Spectral Approach to the Modeling of the Singing Voice

Jordi Bonada, Alex Loscos, Pedro Cano, Xavier Serra
Audiovisual Institute, Pompeu Fabra University
Barcelona, Spain
{jordi.bonada, alex.loscos, pedro.cano, xavier.serra}@iua.upf.es
http://www.iua.upf.es/mtg

Hideki Kenmochi
Advanced System Development Center, YAMAHA Corporation
Hamamatsu, Japan
kenmochi@beat.yamaha.co.jp

[pre-print for 111th AES Convention 2001]

ABSTRACT

In this paper we present two different approaches to the modeling of the singing voice. Each of these approaches has been thought to fit in the specific requirements of two applications. These are an automatic voice impersonator for karaoke systems and a singing voice synthesizer.

1. INTRODUCTION

Singing voice synthesis has been an active research field for almost fifty years [Cook, 1996]. Most of the systems developed until now do not provide enough quality or do not meet the practical requirements to have found real-world commercial applications. Anyhow, it seems that one of the main issues behind singing voice synthesis is to offer not only quality but flexible and musically meaningful control over the vocal sound. In that sense, we may think of applications where impossible singing voices can be synthesized or where existing voices can be enhanced.

In a broad sense, and according to whether the focus is put on the system or its output, synthesis models used in singing voice synthesis can be classified into two main groups: spectral models and physical models. Spectral models are based on perceptual mechanisms of the listener while physical models focus on modeling the production mechanisms of the original system. Any of these two models might be regarded as suitable depending on the specific requirements of the application or may even be combined for taking advantages of both approaches.

The main benefit of using Physical models is that the parameters used in the model are closely related to the ones a singer uses to control his/her own vocal system. As such, some knowledge of the real-world mechanism can be brought on the design. The model itself can provide intuitive parameters if it is constructed so that it sufficiently matches the physical system. Conversely, such a system usually has a large number of parameters and the mapping of those quite intuitive controls of the production mechanism to the final output of the model, and so to the listener's perceived quality, is not a trivial task.

Alternatively, spectral models are closely related to some aspects of the human perceptual mechanism. Changes in the parameters of a spectral model can be more easily mapped to a change of sensation in the listener. Yet parameter spaces yielded by these systems are not necessarily the most natural ones for manipulation.

On this context, in this article we introduce two different applications related to singing voice synthesis. First, in section 2, we introduce the basis of the Spectral Modeling Synthesis technique, which has inspired many of the models characteristics. We then introduce a singing voice impersonator application that is

able to morph the user's voice with a professional singer's version in real-time. We finish by outlining, in section 4, the basic approach towards a singing voice synthesizer that is currently being developed by our team.

2. SPECTRAL MODELING SYNTHESIS

The Spectral Modeling Synthesis (SMS) is a synthesis by analysis technique based on modeling the sounds as stable sinusoids (partials) plus noise (residual component). This sinusoidal plus residual model can be seen as a generalization of the *STFT* and the S*inusoidal* representations as we can decide what part of the sound to model as sinusoidal and what part to leave as STFT. [Serra 1996; Serra and Smith 1990].

The input sound *s(t)* is modeled by,

$$s(t) = \sum_{r=1}^{R} A_r(t)\cos\left[\theta_r(t)\right] + e(t) \qquad (1)$$

where $A_r(t)$ and $\theta_r(t)$ are the instantaneous amplitude and phase of the r^{th} sinusoid, respectively, and *e(t)* is time-varying noise component.

This estimation of the sinusoidal component is generally done by first computing the STFT of the sound, then detecting the spectral peaks (and measuring the magnitude, frequency and phase of each one), and organizing them as time-varying sinusoidal tracks. By using the fundamental frequency information in the peak continuation algorithm, we can identify the harmonic partials.

The sinusoidal plus residual model assumes that the sinusoids are stable partials of the sound with a slowly changing amplitude and frequency. With this restriction, we are able to add major constraints to the detection of sinusoids in the spectrum and omit the detection of the phase of each peak. The instantaneous phase that appears in the equation is taken to be the integral of the instantaneous frequency $\omega_r(t)$, and therefore satisfies

$$\theta_r(t) = \int_o^t \omega_r(\tau)\, d\tau \qquad (2)$$

where $\omega(t)$ is the frequency in radians, and r is the sinusoid number. When the sinusoids are used to model only the stable partials of the sound, we refer to this part of the sound as the deterministic component.

The residual component is obtained by first generating the sinusoidal component with additive synthesis, and then subtracting it from the original waveform. This is possible because the instantaneous phases of the original sound are matched and therefore the shape of the time domain waveform preserved. A spectral analysis of this time domain residual is done by first windowing it, window which is independent of the one used to find sinusoids, and thus we are free to choose a different time-frequency compromise. Finally, the FFT is computed.

Within this model we can either leave the residual signal, *e(t)*, to be the difference between the original sound and the sinusoidal component, resulting into an identity system, or we can assume that *e(t)* is a stochastic signal. In this case, the residual can be described as filtered white noise,

$$e(t) = \int_0^t h(t,\tau)\, u(\tau)\, d\tau \qquad (3)$$

where *u(t)* is white noise and $h(t,\tau)$ is the response of a time varying filter to an impulse at time *t*. That is, the residual is modeled by the time-domain convolution of white noise with a time-varying frequency-shaping filter.

The calculation of the residual component can be optimized by subtracting the sinusoids directly in frequency domain (see figure 1). This can be done subtracting the spectrum resulting of the convolution of each sinusoid with the transform of the same window used in the residual spectral analysis.

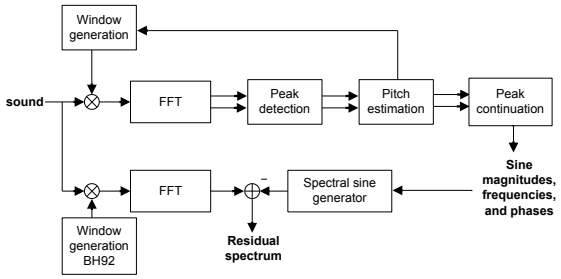

Fig 1 *Block diagram of the SMS analysis*

From the output of the analysis techniques presented we can obtain several features of the input sound and its frequency-domain representation. These features are then used in the transformation block in order to modify the characteristics of the sound in a meaningful way. Transformations as the ones used in the applications here presented (such as morphing, pitch shifting, spectral shape modification...) can be performed using this approach. All these transformations can be done in the frequency domain. Afterwards, the output sound can be synthesized.

The sinusoidal component is generated using some type of additive synthesis approach and the residual, if present, is synthesized using a subtractive synthesis approach using an IFFT approach, efficient implementations may be provided. Figure 2 shows a block diagram of the final part of the synthesis process.

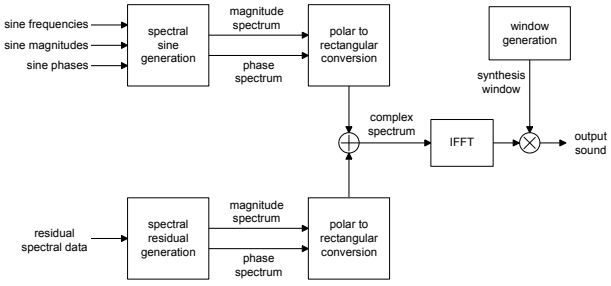

Fig 2 *Block diagram of the SMS synthesis*

Several modifications have been done to the basic SMS procedures to adapt them to the requirements of the applications outlined in this article. The major changes include the real-time implementation of the whole analysis/synthesis process with a processing latency of less than 30 milliseconds and the tuning of all parameters to the particular case of the singing voice. The later ones include the extraction of the most appropiate higher-level parameters for the case of the singing voice.

3 THE SINGING VOICE IMPERSONATOR

Morphing is a technique with which, out of two or more elements, we can generate new ones with hybrid properties. With different names, and using different signal processing techniques, the idea of audio morphing is well known in the Computer Music community [Serra, 1994; Tellman, Haken, Holloway, 1995; Osaka, 1995; Slaney, Covell, Lassiter. 1996; Settel, Lippe, 1996]. In most of these techniques, morph is based on the interpolation of sound parameterizations resulting from analysis/synthesis techniques, such as the Short-time Fourier Transform (STFT), Linear Predictive Coding (LPC) or Sinusoidal Models.

3.1 The application

The application we present here is a very particular case of audio morphing. What we want is to be able to morph, in real-time, two singing voice signals in order to control the resulting synthetic voice by mixing the characteristics of the two sources. In such a context, a karaoke-type application was developed in which the user can sing like his/her favorite singers [Cano, Loscos, Bonada, Boer, Serra, 2000; Boer, Bonada, Cano, Loscos, Serra, 2000]. The result is an automatic impersonating system that allows the user to morph his/her voice attributes (such as pitch, timbre, vibrato and articulations) with the ones from a prerecorded singer, which from now on we will refer to as *target*.

In this particular implementation, the target's performance of the complete song to be morphed is recorded and analyzed beforehand. In order to incorporate the corresponding characteristics of the target's voice to the user's voice, the system first recognizes what the user is singing (phonemes and notes), looks for the same sounds in the target performance (i.e. synchronizing the sounds), interpolates the selected voice attributes, and synthesizes the output morphed voice. All this is accomplished in real-time.

Figure 3 shows the general block diagram of the voice impersonator system. The system relies on two main techniques that define and constrict the architecture: the SMS framework and a Hidden Markov Model based Automatic Speech Recognizer (ASR). The SMS implementation is responsible of providing a suitable parameterization of the singing voice in order to perform the morph in a flexible and musically meaningful way. On the other hand, the ASR is responsible for aligning the singing voice of the user with that of the target.

Before we can morph a particular song, we have to supply information about the song to be morphed and the song recording itself (Target Information and Song Information). The system requires the phonetic transcription of the lyrics, the melody as MIDI data, and the actual recording to be used as the target audio data. Thus, a good impersonator of the singer that originally sang the song has to be recorded. This recording has to be analyzed with

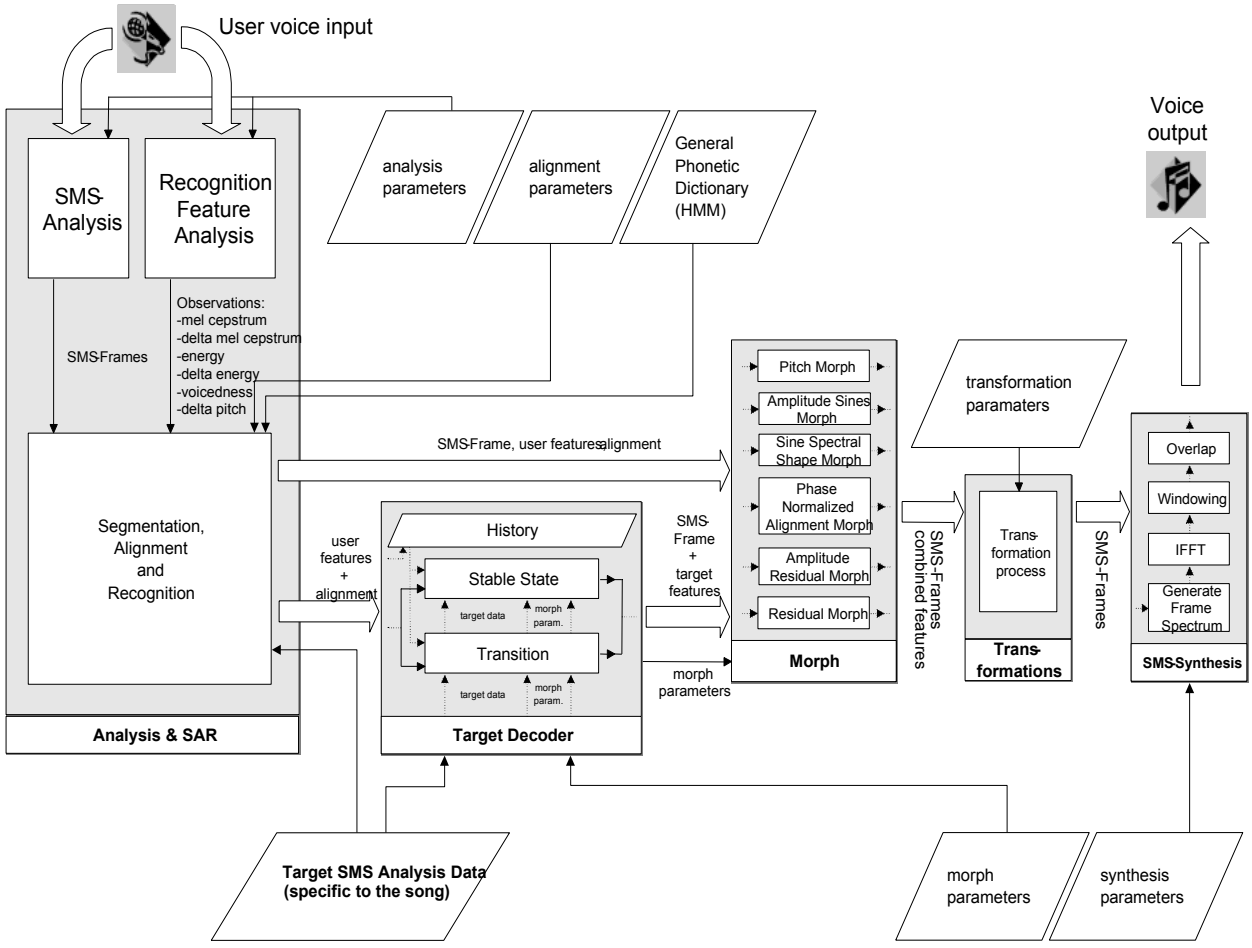

Fig 3 *Block diagram of the singing voice impersonator*

SMS, segmented into morphing units (phonemes), and each unit has to be labeled with the appropriate note and phonetic information of the song. This preparation stage is done semi-automatically, using a non-real time application developed for this purpose.

Once we have all the required inputs set we can start processing the user's voice. The first module of the running system includes the real-time analysis and the recognition/alignment steps. Each analysis frame, with the appropriate parameterization, is associated with the phoneme of a specific moment of the song and thus with a target frame. Once a user frame is matched to a target frame, we morph them interpolating data from both frames and we synthesize the output sound. Only voiced phonemes are morphed and the user has control over which and by how much each parameter is interpolated. The frames belonging to unvoiced phonemes are left untouched, thus always having the user's unvoiced consonants in the output.

Therefore, depending on the phoneme the user is singing, a unit from the target is selected and then each frame from the user is morphed with a different frame from the target, advancing sequentially in time as illustrated in figure 4. The user has the choice to interpolate the different parameters extracted at the analysis stage, such as amplitude, fundamental frequency, spectral shape, residual signal, etc. In general, the amplitude will not be interpolated, always using the amplitude from the user. The unvoiced phonemes will not be morphed either, so we will have always the user's. This will give the user the feeling of being in control of the synthesis.

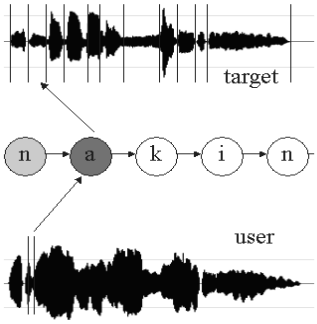

Fig 4 *Recognition and matching of morphable units*

In most cases, the durations of the user and target phonemes to be morphed will be different. If a given user's phoneme is shorter than the one from the target, the system will simply skip the remaining part of the target phoneme and go directly to the articulation portion. In the case when the user sings a longer phoneme than the one present in the target data, the system enters in the loop mode. Each voiced phoneme of the target has a loop point frame, marked in the preprocessing, non-real time stage. The system uses this frame to loop-synthesis in case the user sings beyond that point in the phoneme. Once we reach this frame in the target, the rest of the frames of the user will be interpolated with that same frame until the user ends the phoneme. In these cases, in order to avoid unnaturalness, we apply pitch templates obtained from longer utterances to the last frame of the target. This process is illustrated in figure 5.

Once all the chosen parameters have been interpolated for a given frame, they are added back to the basic SMS synthesis frame. Synthesis is done with the standard synthesis procedures of SMS.

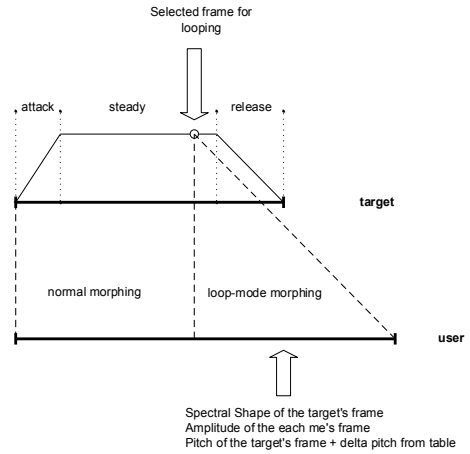

Fig 5 *Synthesis loop diagram*

3.2 The aligner

To solve the matching problem the system includes an ASR based on phoneme-base discrete HMM's. This ASR has been trained using a 22 minutes long Japanese singing database and the following front-end parameterization:

Mel Cepstrum	12 coefficients
Delta Mel Cepstrum	12 coefficients
Energy	1 coefficient
Delta Energy	1 coefficient
Voiceness	2 coefficients

The voiceness vector consists of a Pitch Error measure and the Zero Crossing rate. The Pitch Error component is a by-product from the fundamental frequency analysis, which is based on [Cano, 98]. The zero crossing rate is calculated by dividing the number of consecutive samples with different signs by the number of samples of the frame. The differentials (deltas) ponder up to two frames of the future and two frames of the past.

The alignment process starts with the generation of a phonetic transcription out of the lyrics text. This phonetic transcription is used to build the composite song Finite State Network (FSN) concatenating the models of the phonemes transcribed.

The phonetic transcription previous to the alignment process has to be flexible and general enough to account for all the possible realizations of the singer. It is very important to bear in mind the non-linguistic units silence and aspiration, as their appearance cannot be predicted. Different singers place silences and aspirations in different places. This is why, while building the FSN, between each pair of phoneme models, we insert both silence and aspiration models. In the transition probability matrix of the FSN, the jump probability a_{ij} from each speech phonetic unit to the next silence, aspiration or speech phonetic unit will be the same, as shown in figure 6.

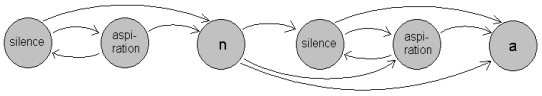

Fig 6 *Concatenation of silences and aspirations in the FSN*

The aspiration is problematic since in singing performance its dynamics are more significant. This causes the aspiration to be easily confused with a fricative. Moreover, different singers not only sing different but also, as in speech, pronounce different. To take into account these different pronunciations we modify the FSN to add parallel paths as shown in figure 7.

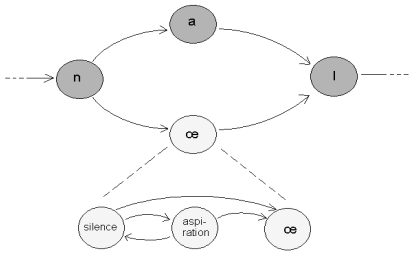

Fig 7 *Representation of a phonetic equivalence in the FSN*

The alignment resulting from the Viterbi decoding will follow the most likely path, so it will decide whether it is more probable that phoneme [a] or phoneme [œ] was sung.

The ASR has been adapted to handle musical information and works with very low delay [Loscos, Cano, Bonada, 1999] since the system cannot wait for a phoneme to be finished before it is recognized. Moreover, a phoneme has to be assigned to each frame. This would be a rather impossible/impractical situation if the lyrics of the song were not known beforehand. This constraint reduces the search problem: all the possible paths are restricted to just one string of phonemes, with several possible pronunciations. The problem is cut down to the question of locating the phoneme in the lyrics and placing the start and end points.

Besides knowing the lyrics, music information is also available. The user is singing along with the music, and hopefully according to a tempo and melody already specified in the score. Thus, we also know the time at which a phoneme is supposed to be sung, its approximate duration, its associated pitch, etc. All this information is used to improve the performance of the recognizer and also to allow resynchronization, for example in the case that the singer skipped a part of the song. The tempo information, for example, is used to modify the output Viterbi probabilities by the function shown in figure 8.

t_s is the time at which the phoneme happens in the singer performance, t_m is the time at which this phoneme happens in the song score, parameters *a* and *b* are tempo and duration dependent, and *alpha* has a very low value, nearly null. This function can be defined differently for the case in which the singer comes from silence and attacks the beginning of a note/word, and for the case where the singer has already started a note/word, due to the very different behaviors of these two situations.

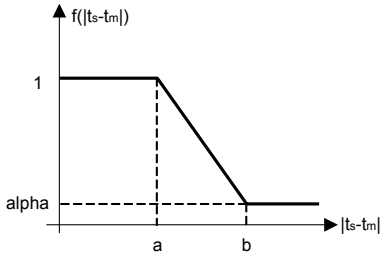

Fig 8 *Function of the factor applied to the Viterbi probability*

3.3 The impersonator singing voice model

The basic SMS analysis results in a simple parameterization appropriate for describing the inner properties of a sound, namely the instantaneous frequency, the amplitude and phase of each partial and the instantaneous spectral characteristics of the residual signal. Still, there are other useful instantaneous attributes that give a higher-level abstraction of the sound characteristics. These attributes are calculated at each analysis frame from the output of the basic sinusoidal plus residual analysis. [Serra, Bonada, 98]

The attributes we use for this application are: the amplitude of sinusoidal component, the amplitude of residual component, the fundamental frequency, the spectral shape of sinusoidal component, and the spectral shape of the residual component. These attributes are extracted from the frame data, leaving a normalized frame. This way, we can morph by interpolating the high-level attributes and later add these back to the synthesis frame. By carrying out the morph at the high-level plane, we have a more intuitive and musical control of the process and we minimize crossed effects between interpolations of different attributes.

The amplitude of the sinusoidal component is calculated as the sum of the amplitudes of all harmonics of the current frame expressed in dB,

$$AS_{total} = 20\log_{10}\left(\sum_{i=1}^{I} a_i\right) \tag{4}$$

where a_i is the linear amplitude of the i^{th} harmonic and I is the total number of harmonics found in the current frame.

The amplitude of the residual component is calculated as the sum of the absolute values of the residual of the current frame expressed in dB. This amplitude can also be computed by adding the frequency samples of the corresponding magnitude spectrum,

$$\begin{aligned} AR_{total} &= 20\ \log_{10}\left(\sum_{n=0}^{M-1}|x_R(n)|\right) \\ &= 20\ \log_{10}\left(\sum_{k=0}^{N-1}|X_R(k)|\right) \end{aligned} \tag{5}$$

where $x_R(n)$ is the residual sound, M is the size of the frame, $x_R(k)$ is the spectrum of the residual sound, and N is the size of the magnitude spectrum.

The fundamental frequency is defined as the frequency that best explains the harmonics of the current frame. This can be computed by taking the weighted average of all the normalized harmonic frequencies,

$$F_0 = \sum_{i=1}^{I} \frac{f_i}{i} \times \frac{a_i}{\sum_{i=1}^{I} a_i} \tag{6}$$

where f_i is the frequency of the i^{th} harmonic. A more complete discussion on the issue of fundamental frequency in the context of SMS can be found in [Cano, 98].

The spectral shape of the sinusoidal component is expressed as an envelope described by the amplitudes and frequencies of the harmonics, or its approximation,

$$SShape = \{(f_1, a_1)(f_2, a_2)\ldots(f_I, a_I)\} \tag{7}$$

This set of points that define the spectral shape envelope is joined with a third order spline interpolation instead of linear interpolation. Spline interpolation gives a better approximation of the resonant character of the spectrum of the singing voice.

For the sinusoidal component, we also store the phase envelope at the beginning of each period. This envelope is computed applying a time shift depending of the pitch found at each frame. This phase envelope is the one responsible of preserving the phase alignment.

However, whenever the spectral shape is interpolated, and the morph factor is set around 50%, the resulting spectral shape is smoothed and looses much of its timbre characteristics. This problem can be solved if we include anchor points (i.e. resonances) in the spectral shape model and we take them into account in the interpolation step as shown in figure 9.

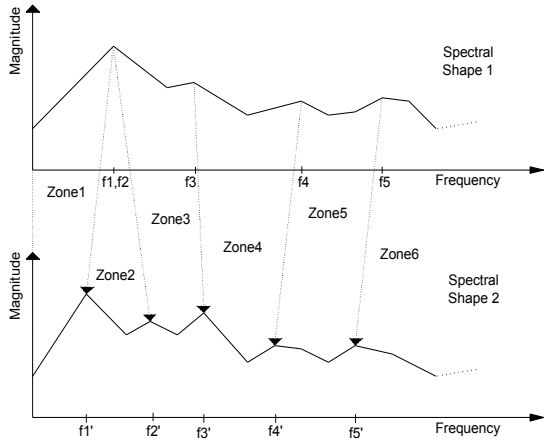

Fig 9 *Anchor points in the sinusoidal spectral shapes*

By using anchor points we can preserve the maximums and the valleys of the spectrum so that when going from one timbre to another, resonances would not appear and disappear but move from one frequency to another.

The spectral shape of the residual component is expressed as an approximation of the magnitude spectrum of the residual sound at the current frame. A simple function is computed as the line segment approximation of the spectrum,

$$Rshape = \left\{ e_1, e_2, \ldots, e_q, \ldots, e_{N/M} \right\} = \max_k \left[\left| X_R \left(qM + k \right) \right| \right] \quad (8)$$

where k=*-M/2,-M/2+1,...,...,M/2-1*, and M is the number of frequency samples used for each calculation of a local maximum.

For both the sinusoidal and residual components the magnitudes of the spectral shape envelope are interpolated whereas the phases are always taken either from the user or the target but never interpolated.

3.4 Discussion

In this section we have presented a singing voice morphing system. The final purpose of the system was to make an average singer sing any song like any professional singer. However the system has some weaknesses that arise from its specifications and disallow its practical commercialization.

At this point of time, in order for a user to sing a song like somebody else's voice, we need to get before hand a dry recording of the target singer singing the selected song. Since it is not easy to afford dry recordings of the most probable candidates to be chosen as targets (i.e. Frank Sinatra or Aretha Franklin), in this project we have used recordings of professional impersonators. Anyhow, the requirement pointed out here makes the system by no means efficient.

The inefficiency comes from two main intractable hitches. First, it entails an unreachable cost to have one popular singers or even his impersonator recording in a studio all the songs available in a karaoke database. Second, we need to analyze and store before hand the targets performances and this process generates such a huge amount of data it turns the database into something impossible to handle.

There are many solutions that can be thought to solve these problems. For example, we could think of a system in which the user could only sing Elvis's songs like Elvis, and would not allow possibilities such as singing "Walk like an Egyptian" like John Lennon. This would not only reduce the recording requirements but also would open the possibility of coming to an agreement with some record labels in order to get dry recordings of their most well-known singers. Also all sorts of data-compression techniques could be applied to the system to reduce the amount of data stored in the database.

Anyway, our future plan does not consider any of the mentioned propositions to be the one to concentrate on. We believe the solution implies accomplishing a more flexible system in which we would work with singer models rather than singer performances. The idea is to record the target singer singing all possible phonetics at different registers, with different intensities, in different emotion contexts, etcetera, sampling this way the space of all the possible phonetic and musical contexts. The analysis of these recordings would be the basis of a singer model from which we could later synthesize, out of the score and the lyrics of a song, a possible performance of the singer modeled. That is what brought us to the singing synthesizer application.

4. THE SINGING VOICE SYNTHESIZER

We have developed a source-filter type singing synthesizer application based on the sinusoidal plus residual decomposition of the sound. This system generates a performance of an artificial singer out of the musical score and the phonetic transcription of a song.

Mimicking the performance of a singer using computer synthesis tools is a very ambitious and complex goal and there are many parameters at different levels that affect the naturalness of the resulting synthesized sound. The system we present takes care of it at two main levels: the expressiveness level and the synthesis level. The expressiveness level is the one in charge of reproducing the musical and emotional expressions. The synthesis level is in charge of reproducing the generation of a human singing voice signal. This synthesis level is the one we present next.

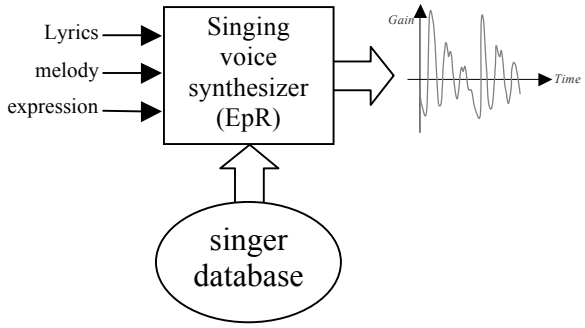

Fig 10 *General diagram of the singing voice synthesizer*

In figure 10 we can see the general diagram of our singing voice synthesizer. The inputs of the system are the lyrics (the phonetic transcription), the melody to be sung and optionally some expression parameters. On the other hand, the system is also fed by a singer database. This database holds the voice characteristics and is created from the analysis of singer's recordings.

4.1 The EpR voice model

Our singing voice synthesizer is based on an extension of the well known source/filter approach [Childers, 94] we call EpR (***E**xcitation **p**lus **R**esonances)*. The excitation can be either voiced, unvoiced, or a combination of both. Besides, in the case of a voiced phonation we model a harmonic source plus a residual source. In figure 11 we can see a graphic with the three types of excitation generated in our system and the corresponding filters. For the case of a voiced phonation, the filter applied to each excitation tries to mimic the singing voice spectral shape using a frequency domain filtering function that can be decomposed into two filters in cascade: an exponential decay curve plus several resonances. After filtering, the voiced residual excitation needs to be transposed because it is a filtered SMS residual recording and

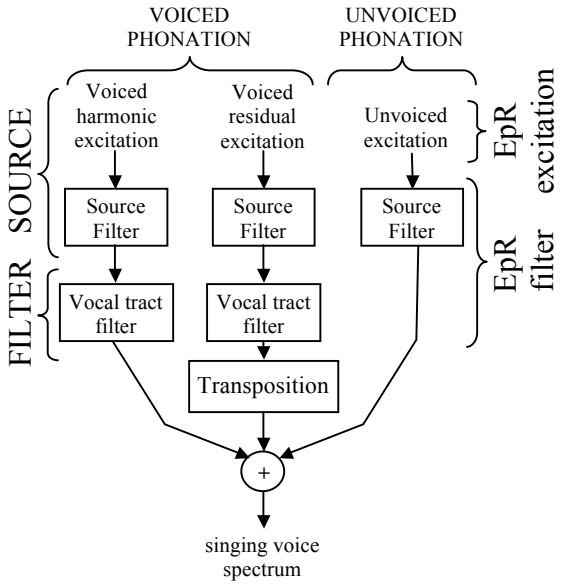

Fig 11 *The EpR voice model*

has traces of the original pitch. Otherwise, in the case of an unvoiced phonation, we apply a filter that just changes the tilt curve and the gain of the STFT of an original recording.

4.1.1 The EpR excitation

Voiced harmonic excitation

The inputs that control the voiced harmonic excitation generation are the desired pitch and gain envelopes. The resulting excitation signal is obtained by generating a delta train in the time domain thus allowing to achieve period resolution and to use some simple excitation templates. This can be useful to generate jitter or different types of vocal disorders. This delta train can be seen as a glottal source wave previous to a convolution with the differentiated glottal pulse.

A fractional delay filter is needed to position the excitation deltas between samples, since we have to go beyond the sampling rate resolution. The filter is implemented using a windowed sinc like function situated at each pulse location with the offset subtracted [Smith, Gosset, 1984]. Finally the windowing and the fft are applied. The result is a spectrum approximately flat that contains the harmonics approximately synchronized in phase. If no excitation template is applied, the spectrum will be perfectly flat and the phase synchronization precise.

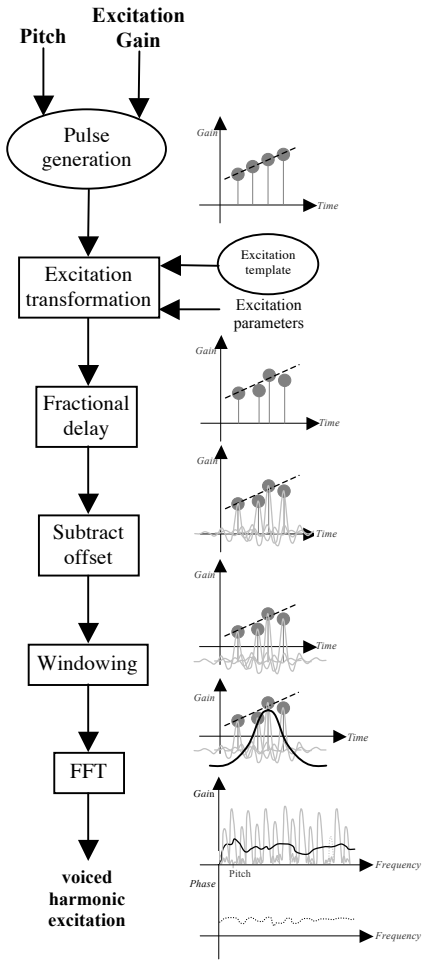

Fig 12 *The EpR voiced harmonic excitation*

Voiced residual excitation

The voiced residual excitation is obtained from the residual of the SMS analysis of a long steady state vowel recorded from a real singer. The SMS residual is inverse-filtered by its short-time average spectral shape envelope to get an approximately flat excitation magnitude spectrum.

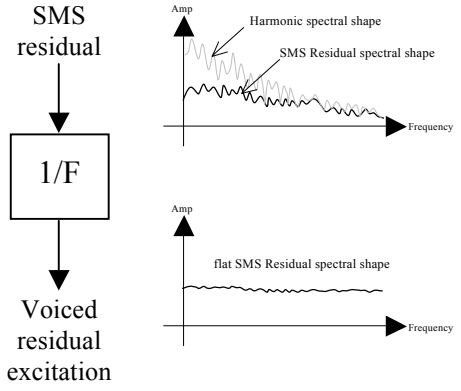

Fig 13 *The voiced residual excitation*

Unvoiced excitation
The excitation in the unvoiced parts is left unmodeled, using directly the original recording of a singer's performance.

4.1.2 The EpR filter

The EpR filter can be decomposed in two cascade filters. The first of them models the differentiated glottal pulse frequency response, and the second the vocal tract (resonance filter).

The EpR source filter
The EpR source is modeled as a frequency domain curve and one source resonance applied to the input frequency domain flat excitation described in the previous section. This source curve is defined by a gain and an exponential decay as follows:

$$Source_{dB} = Gain_{dB} + SlopeDepth_{dB}\left(e^{Slope \cdot f} - 1\right) \quad (9)$$

This curve is obtained from an approximation to the harmonic spectral shape (*HSS*) determined by the harmonics identified in the SMS analysis

$$HSS(f) = envelope_{i=0..n}\left[f_i, 20\log(a_i)\right] \quad (10)$$

where i is the index of the harmonic, n is the number of harmonics, f_i and a_i are the frequency and amplitude of the i^{th} harmonic.

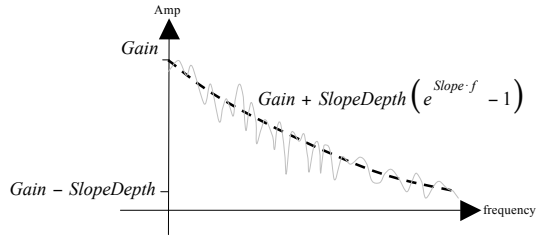

Fig 14 *The EpR source curve*

On top of the source curve, we add a second resonance in order to model the low frequency content of the spectrum below the first formant. This resonance affects the synthesis in a different way than the vocal tract resonances, as will be explained later.

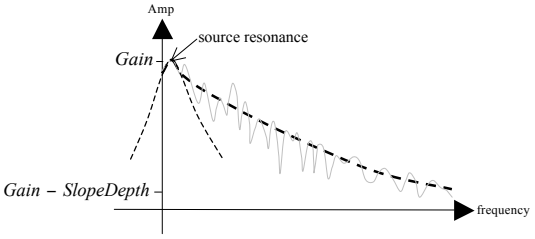

Fig 15 *The EpR source resonance*

The source resonance is modeled as a symmetric second order filter (based on the Klatt formant synthesizer [Klatt, 1980]) with center frequency *F*, bandwidth *Bw* and linear amplitude *Amp*. The transfer function of the resonance *R(f)* can be expressed as follows

$$H(z) = \frac{A}{1 - Bz^{-1} - Cz^{-2}}$$

$$R(f) = Amp\,\frac{H\left(e^{j2\pi\left(0.5+\frac{f-F}{fs}\right)}\right)}{H\left(e^{j\pi}\right)} \quad (11)$$

where

fs = Sampling rate

$$C = -e^{-\frac{2\pi Bw}{fs}}$$

$$B = 2\cos(\pi)e^{-\frac{\pi Bw}{fs}}$$

$$A = 1 - B - C$$

The amplitude parameter (*Amp*) is relative to the source curve (a value of 1 means the resonance maximum is just over the source curve).

The EpR vocal tract filter
The vocal tract is modeled by a vector of resonances plus a differential spectral shape envelope. It can be understood as an approximation to the vocal tract filter. These filter resonances are modeled in the same way as the source resonance (see eq. 11), where the lower frequency resonances are somewhat equivalent to the vocal tract formants.

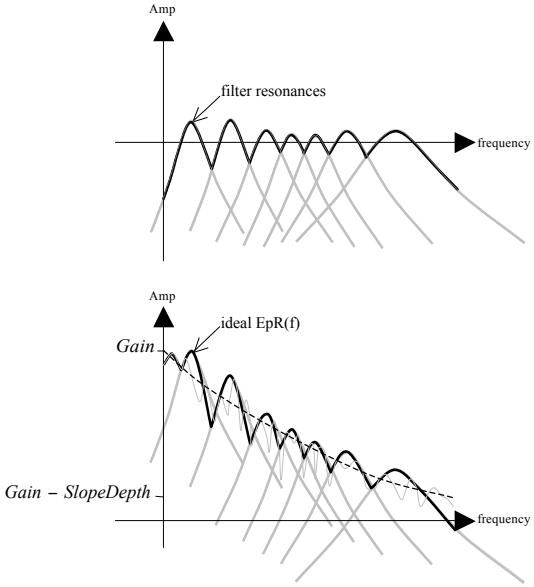

Fig 16 *The EpR filter resonances*

The EpR filters for voiced harmonic and residual excitations are basically the same, but just differ in the gain and slope depth parameters. This approximation has been obtained after comparing the harmonic and residual spectral shape of several SMS analysis of singer recordings. Figure 17 shows these differences.

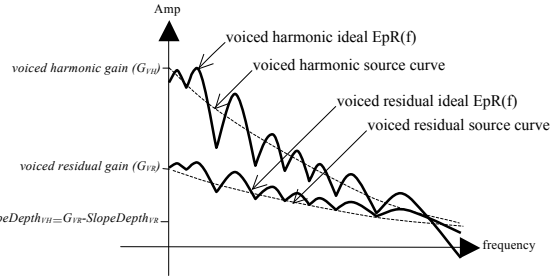

Fig 17 *Differences between harmonic and residual EpR filters*

The differential spectral shape envelope actually stores the differences (in dB) between the ideal EpR model (*iEpR*) and the

real harmonic spectral shape (*HSS*) of a singer's performance. We calculate it as a 30 Hz equidistant step envelope.

$$DSS(f) = envelope_{i=0..}\left[30i, HSS_{dB}(30i) - iEpR_{dB}(30i)\right] \quad (12)$$

The EpR phase alignment

The phase alignment of the harmonics at the beginning of each period is obtained from the EpR spectral phase envelope. A time shift is applied just before the synthesis, in order to get the actual phase envelope at the synthesis time (usually it will not match the beginning of the period). This phase alignment is then added to the voiced harmonic excitation spectrum phase envelope. The EpR spectral phase model states that each vocal tract resonance produces a linear shift of π on the flat phase envelope with a bandwidth depending on the estimated resonance bandwidth. This phase model is especially important for the intelligibility and in order to get more natural low pitch male voices.

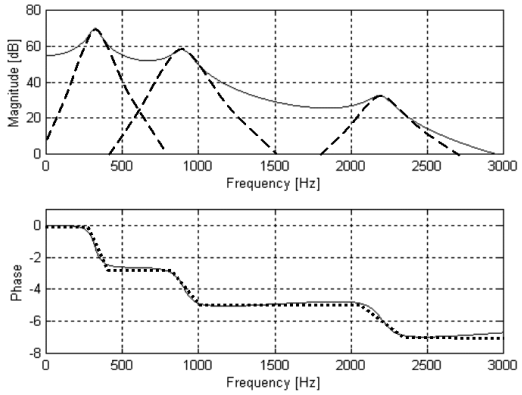

Fig 18 *The phase alignment is approximated as a linear segment, with a phase shift for each resonance*

The EpR filter implementation

The EpR filter is implemented in the frequency domain. The input is the spectrum that results out from the voiced harmonic excitation or from the voiced residual excitation. Both inputs are supposed to be approximately flat spectrums, so we just need to add the EpR resonances, the source curve and the differential spectral shape to the amplitude spectrum. In the case of the voiced harmonic excitation we also need to add the EpR phase alignment to the phase spectrum.

For each frequency bin we have to compute the value of the EpR filter. This implies a considerable computational cost, because we have to calculate the value of all the resonances. However, we can optimize this process by assuming that the value of the sum of all the resonances is equal to the maximum amplitude (dB) of all the filter and excitation resonances (over the source curve). Then we can even do better by only using the two neighbors' resonances for each frequency bin. This is not a low-quality approximation of the original method because the differential spectral shape envelope takes care of all the differences between the model and the real spectrum.

If we want to avoid the time domain voiced excitation, especially because of the computational cost of the fractional delay and the FFT, we can change it to be directly generated in the frequency domain. From the pitch and gain input we can generate a train of deltas in frequency domain (sinusoids) that will be convolved with the transform of the synthesis window and then synthesized with the standard frame based SMS synthesis, using the IFFT and overlap-add method. However the voice quality may suffer some degradation due to the fact that the sinusoids are assumed to have constant amplitude and frequency along the frame duration.

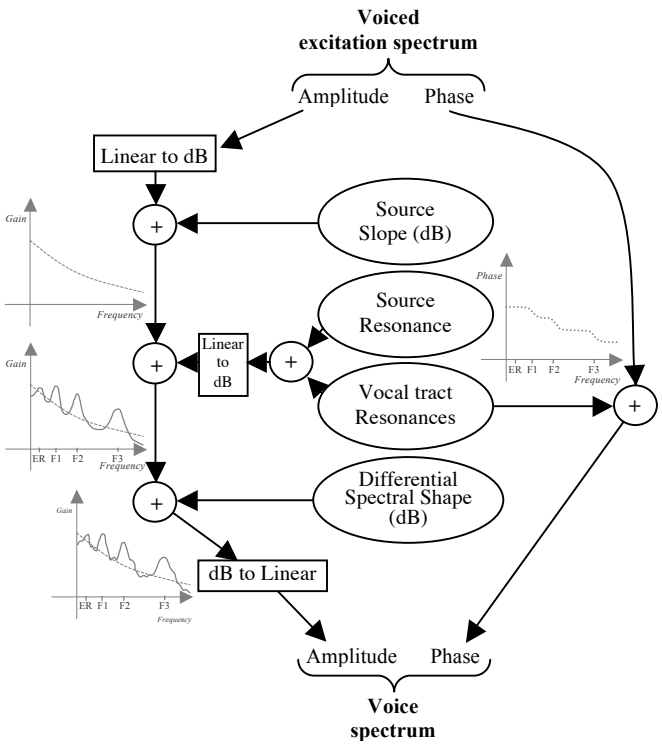

Fig 19 *Frequency domain implementation of the EpR model*

The EpR filter transformation

We can transform the EpR by changing its parameters:

- excitation gain, slope and slope depth
- frequency, amplitude and bandwidth of the resonances

However, we have to take into account that the differential spectral shape is related to the resonances position. Therefore, if we change the frequency of the resonances we should stretch or compress the differential spectral shape envelope according to the resonances frequency change (using the resonances center frequency as anchor points).

4.2 The singer database

The voice characteristics of a real singer are stored in a database. About one hour of recording is needed in order to build one singer database. All the audio data must be recorded in a dry and noiseless environment. The singer database is divided in two parts: timbre DB and voice DB.

4.2.1 The Timbre DB

The timbre DB stores the voice model (EpR) for each of the voiced phonemes. For each of these, we store several samples at different pitches and at different dynamics. When a phoneme with intermediate pitch or dynamics is required, the EpR parameters of the neighboring samples are interpolated.

4.2.2 The Voice DB

The voice DB represents and stores the time varying characteristics of the voice. It is divided in several categories: steady states, phonetic articulations, note attacks, note-to-note articulations, note

releases and vibratos. It is possible to have different templates at different pitches for each of them.

Steady states
There are steady states stored for each of the phonemes. They model the behavior of the stationary part. To do so, they store the time-varying evolution of the EpR parameters (if the phoneme is voiced) and the SMS residual along the steady state.

Phonetic articulations
All the articulations are segmented in two regions. If the regions are different in terms of voiceness, each of the regions is analyzed with different specific parameters. Otherwise, if both regions are voiced or unvoiced the segmentation is used in synthesis to know the onset of the articulation. The behavior of the EpR model is estimated only in the voiced part regions.

Note attacks, note-to-notes and note releases
They model the excitation behavior along their duration. They are phoneme independent since they do not model the EpR changes. They can be organized into different musical evocative classification.

Vibratos
They define the behavior of the source changes along a vibrato. In particular, they track the pitch, gain and source curve changes. Each of them is segmented into attack, body and release. There can be different template labels according to some musical or expression classification.

4.3 Results

A first prototype of the singing voice synthesizer has been implemented. In order to evaluate the prototype, the synthesis obtained from our system was compared with the synthesis obtained from commercial singing synthesizers. Although the synthesis obtained was not comparable with a real singer performance, it resulted more natural than the commercial software synthesis.

However, the system presents important drawbacks. Unnatural artifacts appear in the synthesis, especially in the voiced consonants phonemes. These artifacts emerge from the fact our synthesis engine is built on top of a sinusoidal plus residual decomposition. For these phonemes, it is difficult to determine what should be included in the sinusoidal component and what not. The low register timbres of a male voice suffer from unnaturalness. This problem may come from the simplicity of the EpR phase model. Sharp attacks, specially the ones belonging to plosive phonemes are smeared. This is due to the fact the sinusoidal model can not deal with unstable partials in an accurate manner. Some solutions to this problem are proposed in [Fitz, Haken, Christensen, 2000; Verma, Meng, 2000].

So there is room for improvement in every step of the system but presuming naturalness is the essential feature we take into account whenever we evaluate the quality of a singing synthesizer, results are promising and prove the suitability of our approach.

References

Cano, P. 1998. "Fundamental Frequency Estimation in the SMS Analysis". *Proceedings of the Digital Audio Effects Workshop (DAFX98)*, 1998.

Cook, P. R. 1996. "Singing Voice Synthesis: History, Current Work, and Future Directions." *Computer Music Journal*, 20(3).

M. de Boer; J. Bonada; Cano, P.; A. Loscos; X. Serra; 2000. "Singing voice impersonator for PC." *Proceedings of the 2000 International Computer Music Conference*. San Francisco: Computer Music Association.

Cano, P.; A. Loscos; J. Bonada; M. de Boer; X. Serra; 2000. "Voice Morphing System for Impersonating in Karaoke Applications." *Proceedings of the 2000 International Computer Music Conference*. San Francisco: Computer Music Association.

Childers, D.G. 1994. "Measuring and Modeling Vocal Source-Tract Interaction". *IEEE Transactions on Biomedical Engineering 1994*.

Fitz, K.; L. Haken, and P. Christensen. 2000. "A New Algorithm for Bandwidth Association in Bandwidth-Enhanced Additive Sound Modeling". *Proceedings of the 2000 International Computer Music Conference*. San Francisco: Computer Music Association.

Klatt, D.H. 1980. "Software for a cascade/parallel formant synthesizer" *Journal Acoustics of American Society*, 971-995.

Loscos, A.; Cano, P.; Bonada, J. 1999. "Low Delay Singing Voice Alignment to text" *Proceedings of the ICMC 1999*

Osaka, N. 1995. "Timbre Interpolation of sounds using a sinusoidal model", *Proceedings of the ICMC 1995*.

Serra, X. and J. Smith. 1990. "Spectral Modeling Synthesis: A Sound Analysis/Synthesis System based on a Deterministic plus Stochastic Decomposition.'' *Computer Music Journal* 14(4):12--24.

Serra, X. 1994. "Sound Hybridization Techniques based on a Deterministic plus Stochastic Decomposition Model." *Proceedings of the 1994 International Computer Music Conference*. San Francisco: Computer Music Association.

Serra, X. 1996. "Musical Sound Modeling with Sinusoids plus Noise", in G. D. Poli, A. Picialli, S. T. Pope, and C. Roads, editors, *Musical Signal Processing*. Swets & Zeitlinger Publishers.

Serra, X.; J. Bonada. 1998. "Sound Transformations Based on the SMS High Level Attributes", *Proceedings of the 98 Digital Audio Effects Workshop*.

Settel, Z., C. Lippe. 1996. "Real-Time Audio Morphing", *7th International Symposium on Electronic Art*, 1996.

Slaney, M., M. Covell, B. Lassiter. 1996. "Automatic audio morphing", *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.* 2, 1001-1004 (1996).

Smith, J.O., Gosset, P. 1984. "A flexible sampling-rate conversion method", *Proceedings of the ICASSP*, San Diego, New York, vol. 2, pp 19.4.1-19.4.2. IEEE Press.

Tellman, E., L. Haken, B. Holloway. 1995. "Timbre Morphing of Sounds with Unequal Number of Features", *J. Audio Eng. Soc.*, 43:9 1995.

Tellman, E.; L. Haken; B. Holloway 1995. "Timbre morphing of sounds with unequal numbers of features". *Journal of the Audio Engineering Society*, 43(9), 678-89.

Verma, T. S. ; T. H. Y. Meng. 2000. "Extending Spectral Modeling Synthesis With Transient Modeling Synthesis", *Computer Music Journal* 24:2, pp.47-59.