VOCAINE THE VOCODER AND APPLICATIONS IN SPEECH SYNTHESIS

Yannis Agiomyrgiannakis

Google

agios@google.com

ABSTRACT

Vocoders received renewed attention recently as basic components in speech synthesis applications such as voice transformation, voice conversion and statistical parametric speech synthesis. This paper presents a new vocoder synthesizer, referred to as Vocaine, that features a novel Amplitude Modulated-Frequency Modulated (AM-FM) speech model, a new way to synthesize non-stationary sinusoids using quadratic phase splines and a super fast cosine generator. Extensive evaluations are made against several state-of-the-art methods in Copy-Synthesis and Text-To-Speech synthesis experiments. Vocaine matches or outperforms STRAIGHT in Copy-Synthesis experiments and outperforms our baseline real-time optimized Mixed-Excitation vocoder with the same computational cost. We report that Vocaine considerably improves our statistical TTS synthesizers and that our new statistical parametric synthesizer [1] matched the quality of our mature production Unit-Selection system with uncompressed waveforms.

Index Terms— vocoders, statistical parametric speech synthesis, text-to-speech, non-stationary, AM-FM, fast cosine generators, phase models, overlap-add, sinusoidal speech models

1. INTRODUCTION

A vocoder is a key component of modern speech synthesis applications because it provides a parameterization of the speech waveform that is amenable to quantization, modification and statistical modelling. The commercial interest for vocoders started with speech coding, e.g. the Sinusoidal Transform Codec (STC) [2, 3] and Advanced MultiBand Excitation (AMBE) [4] but was later reduced for about a decade following the dominance of CELP codecs. During that period, many speech waveform models were proposed for use in voice transformation and voice conversion, e.g. the Harmonic-plus-Noise Model (HNM) [5], [6]. Similar notable work was made by the audio processing community with the Sinusoids-plus-Noise model [7] and IRCAM's Super Vocoder Phase [8]. The quest to realistically modify voice quality characteristics led to cumbersome speech models that incorporate characteristics of the glottal source [9] and its variants [10].

A renewed interest for vocoders came with the application of Statistical Parametric Speech Synthesis (SPSS) [11]. Statistical synthesizers require a parametric representation of speech that is amenable to statistical modeling [12, 13]. The two most notable contributions here are STRAIGHT [14, 15] and AhoCoder [16] which is an STC/HNM variant. Both vocoders are reported to be of similar quality [16] but STRAIGHT is more widely used by the community as a baseline. A perceptual evaluation between these as well as other vocoders can be found at [17]; care however must be taken when sinusoidal vocoders with measured phase are compared against vocoders with artificial phase because the former correspond to waveform approximating models that currently are not amenable to statistical modelling. Statistical modelling of phases is possible [18] and it is exciting to see whether it can further improve naturalness. We will henceforth focus on those related to contemporary speech synthesis.

STRAIGHT synthesis is too slow to be used in practice because it relies on high-order FFT for high-resolution spectral synthesis. Attempts to make it faster may replace FFTs with an log-spectrum filter [19] or a variable lattice filter [20] and mixed-excitation [21]. Mixed-excitation is a pulse plus colored noise that sounds buzzy sometimes. STRAIGHT reduces buzziness using an all-pass filter to fuzzify the pulse in higher frequencies, an approach that works well in many cases but alters voicing quality, reduces *brightness* and does not produce realistic voiced fricatives; factors that influence naturalness when Mean Opinion Score (MOS) is above 4.0.

Sinusoidal vocoders like STC, HNM and AhoCoder on the other hand do not have an implicit way to deal with intra-harmonic noise and forcefully split the speech spectrum in two parts: deterministic and noise. HNM and AhoCoder have explicit noise models that use modulation to incorporate noise into the signal, a trick that has also being used successfully in [22], [9]. Band-splitting has the disadvantage that it is biased towards synthesizing noisy higher formants that penalize brightness and that it is sensitive to voicing decisions, frequently suffering from artifacts in unvoiced-voiced transients and onsets.

Vocaine was designed to overcome shortcomings of STRAIGHT and HNM with low computational complexity. It is a *universal vocoder synthesizer*, a speech waveform renderer, that can be used to synthesize speech from diverse parameterizations originating from different analysis methods. Section 2 presents a novel speech model that describes the speech signal in a single equation as a sum of non-stationary sinusoids and incorporates a coherent noise-modulation model for frication and breathiness. Section 3 presents a phase-locked pitch-synchronous synthesis mechanism that allows explicit phases to be set at glottal closure instants and successfully deals with transients. Section 4 presents a novel phase interpolation technique that is referred to as *quadratic phase splines* that is very fast and automatically introduces dithering noise at non-stationary parts of speech. Section 5 presents a novel super-fast cosine generation procedure that significantly lowers computational complexity. Section 6 presents objective and subjective results in Copy-Synthesis and SPSS.

2. VOCAINE SPEECH MODEL

Vocaine describes the speech signal $s(n)$ as a sum of non-stationary modulated sinusoids:

$$s(n) = A_1(n)\cos(\phi_1(n)) + \sum_{k=2}^{K} A_k(n)\left[\gamma_0 - \gamma_1\alpha_k(n)\cos(\phi_1(n))\right]\cos(\phi_k(n)), \quad (1)$$

where K is the number of sinusoids, $n = 1, 2, ..., T_s$ is the time index in samples, T_s is the synthesis period, $A_k(n)$, $\phi_k(n)$ and $\alpha_k(n) \in [0, 1]$ are the *instantaneous amplitude*, the *instantaneous phase* and the *instantaneous aperiodicity* of k-th sinusoid, respectively. γ_0 is the *modulation bias* and γ_1 is the *modulation factor*. The term aperiodicity is borrowed from STRAIGHT and in this paper it is used equivalently to the term *non-deterministic*. Let $\omega_k(n) = \frac{\partial \phi_k(n)}{\partial n}$ be the *instantaneous frequency* of the k-th sinusoid. The sinusoids are by construction harmonically related at the end-points of the synthesis period: $\omega_k(0) = k\omega_1(0)$, $\omega_k(T_s) = k\omega_1(T_s)$.

Each sinusoid except the first one is modulated by the *Coherent Noise-Modulation (CNM)* signal:

$$g_k(n) = \gamma_0 + \gamma_1 \alpha_k(n) \cos(\phi_1(n)). \quad (2)$$

Typical values used for γ_0 and γ_1 are 1.0 and 0.5, respectively. The CNM signal modulates the k-th sinusoid according to instantaneous aperiodicity $\alpha_k(n)$ so that the modulated sinusoid exhibits a time-domain structure that concentrates the energy around the maxima of the first sinusoid $\cos(\phi_1(n))$. When the sinusoid is deterministic ($\alpha_k(n) = 0.0$) there is no modulation. When the sinusoid is purely non-deterministic ($\alpha_k(n) = 1.0$) the modulation shapes the time-domain envelope of the sinusoid according to aperiodicity. Thus, the more aperiodic the sinusoid the stronger the time-shaping. In the frequency domain, the modulation introduces images of the sinusoid at $\omega_k(n) \pm \omega_1(n)$ where $\omega_i(n) = \frac{\partial \phi_i(n)}{\partial n}$ is the instantaneous frequency of the i-th sinusoid that is by construction linked to the fundamental frequency.

Justification for the modulation signal arises from many directions. From a signal perspective, speech can be regarded as an AM-FM (Amplitude Modulation / Frequency Modulation) process [23]. High-band speech exhibits a pitch-synchronous time envelope [24]. From a production perspective, the cyclic behaviour can be attributed to the fact that the power of the vocal source is minimized during the closed phase of the glottal cycle. Coupling between the glottal source and the laryngeal cavity generates a laryngeal formant between 3 kHz and 7 kHz that also exhibits a cyclic behaviour [25]. From an auditory perspective, noise bursts are masked when synchronized with pulses [26]. Finally, many papers report better synthesis quality when they used modulated noise [6], [9], [22], [24], [27], [28].

In Vocaine, the modulation signal improves significantly the quality of voiced speech and, voiced fricatives in particular, resulting to much higher MOS values for languages rich in voiced fricatives like French. Vocaine uses the same signal model for voiced and unvoiced speech; what differs is the randomicity of the endpoint phases at $n = 0$ and $n = T_s$. Unvoiced speech is synthesized with sinusoids with fundamental endpoint frequencies of 100 Hz and uniformly random endpoint phases. The endpoint phases ϕ_k of voiced speech are randomized according to aperiodicity as follows:

$$\hat{\phi}_k = \psi_k + U\left(-h(\alpha_k)\frac{\pi}{4}, +h(\alpha_k)\frac{\pi}{4}\right), \quad (3)$$

where ψ_k are the *deterministic endpoint phases*, α_k is the *endpoint aperiodicity* and $U(a, b) \in [a, b]$ is the *dispersion phase* which is a uniformly distributed random variable. $h(\cdot) : [0, 1] \mapsto [0, 1]$ is the *aperiodicity conversion function* that is used to compensate an analysis bias that is usually found in aperiodicity estimations and a synthesis bias that is related to the specifics of the wave generation. A sigmoid function seems to work pretty well for STRAIGHT-based aperiodicities. For the results presented in this paper we used $\psi_k = \pi/2$.

3. PHASE-LOCKED PITCH-SYNCHRONOUS SYNTHESIS

Synthesis in Vocaine is made in chunks of audio that correspond to a pitch-period in voiced speech and to 5 ms in unvoiced speech. Every chunk of audio is contained between so-called *Reference Synthesis Instants* (RSI) that define its endpoints. An RSI corresponds to a glottal closure instant in voiced speech and to a regularly sampled time-instant in unvoiced speech. The spectral parameters of the Vocaine signal model are sampled at every RSI from the vocoder parameters. Vocaine operates in a streamed manner, receiving packets of vocoder parameters at a fixed rate, i.e. every 5 ms, storing them in a circular buffer and using as many as needed to synthesize one chunk of audio at a request. Only the packages that contain the endpoint RSIs of the audio chunk are used; the rest are discarded.

The RSIs are computed according to the fundamental pitch period in voiced speech and to a fixed 10 ms period in unvoiced speech. It is easier to demonstrate the process in the example of Figure 1: The first RSI is taken at the beginning of the first packet. The second RSI is taken $T1$ samples after the first one, where $T1$ is the pitch period of the first packet. The second RSI is contained at the 4-th packet which has a pitch period of $T2$ samples, and so on. When Vocaine has synthesized until the third RSI, it has to wait for 4 more packets to synthesize until the forth RSI.

Every audio chunk is synthesized in a way that ensures that the phases at RSIs are exactly those described by equation (3). If the fundamental period is not an integer then phase residuals must be propagated during synthesis. This complicates the implementation but is easily avoided by quantizing the fundamental frequency to correspond to an integer fundamental period.

The pitch-synchronous synthesis is referred to as *phase-locked* and Vocaine as *phase-aware* because the user can set desirable phases at the RSIs. However, the synthesized signal cannot approximate the original because RSIs are set according to the fundamental period. Desirable phases can originate from elaborated phase models or even ones measured from the waveform.

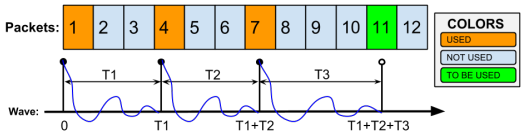

Fig. 1. Synchronous to asynchronous operation.

Voicing transients, in particular unvoiced-to-voiced transients and voiced onsets, are notoriously hard to synthesize without artifacts. Sinusoidal vocoders are also prone to introduce artifacts when pitch is irregular because it is difficult to match sinusoids in that case. Quatieri et al. resorts to a heuristic to match sinusoids [29]. Vocaine uses a hybrid strategy that switches between PSOLA (Pitch Synchronous OverLap Add) [30] synthesis and sinusoidal synthesis. When the period-to-period variation of pitch is relatively small (below 25%) then the sinusoids at the two endpoints are matched and sinusoidal synthesis is used for them. The rest of the sinusoids are synthesized using PSOLA. When pitch variation is above 25% then PSOLA is used for all sinusoids. PSOLA is also used for audio chunks where one of the two endpoints is voiced and the other is unvoiced. This seems to work quite well in all transients.

4. QUADRATIC PHASE SPLINES

This section describes how to synthesize a sinusoid between the two endpoints of an audio chunk. Instantaneous amplitudes and phases are linearly interpolated but phase interpolation is more complicated

due to the circular behaviour of phases. Quatieri et al. suggested a cubic phase model [3] that corresponds to parabolic instantaneous frequency curve. A parabolic instantaneous frequency curve is computationally expensive and occasionally introduces glitches at non-stationary speech where the endpoint frequencies and phases do not correspond to a smooth trajectory. Vocaine uses a novel quadratic phase spline model, where the synthesis period $[0, T]$ is split into two parts, $[0, n_c]$ and $[n_c, T_c]$ and each part is synthesized using a quadratic phase curve. The instantaneous frequency $\omega_{k,c}$ at the break-point $n_c = \lfloor T/2 \rfloor$ is allowed to vary in order to obtain a maximally smooth trajectory. Thus, the quadratic phase spline model corresponds to a piecewise-linear instantaneous frequency model with the intermediate frequency being a degree of freedom that is used to makes it maximally smooth.

Let $\phi_{k,s}(n) = \theta_{k,s} + \omega_{k,s} n + \gamma_{k,s} n^2$, $n \in [0, n_c]$ be the *start-spline* and $\phi_{k,e}(n) = \alpha_{k,s} + \beta_{k,e}(n - n_c) + \gamma_{k,e}(n - n_c)^2 + 2\pi M$, $n \in [n_c, T]$ be the *end-spline*. Phase is known up to an unknown multiple of 2π due to its circular nature. The parameters of the splines are computed by solving the equations using the following *continuity constraints* for the splines:

$$\begin{aligned}
\hat{\phi}_{k,s} &= \phi_{k,s}(0), \text{(start phase)} \\
\hat{\omega}_{k,s} &= \frac{\partial \phi_{k,s}(n)}{\partial n}|_{n=0}, \text{(start frequency)} \\
\hat{\phi}_{k,e} &= \phi_{k,e}(T), \text{(end phase)} \\
\hat{\omega}_{k,e} &= \frac{\partial \phi_{k,e}(n)}{\partial n}|_{n=T}, \text{(end frequency)} \\
\phi_{k,s}(n_c) &= \phi_{k,e}(n_c), \text{(break-point phase continuity)} \\
\frac{\partial \phi_{k,s}(n)}{\partial n} &= \frac{\partial \phi_{k,e}(n)}{\partial n}|_{n=n_c}, \text{(break-point frequency continuity)}
\end{aligned}$$

M is a *phase-unwrapping integer* that is chosen to produce a maximally smooth trajectory in the second derivative sense:

$$\hat{M} = \left\lfloor \arg\min_M \left\{ \int_0^{n_c} \left| \frac{\partial^2 \phi_{k,s}(t)}{\partial n^2} \right|^2 dt + \int_{n_c}^{T} \left| \frac{\partial^2 \phi_{k,e}(t)}{\partial n^2} \right|^2 dt \right\} \right\rfloor$$

Remember that endpoint phases are randomized according to aperiodicity in equation (3). The more aperiodic a sinusoid is, the more random the endpoint phases are and the more noisy a sinusoid should be. An easy trick to make a sinusoid track noisy is to add a frequency dispersion term so that power spreads around the vicinity of the sinusoid frequency. For example, an unvoiced sinusoid track that starts from 100 Hz and ends at 100 Hz with random phases at the endpoints would sound less periodic if the instantaneous frequency at the middle of the track is, say, 140 Hz. In Vocaine, *adding dispersion phase breaks the periodicity of the signal* providing a richer unvoiced sound. This is made because the breakpoint frequency depends on the stationarity of the sinusoid. When the sinusoid is fully stationary, the breakpoint frequency is exactly the middle of the endpoint frequencies. When the sinusoid is non-stationary, the breakpoint frequency deviates from the middle of the endpoint frequencies. The details of the derivation of the spline parameters are omitted due to space restrictions but are easy to derive.

5. UNSAFE SUPER-FAST COSINE GENERATION

Cosine generation constitutes a significant percentage of the computational cost of a sinusoidal speech model like Vocaine and many attempts have been made to reduce the increased complexity of cosine generators [31], [32], [33]. Synthesis of stationary sinusoids can benefit from recursive formulas [31], [34] like the ones used in Goertzel transform [35]. However, for Vocaine we are interested in random access cosine generators where there is no exploitable structure in the series of cosine computations. In [31], a set of codebooks, one for each fundamental period were pre-computed and the codebook with the fundamental period that is closest to the fundamental period of the synthesized sinusoid was obtained using modulo addressing. This method uses an excessive number of codebooks while modulo addressing requires a modulo-division, which is a computationally expensive operation even for modern processors. The large memory requirements are likely to cause expensive cache misses in mobile devices.

The obvious alternative is to use a simple codebook approximation which can be described by equation:

$$\cos(\omega) \approx C_N \left[\lfloor \langle \omega \rangle_{2\pi} \frac{N}{2\pi} \rfloor \right], \tag{4}$$

where $C_N[n] = \cos(2\pi \frac{n}{N})$ is a codebook with N entries, $\langle \cdot \rangle_{2\pi}$ is a modulo-2π operator, and $\lfloor \cdot \rfloor$ is the floor operator. The modulo-2π operator can be implemented using the so-called *additive range reduction* formula:

$$\langle \omega \rangle_{2\pi} = \omega - \lfloor \frac{\omega}{2\pi} \rfloor 2\pi, \tag{5}$$

which requires 2 multiplications, 2 conversions or 1 floor instruction and 1 subtraction: a multiplication by $\frac{1}{2\pi}$, a conversion from float to integer and back for the flooring operation, a multiplication by 2π and a subtraction. The overall computation requires 3 multiplications, 3 conversions, 1 addition and 1 memory read. For the purpose of waveform synthesis, this method can achieve a sufficiently high SNR (Signal-to-Noise ratios) of $\approx$ 37 dB for codebook size $N = 256$, but its accuracy might not be sufficient for other applications. Improved accuracy can be achieved at the cost of an extra addition if we add 0.5 prior the floor operation.

This work proposes to further speed-up the codebook method by forcing $N = 2^B$ (a B-bit codebook) and by *sacrificing some of the available input range*. Instead of allowing ω to take any possible value, we restrict it to be within an operational range so that:

$$\cos(\omega) \approx C_N \left[\left\langle \lfloor \omega \frac{N}{2\pi} \rfloor \right\rangle_N \right]. \tag{6}$$

Assuming 32-bit floating point arithmetic, the equation is equivalent to the C++ instructions:

```
float cos(float omega) {
  static float alpha = N / (2 * PI);
  static int mask_N = (N - 1);
  float omega_Nf = omega * alpha;
  int omega_Ni = (int)omega_Nf;
  return codebook[omega_Ni & mask_N];
}.
```

The multiplication converts the function from a modulo-2π circular function to a modulo-N circular function. The conversion to a 32-bit integer locates the principal value of the circular function to the least significant B bits, while the most significant bits hold the multiple of the period. For example, for $\omega = k2\pi + \phi$, the B least significant bits hold ϕ, while the rest (most-significant) bits hold k. The generator is called *unsafe* because this separation holds only if we have no overflow during the conversion to integer. In the case of overflow, the B least significant bits will contain information for ϕ and k as well. Thus, for an 8-bit codebook, we can afford to have

only 32-8=24 bits for k. In two's complement integer arithmetic - the standard nowadays - the modulo-N operation can be implemented using a bitwise-AND instruction. Thus, the proposed *unsafe codebook-based generator* requires only 1 multiplication, 1 conversion, 1 bitwise-AND and 1 memory read. In many modern DSP processors the last two instructions can be implemented with a single circular addressing memory read. In this work we used a 512-size codebook.

The complexity is further reduced by recasting phase maths in a N-circular space and accordingly modifying all signal processing of Vocaine. This saves the multiplication, so we end-up with a cosine generator that requires 2-3 instructions, less than half of the 7 instructions needed for the safe implementation. Furthermore, since we operate in a modulo-N space, it is possible to use fixed-point arithmetic throughout.

6. RESULTS

Vocaine is almost as fast as our current optimized embedded solution, considering only synthesis speed. A comparison was made against a MATLAB implementation of STRAIGHT and optimized embedded production mixed-excitation implementations with LSP (Line Spectrum Pairs) and MCEP (Mel-Cepstrum) parameterizations [21]. The computational cost of our floating point C++ Vocaine implementation for 22 kHz signals is only 8% higher than the mixed-excitation LSP vocoder, 2% higher than the mixed-excitation MCEP vocoder and remarkably faster than STRAIGHT (no fair measurement can be made with a matlab implementation, though). Please note the production mixed-excitation implementations are faster than real-time in a diverse set of devices in the Android ecosystem. Further optimization of our embedded statistical synthesizers reduced the overall computational cost significantly below our previous baseline.

We evaluated the quality of synthesized speech in Copy-Synthesis experiments and in SPSS experiments. Copy-Synthesis quality is an upper-bound of quality of SPSS. It is misleading to evaluate vocoders solely on the output of a parametric synthesizer because the statistical mapping can alleviate issues of the vocoder. Further, the vocoder can be used to synthesize recorded segments together with synthetic ones.

All subjective evaluations were made for naturalness Mean Opinion Score (MOS) using a supervised high-quality crowd-sourced system. Rating scale from 5 to 1 was the standard: "excellent","good","fair","poor","bad". 100 utterances were presented to hundreds of listeners. Every utterance was independently and randomly assigned for rating to 8 different listeners. Table 1 presents Copy-Synthesis results for two languages, English and French. Only high-quality professional recordings related to Google Text-To-Speech (TTS) products were used: 4 females and 1 male for English and 1 female only for French. We can observe that Vocaine with Mixed-Excitation analysis is equivalent to STRAIGHT and significantly better than the Mixed-Excitation vocoders with no computational overhead. Vocaine with parameters from STRAIGHT analysis (*Vocaine+STRAIGHT*) outperforms STRAIGHT but statistical significance is being reached only for the French case. Thus, STRAIGHT analysis is better than our Mixed-Excitation analysis. Further, Vocaine+STRAIGHT MOS values are remarkably high, 4.114 for English and 4.265 for French, and gets pretty close to recorded speech. Vocaine works particularly well for French because French is rich in voiced fricatives and our speaker has a breathy voice character which is well represented by Vocaine's signal model.

Another listening test was made for naturalness MOS using several TTS synthesizers. The experimental setting was the same with the previous experiment, and we also included the same Copy-Synthesis stimuli for reference and MOS scale calibration. For TTS we used a female voice with a 30K corpus and we evaluated our current production Unit-Selection synthesizer, an Mixed-Excitation HMM-based synthesizer and two Vocaine-based synthesizers, one that uses HMM and another that uses LSTM (Long-Short-Term-Memory recurrent neural network) and is also presented in this conference [1]. The results are shown in Table 2. The Vocaine-based synthesizer outperforms the current HMM-based synthesizer with statistical significance in a fair comparison. The comparison between LSTM- , HMM- and unit-selection-based synthesizers synthesizers is not fair because the LSTM one uses more text features. However, the Vocaine+LSTM synthesizer matched the quality of our production Unit-Selection TTS, which is a remarkable result because it is the first time that we observe a statistical TTS competing with a mature unit-selection system with uncompressed speech. We conducted an AB listening test for further investigation and the result was that the two systems were almost equivalent. However, another AB test revealed that the Vocaine+LSTM synthesizer is not as good as our best Unit-Selection synthesizer. Both AB results are not presented here due to space limitations.

Table 1. Copy-Synthesis Results: MOS + Confidence Interval

Stimuli	MOS (US-EN)	MOS (FR-FR)
Recorded wav	4.493 ± 0.101	4.568 ± 0.058
Vocaine+STRAIGHT analysis	4.144 ± 0.132	4.265 ± 0.073
Vocaine+MixedExc analysis	4.079 ± 0.116	4.031 ± 0.076
STRAIGHT anal./synth.	4.074 ± 0.126	4.016 ± 0.080
MixedExc - MCEP	3.877 ± 0.110	3.544 ± 0.092
MixedExc - LSP	3.699 ± 0.140	3.307 ± 0.106

Table 2. Text-To-Speech Results: MOS + Confidence Interval

Stimuli	MOS (US-EN)
Recorded wav	4.529 ± 0.086
Vocaine+STRAIGHT analysis	4.337 ± 0.094
Vocaine+MixedExc analysis	4.176 ± 0.114
STRAIGHT analysis/synthesis	4.090 ± 0.111
Production Unit-Selection TTS	3.773 ± 0.128
Vocaine+LSTM TTS	3.738 ± 0.095
Vocaine+HMM TTS	3.472 ± 0.103
MixedExc+HMM TTS (MCEP)	3.314 ± 0.120

7. CONCLUSION

SPSS is particularly suitable for embedded speech synthesis and is heavily based on vocoding which has to be fast enough for embedded devices. Vocaine improves the quality of our embedded TTS without a computational penalty and matches or outperforms the state-of-the-art STRAIGHT that requires considerable computational resources. Vocaine's naturalness MOS in Copy-Synthesis experiments ranges between 4.144 and 4.337 in various experiments. Besides significant quality improvements over the baselines we report that our Vocaine+LSTM statistical synthesizer has reached the quality of our production unit-selection speech synthesis system with uncompressed waveforms.

8. REFERENCES

[1] H. Zen and H. Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis," in *Acoustics, Speech and Signal Processing, 2015. ICASSP 2015 Proceedings. 2015 IEEE International Conference on*. IEEE, 2015.

[2] T. J. McAulay and T. F. Quatieri, "Sinusoidal Transform Coding," in *Mobile Satellite Conference*, 1988, vol. 1, pp. 503–508.

[3] T. F. Quatieri, *Discrete-time speech signal processing: principles and practice*, Pearson education, 2008.

[4] D. W. Griffin and J. S. Lim, "Multiband Excitation Vocoder," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 36, no. 8, pp. 1223–1235, 1988.

[5] Y. Stylianou, "Applying the Harmonic-plus-Noise Model in concatenative speech synthesis," *Speech and Audio Processing, IEEE Transactions on*, vol. 9, no. 1, pp. 21–29, 2001.

[6] Y. Stylianou, *Harmonic plus noise models for speech, combined with statistical methods, for speech and speaker modification*, Ph.D. thesis, Ecole Nationale Supérieure des Télécommunications, 1996.

[7] X. Serra and J. Smith, "Spectral modeling synthesis: a sound analysis/synthesis system based on a deterministic plus stochastic decomposition," *Computer Music Journal*, pp. 12–24, 1990.

[8] A. Röbel, "A new approach to transient processing in the phase vocoder," in *Proc. of the 6th Int. Conf. on Digital Audio Effects (DAFx03)*, 2003, pp. 344–349.

[9] Y. Agiomyrgiannakis and O. Rosec, "Towards flexible speech coding for speech synthesis: an LF+ modulated noise vocoder.," in *INTERSPEECH*, 2008, pp. 1849–1852.

[10] Y. Agiomyrgiannakis and O. Rosec, "ARX-LF-based source-filter methods for voice modification and transformation," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 3589–3592.

[11] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[12] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. of ICASSP*. IEEE, 2000, vol. 3, pp. 1315–1318.

[13] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. of ICASSP, Vancouver, Canada, May*, 2013, pp. 7962–7966.

[14] H. Kawahara, "STRAIGHT, exploitation of the other aspect of VOCODER: Perceptually isomorphic decomposition of speech sounds," *Acoustical science and technology*, vol. 27, no. 6, pp. 349–353, 2006.

[15] H. Kawahara, M. Morise, T. Takahashi, R. Nisimura, T. Irino, and H. Banno, "Tandem-straight: A temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, f0, and aperiodicity estimation," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, March 2008, pp. 3933–3936.

[16] D. Erro, I. Sainz, E. Navas, and I. Hernaez, "Harmonics-plus-Noise Model based vocoder for Statistical Parametric Speech Synthesis," 2014.

[17] Q. Hu, K. Richmond, J. Yamagishi, and J. Latorre, "An experimental comparison of multiple vocoder types," in *8th ISCA Workshop on Speech Synthesis, Barcelona, Spain*, 2013, pp. 135–140.

[18] Y. Agiomyrgiannakis and Y. Stylianou, "Wrapped Gaussian Mixture Models for modeling and high-rate quantization of phase data of speech," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 4, pp. 775–786, 2009.

[19] S. Imai, K. Sumita, and C. Furuichi, "Mel log-spectrum approximation (MLSA) filter for speech synthesis," *Electronics and Communications in Japan (Part I: Communications)*, vol. 66, no. 2, pp. 10–18, 1983.

[20] A. V. Oppenheim, R. W. Schafer, J. R. Buck, et al., *Discrete-time signal processing*, vol. 2, Prentice-hall Englewood Cliffs, 1989.

[21] A. V. McCree and T. P. Barnwell III, "A mixed excitation LPC vocoder model for low bit rate speech coding," *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 4, pp. 242–250, 1995.

[22] Y. Agiomyrgiannakis and Y. Stylianou, "Combined estimation/-coding of highband spectral envelopes for speech spectrum expansion," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*. IEEE, 2004, vol. 1, pp. I–469.

[23] A. Potamianos and P. Maragos, "Speech analysis and synthesis using an AM-FM modulation model," *Speech Communication*, vol. 28, no. 3, pp. 195–209, 1999.

[24] Y. Pantazis and Y. Stylianou, "Improving the modeling of the noise part in the Harmonic-plus-Noise model of speech," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 4609–4612.

[25] T. Kitamura, H. Takemoto, S. Adachi, P. Mokhtari, and K. Honda, "Cyclicity of laryngeal cavity resonance due to vocal fold vibration," *The Journal of the Acoustical Society of America*, vol. 120, no. 4, pp. 2239–2249, 2006.

[26] J. Skoglund and W. B. Kleijn, "On time-frequency masking in voiced speech," *Speech and Audio Processing, IEEE Transactions on*, vol. 8, no. 4, pp. 361–369, 2000.

[27] A. McCree, "A 14 kb/s wideband speech coder with a parametric highband model," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*. IEEE, 2000, vol. 2, pp. II1153–II1156.

[28] D. Mehta and T. F Quatieri, "Synthesis, analysis, and pitch modification of the . vowel," in *Applications of Signal Processing to Audio and Acoustics, 2005. IEEE Workshop on*. IEEE, 2005, pp. 199–202.

[29] R. J. McAulay and T. F. Quatiery, "Audio, analysis/synthesis based on a sinusoidal representation," 1986.

[30] H. Valbret, E. Moulines, and J. P. Tubach, "Voice transformation using PSOLA technique," *Speech Communication*, vol. 11, no. 2, pp. 175–187, 1992.

[31] Y. Stylianou, "A simple and fast way of generating a harmonic signal," *Signal Processing Letters, IEEE*, vol. 7, no. 5, pp. 111–113, 2000.

[32] R. J. McAulay and T. F. Quatieri, "Computationally efficient sine-wave synthesis and its application to sinusoidal transform coding," in *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*. IEEE, 1988, pp. 370–373.

[33] R. J. McAulay and T. F. Quatieri, "Low-rate speech coding based on the sinusoidal model," *Advances in speech signal processing*, vol. 6, pp. 165–208, 1992.

[34] M. Vasilakis, Y. Agiomyrgiannakis, and Y. Stylianou, "Fast analysis/synthesis of harmonic signals," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*. IEEE, 2006, vol. 3, pp. III–III.

[35] G. Goertzel, "An algorithm for the evaluation of finite trigonometric series," *American mathematical monthly*, pp. 34–35, 1958.