes and phases), smooth transition points and finally generate the speech signal as a sum of a number of harmonics and of modulated noise. The noise part is produced by filtering Gaussian noise through an AR filter and multiplying the output by a time domain envelope. Prior to this addition, a high-pass filter removes the low frequency component from the noise part. The most expensive operation during HNM synthesis is the generation of the speech signal. How can we reduce this complexity? First, we consider that the noise part may be generated as a sum of harmonics with random phase [10], thus avoiding the use of high-pass filters during synthesis. Then the following question is asked: *what is the fastest way to generate and add  $K$  harmonics?*, where  $K$  may be a big number (limited, however, by half of the sampling frequency of the input speech signal and its fundamental frequency).

In this paper, we review four different methods. The first method is the Straight-Forward, SF, sum of these harmonics (SF method is mainly an inverse DFT). This is the method currently used in the HNM synthesis module. The second method is the use of the Inverse Fast Fourier Transform (IFFT method). The third method makes use of Recurrence Relations for trigonometric functions (RR method). Finally, the fourth method is based on the transformation of the phase spectrum into phase delays and then generate the speech signal as a sum of Delayed Multi-Resampled Cosine functions (DMRC method).

While the first thought is that the IFFT method would be the answer to the above question, in this paper, we will show that the third and fourth methods run much faster than the IFFT method. From these two last methods, the fourth method is by far the fastest. The transformation of the phase spectrum to the phase delay domain provides additional advantages. Preliminary results show that this is a simple way to code the phase, because phase delays are less sensitive to quantization errors than the phase spectrum (actually, the problem of an efficient coding of the phase information is not solved yet for the sinusoidal coders). In addition, this provides an easy way to re-estimate the phase delays of the modified harmonics during pitch modifications.

This paper is organized as follows. A brief description of

HNM for modeling the speech signal and the application of HNM in speech synthesis is given in Section 2. In Section 3 we present the four different methods to synthesize a harmonic signal and in Section 4 the methods are compared in terms of complexity and Signal to Noise Ratio, SNR. SNR is computed from the original speech signal and the synthesized signals obtained by these four methods. Section 4 also presents results from an informal listening test comparing the current version of HNM (using the SF method and the synthesis of the noise part using a filtering process) with a new version of HNM where the DMRC method was used. The Section discusses some other advantages of using phase delays instead of using the phase spectrum for the compression of a speech database and for speech modifications. Finally, conclusions and plans for future work are given in Section 5.

## 2. HNM FOR SPEECH SYNTHESIS

In this section, we present a brief description of HNM for the analysis and synthesis of speech. For a detailed description of HNM for synthesis, see [7]. HNM is based on a harmonic plus noise representation of the speech signal. The harmonic part accounts for the quasi-periodic component of the speech signal; the noise part models its non-periodic components, which include friction noise and period-to-period variations of the glottal excitation.

The spectrum is divided into two bands. The time-varying maximum voiced frequency determines the limit between the two bands. In the lower band, the signal is represented solely by harmonically related sinewaves with slowly varying amplitudes, and frequencies:

$$h(t) = \sum_{k=1}^{K(t)} A_k(t) \cos(k\theta(t) + \phi_k(t)) \quad (1)$$

with  $\theta(t) = \int_{-\infty}^t \omega_0(l) dl$ .  $A_k(t)$  and  $\phi_k(t)$  are the amplitude and phase at time  $t$  of the  $k$ -th harmonic,  $\omega_0(t)$  is the fundamental frequency and  $K(t)$  is the time-varying number of harmonics included in the harmonic part.

The upper band, which contains the noise part, is modeled by an AR model and is modulated by a time-domain amplitude envelope. The noise part,  $n(t)$ , is therefore supposed to have been obtained by filtering a white Gaussian noise  $b(t)$  by a time-varying, normalized all-pole filter  $h(\tau, t)$  and multiplying the result by an energy envelope function  $w(t)$ :

$$n(t) = w(t) [h(\tau, t) \star b(t)] \quad (2)$$

The estimation of HNM parameters is an off-line process where a segmented speech database is analyzed and the HNM parameters (the harmonic amplitudes, the harmonic phases, and the parameters of the AR model) are estimated and saved into an inventory file [7].

At synthesis time, HNM parameters are concatenated and the prosody of some units may be altered in order to match the desired prosody. In case of pitch modification, amplitudes and phases are estimated at the new harmonics [8]. Next, HNM parameters have to be smoothed around concatenation points (this mainly means linear interpolation of harmonic amplitudes [7]). The last step is the generation of the synthetic signal using the stream of (potentially) modified HNM parameters. Synthesis is performed in

a pitch-synchronous way (without any use of glottal closure instants) using an overlap and add (OLA) process. The harmonic part is synthesized directly in the time-domain as a sum of harmonics (Eq.1). The noise part is obtained by filtering a unit-variance white Gaussian noise through a normalized all-pole filter. If the frame is voiced, the noise part is filtered by a high-pass filter with cutoff frequency equal to the maximum voiced frequency. Then, it is modulated by a time-domain envelope synchronized with the pitch period. This modulation of the noise part was shown [11] to be necessary in order to preserve the naturalness of some speech sounds, such as voiced fricatives.

The use of high pass filters for the generation of the noise part increases the complexity of the HNM module. Therefore, we have decided to simplify the synthesis structure by generating the noise part as a sum of harmonics with random phases [10][12]. For unvoiced frames the fundamental frequency has been set to  $100Hz$ , while for voiced frames we have used the estimated fundamental frequency for both bands; for the lower band (periodic part), and for the upper band (non-periodic part). This way, Eq.1 describes the entire spectrum for both unvoiced and voiced frames and for periodic and non-periodic parts. The number of harmonics,  $K(t)$ , is then given by:

$$K(t) = \pi/\omega(t) \quad (3)$$

For convenience of notation we will use  $K$  instead of  $K(t)$  for the rest of the paper.

## 3. DIFFERENT WAYS TO ADD K HARMONICS

In this section we review four different techniques for the generation of the harmonic signal with HNM. This is important for reducing the complexity of HNM because more than 80% of the execution time of the HNM synthesis module is spent on generating the synthetic signal (Eq.1).

### 3.1. Straight-forward synthesis, SF

The first attempt is to directly generate the synthetic signal by applying Eq.1. We will refer to this method as SF. The main problem with this method is the generation of the cosine functions. Although modern machines may have very fast trigonometric functions, this slice is very expensive.

### 3.2. Inverse Fast Fourier Transform, IFFT

The first thought to speed up the generation of the synthetic signal is the use of the Inverse FFT. FFTs may be used when the number of frequency bins (size of the FFT) is a number of a power of two. Because the number of harmonics may not be such a number, an assignment of the known frequency information (harmonics) to the closest frequency bins is necessary. This introduces, however, an error in the synthetic signal. The bigger the size of the FFT, the smaller the error (or, otherwise, the higher the SNR). However, the bigger the size of FFT, the slower the generation of the signal (higher complexity). McAulay and Quatieri, found that for  $4kHz$  bandwidth speech no loss of quality was detected provided the FFT length was at least 512 points [10]. Although the bandwidth we tested the FFT method for is  $8kHz$ , we have found that this length is not enough. Therefore, we have done tests with larger FFT sizes (e.g., 1024, 4096, 8192). While the error in the synthesized signal is reduced by increasing the size of the FFT, the generation of the

harmonic signal is slowing down considerably, as we will show later.

### 3.3. Recurrence Relations for Cosine functions, RR

Trigonometric functions whose arguments form a linear sequence  $\theta = \theta_0 + n\delta$  with  $n = 0, 1, 2, \dots$ , are efficiently calculated by the following recurrence [13]:

$$\cos(\theta + \delta) = \cos \theta - [\alpha \cos \theta + \beta \sin \theta] \quad (4)$$

$$\sin(\theta + \delta) = \sin \theta - [\alpha \sin \theta - \beta \cos \theta] \quad (5)$$

where  $\alpha$  and  $\beta$  are the precomputed coefficients

$$\alpha = 2 \sin^2\left(\frac{\delta}{2}\right) \quad (6)$$

$$\beta = \sin \delta \quad (7)$$

When the increment  $\delta$  is small, then the recurrence relations do not lose significance. For each harmonic,  $k$ , we have to compute the coefficients  $\alpha_k$  and  $\beta_k$  (Eqs.6 and 7) where  $\delta_k = k\omega_0$ .

### 3.4. Delayed Multi-Resampled Cosine functions, DMRC

In this method the phase information is first transformed into phase delays. The phase delay,  $t_k$ , of the  $k$ th harmonic is defined as:

$$t_k = -\phi(k\omega_0)/k\omega_0 \quad (8)$$

where  $\phi(k\omega_0)$  is the measured phase at  $k\omega_0$  frequency. Phase delays are expressed in samples and therefore are less sensitive to quantization errors. Transforming phase spectrum into phase delays allows us to write Eq.1 as following:

$$h(t) = \sum_{k=1}^K A_k(t)X([tk - t_k] \bmod T) \quad (9)$$

where  $\bmod$  stands for modulo,  $T$  is the integer pitch period in samples, and  $X$  denotes the cosine function:

$$X(t) = \cos(t\omega_0), \quad t = 0, \dots, T-1 \quad (10)$$

Eq.9 shows that  $h(t)$  may be generated in a simple way. First, we compute the signal  $X(t)$  (actually,  $X(t)$  is precomputed as there is a limited possible number of integer pitch periods and it is just loaded from the disk during the generation of the harmonic signal), and then for every  $k$  harmonic,  $X(t)$  is delayed by  $t_k$ , and downsampled by a factor  $k$ .

## 4. RESULTS AND DISCUSSION

In this section we compare the four previously presented methods based on their speed to generate a harmonic signal of  $K$  harmonics, and based on the signal to noise ratio (SNR) defined as:

$$SNR = 10 \log_{10} \frac{\sigma_{s(t)}^2}{\sigma_{h(t)-s(t)}^2} \quad (11)$$

where  $\sigma_{h(t)-s(t)}^2$  denotes the variance of the modeling error and  $\sigma_{s(t)}^2$  denotes the variance of the original speech signal  $s(t)$ . We have collected 500 voiced frames (250 of a female voice and 250 of a male voice, having a distribution of fundamental frequency

from 75 Hz up to 300 Hz) and each frame was synthesized 10,000 times. The whole experiment was repeated five times. All the methods were implemented in C, and compiled with optimization. All the experiments were conducted on the same SGI machine with an Irix 6.5 operating system. Table. 1 shows the median SNR for each of these methods and the relative median times allocated by each of the four methods to synthesize a voiced frame for 10,000 times (in order to measure computing time accurately). The relative values are computed based on the median time for the SF method (this is why the relative median time for SF in Table. 1 is one). Also, at the same table we show the results with five different lengths of IFFTs. In the computed times, we neither include

Method	Median SNR (dB)	Relative Median Time
SF	31.21	1
IFFT (512)	5.04	0.206
IFFT (1024)	10.66	0.238
IFFT (2048)	16.65	0.444
IFFT (4096)	21.28	0.984
IFFT (8192)	27.52	2.507
RR	29.89	0.158
DMRC	30.62	0.047

Table 1: Median SNR and relative median times for the four candidate methods from the whole experiment.

the assignment of the frequency information (harmonic amplitudes and phases) to the frequency bins for the IFFT method, nor do we include the computation of the phase delays for the DMRC method<sup>1</sup>. The results presented in Table. 1 are depicted graphically in Fig. 1 where the absolute median time (in seconds) for each of these methods is reported. It is worthwhile to note that the median time for the DMRC method was 3 seconds while for the SF method was 63 seconds. This means a reduction in the complexity of approximately 95%. At the same time the SNR using DMRC is comparable to the SNR of SF. It is worth also noting that a large size of FFT is necessary in order to achieve high SNR. This, however, slows down significantly the generation of the harmonic signal. The second fastest method is the RR method. All the presented methods make use of the phase spectrum with the exception of the DMRC method. Phase, however, is very difficult to code. Previous attempts in sinusoidal coders to cope with the coding of phase include minimum phase approaches [10] and the use of all-pass filters [14]. However, these techniques cannot be used for high-quality speech synthesis because the synthetic speech using minimum phase (and all-pass filters) is perceived, unfortunately, as buzzy. As speech databases for speech synthesis are getting bigger (e.g., for TTS systems based on unit selection [15] [16]) the compression of these databases is becoming an important issue. A compressed database does not only occupy less disk space. The time of accessing a compressed database is also smaller than the time of accessing the same database in its uncompressed form. Preliminary results showed that phase delays were less sensitive to quantization errors. For instance, representing phase delays as a percentage of the integer pitch period, allowed us to use 7 bits for the representation of a phase delay. Using this rather simplistic scheme of ‘‘coding’’ a compression of the speech database by two

<sup>1</sup>Phase delays are actually computed off-line without any need to compute them during synthesis. For the FFT method, however, the operation of the assignment of the available frequency information to the frequency bins should be done during synthesis.

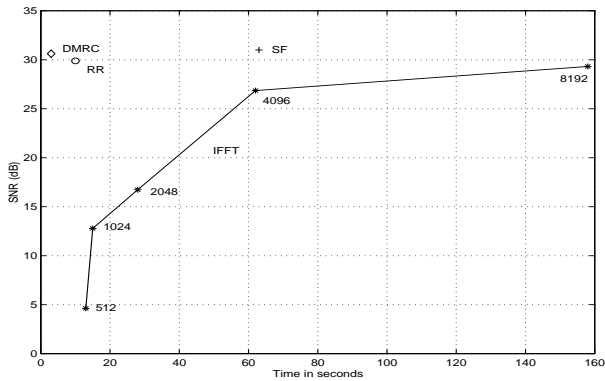


Figure 1: Comparing the four methods for the generation of a harmonic signal. SF (+), FFT (\*), RR(o), DMRC(∞).

was achieved. In the future, a more sophisticated coding scheme will allow a much higher compression ratio.

Motivated by these nice properties of phase delays (fast generation of the speech signal, possible compression of the speech database) we decided to update HNM with the DMRC method and compare it with the current version of HNM (which is using the SF method) in the task of synthesis of continuous speech. Informal listening tests with 8 listeners were conducted. Results showed that speech synthesized by the updated HNM system was superior to the one synthesized by the current HNM. As it was expected, the updated HNM system was about 12 times faster than the current HNM synthesis module. Finally, as mentioned above, with a rather simplistic coding scheme of phase information (using phase delays) the speech database was compressed by a factor of two.

## 5. CONCLUSION

In this paper, four different techniques were tested for fast generation of a signal represented as the sum of  $K$  harmonics with the condition of high SNR. We compared the straightforward synthesis, SF, synthesis based on the inverse Fast Fourier Transform, IFFT, synthesis based on recurrence relations for trigonometric functions, RR, and finally, synthesis based on Delayed Multi-Rsampled Cosine functions, DMRC. DMRC was found to be the fastest of all of the other techniques allowing a reduction of the complexity of the current HNM by 95%. When this new way to synthesize harmonic signals was included into the HNM synthesis module, HNM was found to run 12 times faster than the duration of the original speech signal. Moreover, informal listening tests showed that the synthetic signal obtained using the DMRC method was superior in quality to the one obtained with the version of HNM used so far (SF method plus modulated noise). Finally, in the paper we propose the use of phase delays instead of phase for a less sensitive to quantization error way to represent phase. In the future, we plan to use phase delays for an effective coding of the phase information in order to compress large speech databases for speech synthesis. Phase delays may also be used for the re-estimation of phase information in case of pitch modifications as well as for smoothing the phase information around concatenation points in concatenative speech synthesis.

## 6. REFERENCES

- [1] R. Sproat and J. Olive, "An Approach to Text-To-Speech Synthesis," in *Speech Coding and Synthesis*, pp. 611–633, Elsevier, 1995.
- [2] E. Moulines and F. Charpentier, "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones," *Speech Communication*, vol. 9, pp. 453–467, Dec 1990.
- [3] T. Dutoit and H. Leich, "Text-To-Speech synthesis based on a MBE re-synthesis of the segments database," *Speech Communication*, vol. 13, pp. 435–440, 1993.
- [4] D. Griffin and J. Lim, "Multiband-excitation vocoder," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-36, pp. 236–243, Feb 1988.
- [5] M. W. Macon, *Speech Synthesis Based on Sinusoidal Modeling*. PhD thesis, Georgia Institute of Technology, Oct 1996.
- [6] M. Crespo, P. Velasco, L. Serrano, and J. Sardina, "On the Use of a Sinusoidal Model for Speech Synthesis in Text-to-Speech," in *Progress in Speech Synthesis* (J. V. Santen, R. Sproat, J. Olive, and J. Hirschberg, eds.), pp. 57–70, Springer, 1996.
- [7] Y. Stylianou, "Concatenative speech synthesis using a harmonic plus noise model," *Third ESCA Speech Synthesis Workshop*, pp. 261–266, Nov. 1998.
- [8] Y. Stylianou, J. Laroche, and E. Moulines, "High-Quality Speech Modification based on a Harmonic + Noise Model.," *Proc. EUROSPEECH*, pp. 451–454, 1995.
- [9] A. Syrdal, Y. Stylianou, L. Garisson, A. Conkie, and J. Schroeter, "TD-PSOLA versus Harmonic plus Noise Model in diphone based speech synthesis," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 273–276, 1998.
- [10] R. J. McAulay and T. F. Quatieri, "Low-rate speech coding based on the sinusoidal model," in *Advances in Speech Signal Processing* (S. Furui and M. Sondhi, eds.), ch. 6, pp. 165–208, Marcel Dekker, 1991.
- [11] J. Laroche, Y. Stylianou, and E. Moulines, "HNS: Speech modification based on a harmonic + noise model.," *Proc. IEEE ICASSP-93, Minneapolis*, pp. 550–553, Apr 1993.
- [12] Y. Stylianou, *Harmonic plus Noise Models for Speech, combined with Statistical Methods, for Speech and Speaker Modification*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, Jan 1996.
- [13] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C, Second Edition*. Cambridge University Press, 1994.
- [14] S. Ahmadi and A. S. Spanias, "A new phase model for sinusoidal transform coding of speech," *IEEE Trans. Speech and Audio Processing*, vol. 6(5), pp. 495–501, Sept. 1998.
- [15] W. N. Campbell, "CHATR: A High-Definition Speech Re-Sequencing System," in *Proc. 3rd ASA/ASJ Joint Meeting, (Hawaii)*, pp. 1223–1228, 1996.
- [16] M. Beutnagel, A. Conkie, J. Schroeter, Y. Stylianou, and A. Syrdal, "The AT&T Next-Gen TTS System.," *137th meeting of the Acoustical Society of America*, 1999. <http://www.research.att.com/projects/tts>.