HIGH QUALITY VOICE TRANSFORMATIONS BASED ON MODELING RADIATED VOICE PULSES IN FREQUENCY DOMAIN

Jordi Bonada

MTG, Institut Universitari de l'Audiovisual
Universitat Pompeu Fabra, Barcelona, Spain
jordi.bonada@iua.upf.es

ABSTRACT

This paper introduces a method to transform voice based on modeling the radiated voice pulses in frequency domain. This approach tries to combine the strengths of classical time and frequency domain techniques into a single framework, providing both an independent control of each voice pulse and flexible timbre and phase modification capabilities.

1. INTRODUCTION

Our goal is to achieve high quality natural transformations of singing and speech voiced utterances. The type of targeted transformations is quite wide and covers both high (pitch-shifting, time-scaling, timbre modification) and low level ones (vocal disorders).

During voiced utterances the airflow going trough the vocal folds is chopped into a set of glottal pulses which are then filtered by the vocal tract, tongue and lips. The resulting waveform can be understood as the result of overlapping all the filtered glottal pulses, therefore showing an impulsiveness feature. In terms of frequency domain representation, this impulsiveness comes out from the phase synchronization between harmonics existing at the pulse onsets.

Classical time domain techniques [1][2] can transform the voice and preserve this phase alignment with a relatively low computational cost. The basic procedure consist on cutting short fragments from the waveform (typically one period) and overlap and add them at synthesis. However most difficulties appear when we want to achieve complex timbre modifications. Maybe the main drawback of these techniques is that they can't isolate one single impulse response but they cut the result of several overlapped ones, thus degrading the quality.

FOF (Formant Wave Function Synthesis, [3]) is another time-domain technique which features a flexible and fast additive synthesis method. It models the human voice as the sound from an impulse generator, equivalent to the vocal chords, passing through a set of band-pass filters which represent the characteristics of the vocal tract (each filter corresponds to a vocal formant). This technique can successfully control independently each pulse. However, it roughly represents the analysis amplitude spectrum and doesn't reproduce the voice phase alignment at glottal pulses.

On the other hand, classical frequency domain techniques [4][5][6] can accomplish a wider range of transformations with a high computational cost, including complex timbre modifications and exotic effects (inharmonizer, morphing). However, they require continuing the phase of the harmonics and preserving their phase relation at pulse onsets (which is rather tricky). Besides, if the signal is modeled with pure sinusoids the transformations sound often artificial and synthetic, even adding some residual. Advanced phase vocoder techniques [7] have shown encouraging results preserving the local spectrum (amplitude and phase) behavior around the harmonics. This local spectrum describes somehow the context of the partial (amplitude, frequency evolution), but it is not yet clear how to properly modify this context (for example adding a vibrato, so changing the frequency evolution). In addition complex peak picking strategies must be considered to avoid problems with noisy or masked peaks.

In this paper we present an algorithm which combines the waveform preservation ability of time-domain techniques with the flexible and wide transformations of frequency-domain techniques, while at the same time avoiding the complexity and the contextual problems of them. This algorithm consists on modeling the radiated voice pulses in frequency domain, and is able to transform independently each one of them.

2. MODELING RADIATED VOICE PULSES IN FREQUENCY DOMAIN

The STFT of a windowed frame $x(n)$ can be expressed as

$$x(n)=s(n)\cdot w(n)$$
$$X(e^{j\Omega})=\sum_{n=0}^{N-1}x(n)e^{-j\Omega n} \tag{1}$$

where $s(n)$ is the input signal and $w(n)$ is the window function.

Let's assume a rectangular window of N samples and a finite input signal shorter than N and covered by the window. If the input signal is delayed by Δn samples, then its STFT will be

$$\begin{aligned} S_{delayed\,\Delta n}\left(e^{j\Omega}\right) &= \sum_{n=0}^{N-1}s(n-\Delta n)e^{-j\Omega n} = \\ &= \sum_{n=0}^{N-1}x(n-\Delta n)e^{-j\Omega n} = \\ &\overset{m=n-\Delta n}{=} \sum_{m=-\Delta n}^{N-1-\Delta n}x(m)e^{-j\Omega(m+\Delta n)} \approx X(e^{j\Omega})e^{-j\Omega\Delta n} \end{aligned} \tag{2}$$

where the last approximation would become an identity if the delayed signal was also fully covered by the window.

Let's now consider $y(n)$ as the sum of R identical signals $s(n)$ delayed by Δn samples, where some overlap is possible. We can calculate its STFT as follows:

$$y(n)=s(n)+s(n-\Delta n)+s(n-2\Delta n)+...+s(n-(R-1)\Delta n) \tag{3}$$

therefore

$$\begin{aligned} Y(e^{j\Omega}) &= \sum_{n=0}^{N-1} y(n) e^{-j\Omega n} = \\ &\approx X(e^{j\Omega})\left[1+e^{-j\Omega\Delta n}+e^{-2j\Omega\Delta n}+\ldots+e^{-(R-1)j\Omega\Delta n}\right]= \\ &= X(e^{j\Omega})\sum_{r=0}^{R-1} e^{-j\Omega\Delta n r} = X(e^{j\Omega})\frac{1-e^{-j\Omega\Delta n R}}{1-e^{-j\Omega\Delta n}} = \\ &= X(e^{j\Omega}) e^{-j\Omega\Delta n\frac{R-1}{2}} \frac{\sin(0.5\Omega\Delta n R)}{\sin(0.5\Omega\Delta n)} \triangleq \\ &\triangleq X(e^{j\Omega})\operatorname{sinc}_R(\Omega\Delta n) \end{aligned} \tag{4}$$

The effect of the term $\operatorname{sinc}_R(\Omega\Delta n)$ is somehow sampling the spectrum of $X(e^{j\Omega})$, since this term can be seen as a train of equidistant pulses located each $2\pi/\Delta n$ radians (see Figure 1). All the pulses have a constant value of *R*, as can be seen below:

$$\begin{aligned} &\operatorname{sinc}_R(\Omega\Delta n)\Big|_{\Omega_r = r\frac{2\pi}{\Delta n}} = e^{-j\pi r(R-1)}\frac{\sin(r\pi R)}{\sin(r\pi)} \\ &\lim_{\Omega_r \to r\frac{2\pi}{\Delta n}} \left|\operatorname{sinc}_R(\Omega\Delta n)\right| = R \\ &\lim_{\Omega_r \to r\frac{2\pi}{\Delta n}} \measuredangle \operatorname{sinc}_R(\Omega\Delta n) = 0 \end{aligned} \tag{5}$$

Consequently, if we assume $\left|X(e^{j\Omega})\right|$ and $\measuredangle X(e^{j\Omega})$ vary slowly along frequency, we can estimate $X(e^{j\Omega})$ from $Y(e^{j\Omega})$ by interpolating the values at frequencies Ω_r. The interpolation algorithm could be just a linear interpolation but a spline method is preferred. The resolution depends on Δn where a bigger value means better resolution. The reconstructed signal $x'(n)$ can be computed by means of the inverse STFT of the estimated $X'(e^{j\Omega})$

$$x'(n) = \frac{1}{N}\sum_{n=0}^{N-1} X'(e^{j\Omega}) e^{j\Omega n} \tag{6}$$

and finally we could approximate the input signal as

$$s'(n) = x'(n).$$

In the case of voiced utterances in a speech or singing voice recording, $s(n)$ corresponds to a radiated glottal pulse filtered by the vocal tract. However, what we usually deal with is the result of several consecutive voice pulses, as we see in Figure 2, which are overlapped along time and not fully covered by the window. In this case Δn is straightforwardly related to the voice pitch by $\Delta n = f_s / pitch$, where f_s is the sampling rate. Besides, the radiated voice pulses are not identical because the voice production system is changing its characteristics continuously along time. However, if the window is short enough so that the signal is quasi-stationary, then the above method would allow estimating a single radiated voice pulse.

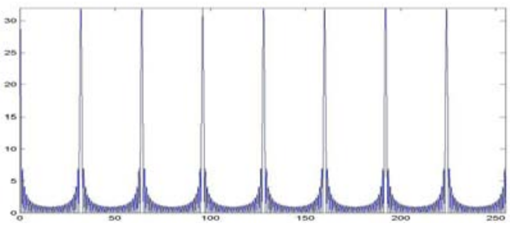

Figure 1: $\left|\operatorname{sinc}_R(\Omega\Delta n)\right|$ *, R=32, Δ=8, N=256.*

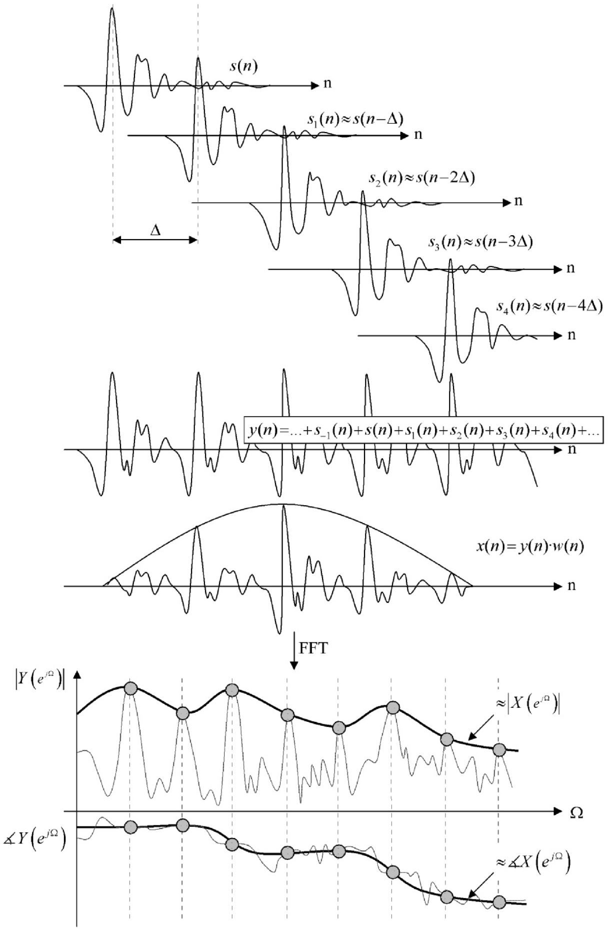
$s(n)$
n
$s_1(n)\approx s(n-\Delta)$
n
$s_2(n)\approx s(n-2\Delta)$
n
$s_3(n)\approx s(n-3\Delta)$
n
$s_4(n)\approx s(n-4\Delta)$
n
Δ
$y(n)=\ldots+s_{-1}(n)+s(n)+s_1(n)+s_2(n)+s_3(n)+s_4(n)+\ldots$
n
$x(n)=y(n)\cdot w(n)$
n
FFT
$\left|Y(e^{j\Omega})\right|$
$\approx\left|X(e^{j\Omega})\right|$
Ω
$\measuredangle Y(e^{j\Omega})$
$\approx\measuredangle X(e^{j\Omega})$

Figure 2: *Spectrum estimation of a single radiated voice pulse*

3. VOICE ANALYSIS

The analysis is done as a constant frame-rate process where we get an estimation of the STFT of a centered radiated pulse for each step (see Figure 3). In our experiments we have used about 172 frames per second. The analysis starts windowing the input voice signal and computing its STFT. Then the spectral peaks are detected out of the amplitude spectrum using a parabolic interpolation and inputted into the pitch detection algorithm, actually an extension of the TWM algorithm [8]. The pitch is then used by the peak selection module to choose the harmonic peaks considering as a guide both the perfect harmonic distribution and the amplitude and frequency of spectral peaks. Next, the phase of the harmonic peaks is modified by the *maximally flat phase alignment* (MFPA, see Section 3.1) algorithm which centers the voice pulse in the window. The final step is to interpolate the harmonic peaks and estimate the spectrum of the radiated voice pulse $X'(e^{j\Omega})$. The amplitude spectrum is interpolated using a 3rd order spline method and afterwards scaled by 1/*R*, where *R* is the number of pulses contained in the analysis window. *R* can be computed from the pitch by

$$R = \frac{N \cdot pitch}{f_s} \tag{7}$$

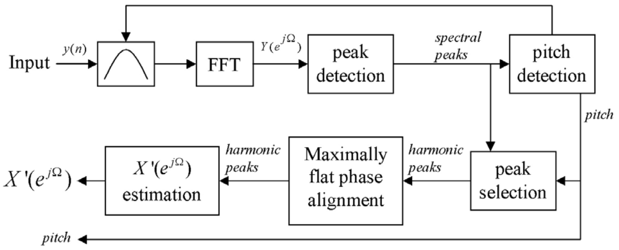

Figure 3: *Block diagram of the analysis process.*

Related to the window size *N*, we need a good frequency resolution in order to precisely detect the spectral peaks, therefore a long window. However, in such case, the transitions, attacks and releases would become considerably smoothed by the windowing since radiated pulses with different characteristics (energy, timbre) would be processed together and pitch would not be approximately constant along the window. The compromise adopted has been to adapt the size of the window to cover around three periods, which assures enough resolution to discern the peaks while at the same time increases the temporal resolution and improves the handling of non stationary parts. Besides, zero padding is applied in order to improve the spectrum resolution.

3.1. Maximally flat phase alignment (MFPA)

As will be seen later in the synthesis section, there is a need to control the position in time of each voice pulse. Therefore we need a way to estimate the position of each pulse or at least of one of them. One way to deal with this issue is to shift in time the input signal so that one of the voice pulses becomes centered in the window.

When the voice pulse is almost centered it happens that harmonics are usually synchronized in a way that the phase spectrum is nearly flat with phase shifts under each resonance (i.e. formant) area. This can be seen in Figure 2. Whenever we move the voice pulse, the corresponding time shift adds a phase shift which varies linearly along frequency ($e^{j\Omega\Delta t}$). Thus, one way to estimate the pulse location is to estimate the slope of this linear phase shift. However, phase wrapping complicates the problem because all phase values are contained in the range $]-\pi, \pi]$.

We have come up with an easier method to approximate the center location. With this procedure we pretend to find the time-shift Δt which minimizes the phase differences between harmonics, therefore obtaining a maximally flat phase alignment (MFPA), as explained in the following:

a) Define several fundamental phase candidates φ'_{c0} in the interval $[-\pi, \pi)$

b) For each candidate, apply the corresponding time shift Δt_c to each harmonic peak. The phase of each harmonic φ_i will be rotated by $\Delta\varphi_i = 2\pi f_i \; \Delta t_c$, where f_i is the frequency of the harmonic.

c) Compute the sum of rotated phase differences as $\varphi_{diff} = \sum_i |\,\mathrm{princarg}(\varphi'_{i+1} - \varphi'_i)\,|$, where *princarg* is a function which puts an arbitrary radian phase value into the interval $]-\pi,\pi]$ by adding an integer number of periods (2π).

After several candidates are estimated we obtain an error function which is similar to a sinusoid and whose minimum sets the desired fundamental phase $\varphi_{\min}$ which approximately centers the voice pulse (see Figure 4).

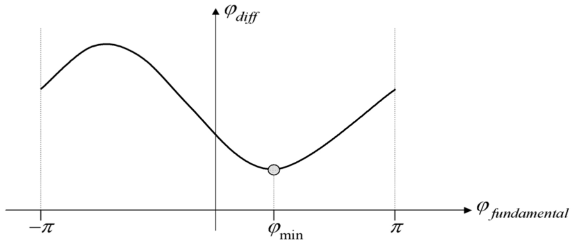

Figure 4: *Maximally flat phase alignment*

Actually only low frequency harmonics (up to around 3 Khz) are considered since usually most energy is located at low frequencies and also because higher frequency harmonics are often unstable or noisy. Besides a continuation algorithm for $\varphi_{\min}$ is used for better handling noisy parts.

3.2. Phase Unwrapping and interpolation

Whenever we want to interpolate the peak's phases we have to consider the wrapping issue and choose the shortest way between two consecutives phases. The unwrapping procedure sets the initial phase to be $\phi_0 = \varphi_0$. The next unwrapped phases ϕ_i can be obtained iteratively by

$$\phi_i = \phi_{i-1} + \mathrm{princarg}(\varphi_i - \varphi_{i-1}) \qquad (8)$$

The estimated phase spectrum $\measuredangle X(e^{j\Omega})$ is then computed interpolating the unwrapped phases ϕ_i . It may happen for successive harmonics that during phase unwrapping different number of periods are added for consecutive frames. In such case we will get discontinuities between the harmonics' frequencies after interpolation (see Figure 5).

In order to avoid this artifact, for each harmonic a phase correction ($\Delta\phi^i_{correc}$) is added which compensates the period differences and slowly decays to zero, getting 0.1 radians closer each frame. This way, if in frame *n*-1 two consecutive harmonics (*i*–1 and *i*) have a phase difference of -0.9π and in the next frame *n* the difference become −1.1π, the phase correction would be $\Delta\phi^i_{correc} = -2\pi$. In the next frame *n*+1, the phase correction would decay a little, thus $\Delta\phi^i_{correc} = -2\pi + 0.1$, and if the phase difference becomes again −0.9π then 2π would be added to the phase correction, so it would be $\Delta\phi^i_{correc} = 0.1$. The final sequence

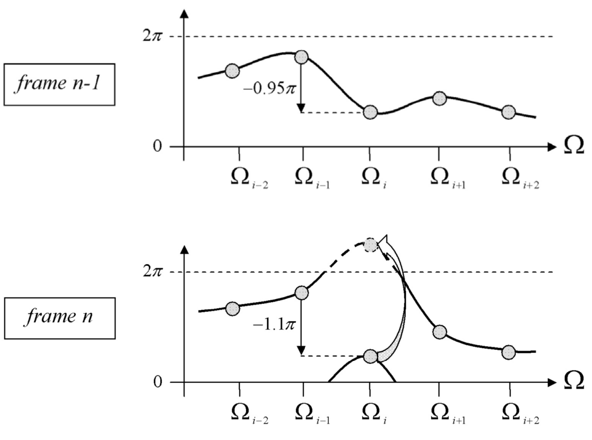

Figure 5: *Phase unwrapping problem: different number of 2π periods are added for harmonic i*th.

of phases for the harmonic *i-th* would be $\phi_{i-1} - 0.9\pi$, $\phi_{i-1} - 1.1\pi$, $\phi_{i-1} - 0.8\pi$, with no unwrapping artifact.

4. SYNTHESIS

The analysis outputs the estimated pitch and spectrum of radiated voice pulses centered in the window. In order to synthesize a voiced utterance, the first thing we have to do is to generate a sequence of pulse locations. Such sequence can be derived from the pitch envelope by separating consecutive pulses by one period time $T = 1/pitch$. If we want to better match the input signal pulse sequence we could also use the unwrapped fundamental phase envelope and the outputs of the MFPA to determine the position of each pulse in the original sequence. We don't need to care about each pulse's amplitude since the original amplitude is preserved trough the estimated spectral amplitude itself.

Once we have filled the sequence we have to map each pulse to one analysis frame and get the radiated voice pulse spectrum. We can choose the nearest frame or interpolate the spectrum of the two closest frames to get smoother timbre transitions, which can be useful specially when applying a time-stretch transformation. At this point we can synthesize each pulse independently by means of an IFFT and then position it in the desired location. This would require an interpolation of the samples in time domain because pulse time onsets are not quantized to the sampling rate. However the positioning can also be done directly in frequency domain by adding a linear phase slope to the phase spectrum proportional to the required time-shift, which can written as

$$\Delta\phi_r(\Omega) = (t_r - t_{frame})\Omega \tag{9}$$

where $\Delta\phi_r(\Omega)$ is the phase increment for the r^{th} pulse and t_{frame} is the center time of the current frame. Here we have to be careful about the hop and window size values because there is a limit of how much the signal can be shifted by this method since the window is finite and short. If the window size is *N* we can not time-shift the pulse more than *N*/2 samples, otherwise the pulse will appear in the opposite part of the window as an aliasing effect. This time-shift operation can be computed efficiently on a complex spectrum since the phase shift increases by a constant amount for consecutive bins, so we can compute only two cosinus for the whole spectrum and then two complex multiplications per bin.

Depending on the synthesis configuration it can be good to combine into a single IFFT several pulses so to speed up the process. This way we would generate a complex spectrum for each pulse and add it to the IFFT buffer. The computational cost for each synthesis frame would include a polar to complex conversion for each analysis frame used plus a time shift of the whole spectrum (see equation (9)) for each synthesized pulse. The computational cost would then be given by

$$C = n_A \cdot p2c + 2 \cdot n_S (cmul \cdot N + cos) + IFFT_N$$

$$\begin{aligned} p2c &= Polar\ to\ complex\ conversion \\ cmul &= complex\ multiplication \\ cos &= cosinus\ calculation \\ N &= synthesis\ window\ size \\ n_A &= analysis\ \ frames\ used \\ n_S &= synthesis\ \ frames\ used = \frac{pitch \cdot hopsize}{f_S} \end{aligned} \tag{10}$$

However, in certain contexts the polar to complex conversion can be done in the analysis stage, this way reducing significantly the synthesis computational cost. This would be the case, for example, of a sample based singing voice synthesizer which reads a preanalyzed database.

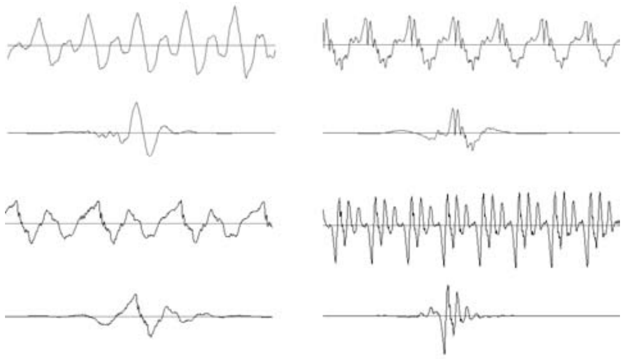

Figure 6: *Synthesis of a single pulse from the analysis of a recorded voiced utterance. On three of the examples the synthesized pulse location has been synchronized to the original one.*

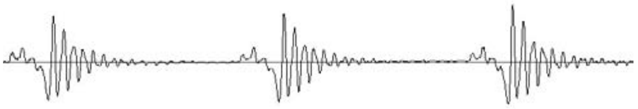

Figure 7: *Recording of a vocal fry phonation where the whole radiated glottal pulse is visible*

The resulting time domain signal after the IFFT must be multiplied by an overlapping window and added to the output sound buffer. A triangular window would be fine but it must have a size shorter then *N* to avoid artifacts due to time aliasing (the edges of the synthesized buffer will not be used). Here it is interesting to point out that the estimated spectrum of the radiated voice pulse $X'(e^{j\Omega})$ is not convolved anymore with the analysis window, thus the reconstructed signal doesn't need to be divided by it before overlapping.

In Figure 6 we can observe several examples where a single pulse has been synthesized out of the analysis of the voiced utterance. Besides, in Figure 7 we can see how it looks an isolated radiated voice pulse obtained from a vocal fry phonation with an extremely low pitch of about 30 Hz.

4.1. Voice residual

In the estimation of $X'(e^{j\Omega})$ we have obtained a spectrum from the interpolation of the harmonic peaks. It is clear that we have disregarded all the data contained in the spectrum bins between the harmonic frequencies. This data often explain irregularities of the pulse sequence (time and amplitude) plus noisy or breathy characteristics of the voice. These irregularities could be somehow reproduced by analyzing the unwrapped fundamental phase envelope and the outputs of the MFPA. On the other hand, the breathy part could be synthesized using some of the original spectrum data.

Initially we tried to use the residual obtained by subtracting in frequency domain perfect sinusoids with the amplitude, frequency and phase values of the detected harmonics to the original spectrum. However, sometimes we found that some harmonic information was kept, especially in transitions, producing artifacts. Better results can be obtained by applying comb-like filters to this residual which attenuate the frequency bands around the harmonic frequencies. Adding this residual to the synthesized signal adds a natural breathy characteristic very close to the original one.

4.2. Unvoiced segments

The proposed algorithm is thought to be used only in voiced sections, since it assumes a periodic signal as input. Actually, in our experiments we have combined it with a phase-vocoder based technique for unvoiced segments, in which a white noise source is filtered with the spectral amplitude of the input signal. This method allows applying transformations with high quality results. However, it fails to properly handle transients, so plosives consonants are smeared and intelligibility is degraded. In order to improve the results we have adapted a transient processing algorithm which is able to detect them and discriminate which spectral peaks contribute to them [9].

5. VOICE TRANSFORMATIONS

Several voice transformations can be achieved using this technique, from high level (pitch-shifting, time-scaling, timbre modifications) to low level ones (independent pulse control). In this Section we will briefly present how some of them can be applied.

Pitch and Time-Scaling

Pitch and time-scaling transformations can be implemented in a similar way to TD-PSOLA, controlling the speed at which the sequence of analysis frames is read and distance between pulses. The timbre will be preserved as long as the estimated amplitude spectrum is not modified.

Timbre Modification

Timbre can be modified by scaling, warping or equalizing the estimated spectral amplitude of each pulse. In the real case, if a formant is shifted in frequency the related phase shift (see 3.1) is shifted as well. This means it would be desirable to scale and warp the spectral phase envelope as well. This could be done by applying the same transformation to both amplitude and phase envelopes of the polar spectrum or just applying it once to the complex spectrum.

Voice Disorders

Several voice disorders, intentional or not, can be characterized by irregularities in the excitation glottal pulse sequence, both in time (jitter) and amplitude (shimmer). These irregularities can be described mostly by the appearance of subharmonics in the spectrum, which are hard to manipulate in frequency domain. However, in our case, we have an independent control of the location and amplitude of each pulse, thus easily different patterns of irregularities can be synthesized and even vary along time. We have observed that sometimes in a growl pattern (as the one shown in Figure 8) voice pulses have different timbres which differ by some equalization. In this example the two pulses with more amplitude experience a significant boost around 4 Khz. This behavior can also be reproduced with the presented technique by equalizing differently each pulse's amplitude spectrum.

6. RESULTS AND CONCLUSIONS

The presented method is able to isolate radiated voice pulses and model them in frequency domain. Several high and low level transformations can be applied to voiced utterances with high quality results. It provides an independent control of each synthesized pulse allowing to easily apply voice disorders patterns. The proposed algorithm is not suitable for unvoiced sections and therefore we have combined it with a phase-vocoder based technique to deal with them.

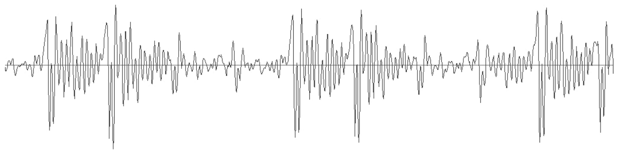

Figure 8: *Growl example.*

Compared to classical time-domain methods, the presented algorithm is able to obtain the whole voice pulse response and to apply complex timbre transformations. Compared to frequency-domain techniques it provides a way to control each pulse independently and minimizes artifacts due to noisy peaks and local spectrum contextualization.

This method has some difficulties to process voice utterances with vocal disorders. In such case, pulse onsets need to be detected, as well as some timbre differences at pulse level. More research should be done in this area.

We have tested the algorithm only with vocal sounds. It would be nice to test it with other instruments as well. This would lead us to think that maybe the presented algorithm is not so much related to voice modeling but to periodic signal modeling. However, it has been developed thinking on dealing with voice most relevant characteristics (periodicity, impulsive characterization as result of filtering a sequence of pulses, formants and their effect on the harmonic phase synchronization…) and also trying to achieve the most typical and interesting voice transformations.

7. REFERENCES

[1] E. Moulines, F. Charpentier, "Pitch synchronous waveform processing techniques for text to speech synthesis using diphones," *Speech Communication*, vol. 9, no. 5/6, pp. 453–467, 1990.

[2] W. Verhelst, M. Roelands, "An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech," *Proc. of the Int. Conf. on Acoustics, Speech, and Signal Processing*, Minneapolis, 1993.

[3] X. Rodet, Y. Potard, J. B. B. Barrière, "The CHANT Project: From the Synthesis of the Singing Voice to Synthesis in General," *Computer Music Journal,* vol. 8, no.3, pp.15–31, 1984.

[4] A. De Götzen, N. Bernadini, D. Arfib, "Traditional (?) implementations of a phase vocoder: The tricks of the trade," *Proc. of DAFx*, Verona, 2000, pp. 37–43.

[5] X. Amatriain, J. Bonada, A. Loscos, X. Serra, "Spectral Processing," in *DAFX: Digital Audio Effects,* Udo Zölzer Ed., Chapter 10, pp. 373–438. John Wiley & Sons, 2002.

[6] K. Fitz, *The reassigned bandwidth-enhanced method of additive synthesis*, Ph.D. Thesis, University of Illinois at Urbana-Champaign, 1999.

[7] J. Laroche, "Frequency-domain techniques for high-quality voice modification," *Proc. of DAFx*, London, 2003.

[8] P. Cano, "Fundamental Frequency Estimation in the SMS analysis," *Proc. of DAFx*, Barcelona, 1998.

[9] A. Röbel, "A new approach to transient processing in the phase vocoder," *Proc. of DAFx,* London, 2003.