Spectral Modeling for Higher-level Sound Transformations

Xavier Amatriain, Jordi Bonada, Alex Loscos, Xavier Serra
Music Technology Group, Pompeu Fabra University
{xavier.amatriain, jordi.bonada, alex.loscos, xavier.serra}@iua.upf.es,
http://www.iua.upf.es/mtg

Abstract

When designing audio effects for music processing, we are always aiming at providing higher-level representations that may somehow fill in the gap between the signal processing world and the end-user. Spectral models in general, and the Sinusoidal plus Residual model in particular, can sometimes offer ways to implement such schemes.

1 Introduction

When dealing with digital audio effects, we are looking for representations of sound signals and signal processing systems that can give us ways to design sound transformations in a variety of music applications and contexts.

The basic idea of spectral processing is that we can analyze a sound to obtain alternative frequency domain representations, which can then be transformed and inverted to produce new sounds (see Figure 1). Most of the approaches start by developing an analysis/synthesis system from which the input sound is reconstructed without any perceptual loss of sound quality. Then, the main issue is what the intermediate representation is and what parameters are available for applying the desired transformations.

By understanding the basic concepts of frequency domain analysis, we will be able to acquire the tools to use a large number of effects processors and to understand many types of sound transformations systems. Moreover, being the frequency domain analysis a somewhat similar process than the one performed by the human hearing system, it yields fairly intuitive intermediate representations.

Perceptual or musical concepts such as timbre or pitch are clearly related to the spectral characteristics of a sound. Some common processes for sound effects are also better explained using a frequency domain representation. We usually think on the frequency axis when we talk about equalizing, filtering, pitch shifting, harmonizing... In fact, some of them are specific to this signal processing approach and do not have an immediate counterpart on the time domain. Another issue is whether or not this approach is the most efficient, or practical, for a given application. The process of transforming a time domain signal into a frequency domain representation is, by itself, not an immediate step. Some parameters are difficult to adjust and force us to take several compromises. Some settings, such as the size of the analysis window, have little or nothing to do with the high-level approach we intend to favor, and require the user to have a basic signal processing understanding.

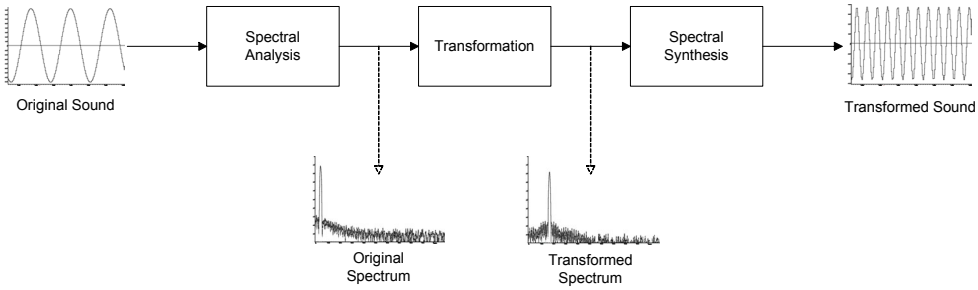

Fig 1. Block diagram of a simple spectral analysis

In that sense, when we talk about higher level spectral processing we are thinking of an intermediate analysis step in which relevant features are extracted or computed from the spectrum. These relevant features should be much closer to a musical or high-level approach. We can then process the features themselves or even apply transformations that keep some of the features unchanged. For

example, we can extract the fundamental frequency and the spectral shape from a sound and then modify the fundamental frequency without affecting the shape of the spectrum. [Serra and Bonada 98]

In this section, we will briefly mention the Sinusoidal Model and will concentrate in the Sinusoidal plus Residual Model. Anyhow, the decision as to what spectral representation to use in a particular situation is not an easy one. The boundaries are not clear and

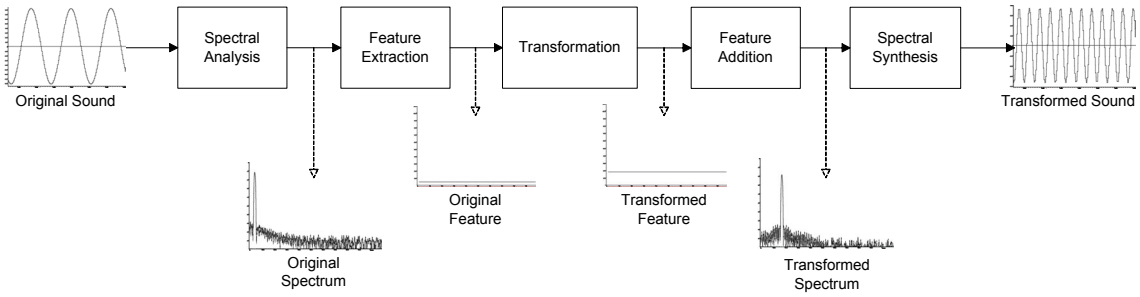

Fig 2. Block diagram of a higher-level spectral processing framework

Assuming the fact that there is no single representation and processing system optimal for everything, our approach will be to present a set of complementary spectral models that can be combined to be used for the largest possible set of sounds and musical applications.

In the next sections, we introduce two spectral models: Sinusoidal and Sinusoidal plus Residual. These models already represent a step up on the abstraction ladder and, from either of them, we can identify and extract higher-level information of a sound, such as: harmonics, pitch, spectral shape, vibrato, or note boundaries, that is Higher Level Features. This analysis step brings the representation closer to our perceptual understanding of a sound. The complexity of the analysis will depend on the type of feature that we want to identify and the sound to analyze. The benefits of going to this higher level of analysis are enormous and open up a wide range of new musical applications.

In section 3 we will provide a set of basic audio effects and transformations based on the implemented Sinusoidal plus Residual analysis/synthesis. We will finish with an explanation of content dependant processing implementations: a real-time singing voice conversion application that has been developed for use in Karaoke applications, and a nearly loss less Time Scaling algorithm.

2 Spectral Models

The most common approach for converting a time domain signal into its frequency domain representation is the Short-Time Fourier Transform (STFT). It is a general technique from which we can implement loss-less analysis/synthesis systems. Many sound transformation systems are based on direct implementations of the basic algorithm.

there are always compromises to take into account, such as: (1) sound fidelity, (2) flexibility, (3) coding efficiency, and (4) computational requirements. Ideally, we want to maximize fidelity and flexibility while minimizing memory consumption and computational requirements. The best choice for maximum fidelity and minimum compute time is the STFT that, anyhow, yields a rather inflexible representation and inefficient coding scheme. Thus our interest in finding higher-level representations as the ones we present in this section.

2.1 Sinusoidal Model

Using the output of the STFT, the Sinusoidal model represents a step towards a more flexible representations while compromising both sound fidelity and computing time. It is based on modeling the time-varying spectral characteristics of a sound as sums of time-varying sinusoids. The input sound is modeled by,

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

where $A_r(t)$ and $\theta_r(t)$ are the instantaneous amplitude and phase of the r^{th} sinusoid, respectively. [McAulay and Quatieri 86; Smith and Serra 87].

To obtain a sinusoidal representation from a sound, an analysis is performed in order to estimate the instantaneous amplitudes and phases of the sinusoids. This estimation 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 finally organizing them as time-varying sinusoidal tracks.

It is a quite general technique that can be used in a wide range of sounds and offers a gain in flexibility compared with the direct STFT implementation.

2.2 Sinusoidal plus Residual Model

The Sinusoidal plus Residual model can cover a wide "compromise space" and can in fact be seen as the generalization of both the STFT and the Sinusoidal models. Using this approach, we can decide what part of the spectral information is modeled as sinusoids and what is left as STFT. With a good analysis, the Sinusoidal plus Residual representation is very flexible while maintaining a good sound fidelity and the representation is quite efficient. In this approach, the Sinusoidal representation is used to model only the stable partials of a sound. The residual, or its approximation, models what is left, which should ideally be a stochastic component. This model is less general than either the STFT or the Sinusoidal representations but it results in an enormous gain in flexibility [Serra 96; Serra and Smith 90].

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

$$s(t) = \sum_{r=1}^{R} A_r(t)\cos\left[\theta_r(t)\right] + e(t) \qquad (2)$$

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

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

$$\theta_r(t) = \int_0^t \omega_r(\tau)d\tau \qquad (3)$$

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.

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,

$$e(t) = \int_0^t h(t,\tau)u(\tau)d\tau \qquad (4)$$

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 implementation of the analysis for the Sinusoidal plus Residual Model is more complex than the one for the Sinusoidal Model. Figure 3 shows a simplified block- diagram of this analysis.

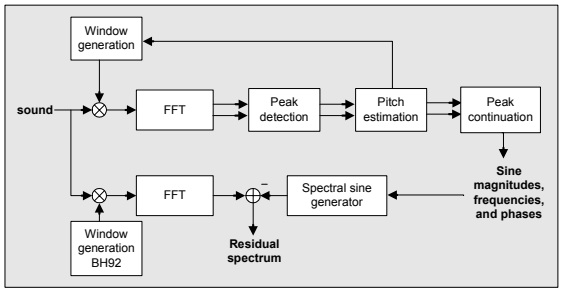

Fig 3. Block diagram of the Sinusoidal plus residual analysis.

The first few steps are the same than in a sinusoidal-only analysis. The major differences start on the peak continuation process since in order to have a good partial-residual decomposition we have to refine the this process in such a way as to be able to identify the stable partials of the sound.

The residual component is obtained by first generating the sinusoidal component with additive synthesis, and then subtracting it from the original waveform. A spectral analysis of this time domain residual is done by first windowing it using a window which is independent of the one used to find sinusoids, and thus we are free to choose a different time-frequency compromise. Then the FFT is computed and the resulting spectrum can be modeled using several existing techniques.

The original sinusoidal plus residual model has led to other different spectral models that still share some of its basis. [Ding and Qian, 97; Fitz, Haken and Christensen, 00; Verma,00]

2.2.1 Feature analysis

The accomplishment of a meaningful parameterization for sound transformation applications is a difficult task. We want a parameterization that offers an intuitive control over the sound transformation process, with which we can access most of the perceptual attributes of a sound. The analysis techniques described so far result in a simple parameterization, appropriate for describing

the lower physical characteristics of the sound. In the Sinusoidal plus Residual model, these parameters are the instantaneous frequency, amplitude and phase of each partial and the instantaneous spectral characteristics of the residual signal.

There are other useful instantaneous attributes that give a higher-level abstraction of the sound characteristics. For example we can describe fundamental frequency, amplitude and spectral shape of sinusoidal component, amplitude and spectral shape of residual component, and overall amplitude. These attributes are calculated at each analysis frame from the output of the basic Sinusoidal plus Residual analysis. Afterwards, some of them can be extracted.

From a digital effects designer point of view, the extraction of such attributes allows us to implement transformations that modify only one of those features without affecting the rest. A clear example is illustrated in Fig 2 where the fundamental frequency is extracted, multiplied by a scaling factor, and then incorporated back to the original spectral data.

Apart from the instantaneous, or frame, values, it is also useful to have parameters that characterize the time evolution of the sound. The time changes can be described by the derivatives of each one of the instantaneous attributes computed as follows:

$$\Delta = \frac{Val(l) - Val(l-1)}{H/SR} \quad (5)$$

where $Val(l)$ is the attribute value for the current frame, $Val(l-1)$ is the attribute value for the previous one, H is the hop-size and SR the sampling rate.

Another important step towards a musically useful parameterization is the segmentation of a sound into regions that are homogeneous in terms of its sound attributes. Then we can identify and extract region attributes that will give higher-level control over the sound. For our purposes it is very valuable as a way to apply region dependent transformations. For example, a time stretching algorithm would be able to transform the steady state regions, leaving the rest unmodified.

Once a given sound has been segmented into regions we can compute the attributes that describe each one. Most of the interesting attributes are the mean and variance of each of the frame attributes for the whole region.

Global attributes that can characterize attacks and releases make use of the average variation of each of the instantaneous attributes, such as average fundamental frequency variation, average amplitude variation, or average spectral shape change. In the steady state regions it is important to extract the average value of each of the instantaneous attributes and measure other global attributes such as time-varying rate and depth of vibrato.

Some region attributes can be extracted from the frame attributes in the same way that these were extracted from the Sinusoidal plus Residual data. The result of the extraction of the frame and region attributes is a hierarchical multi-level data structure where each level represents a different sound abstraction.

From the basic sinusoidal plus residual representation it is possible to extract some of the attributes mentioned above. The critical issue is how to extract them while minimizing interferences, thus obtaining significant high level attributes free of correlations [Serra and Bonada 98]. The general process will be to first extract instantaneous attributes and their derivatives, then segment the sound based on that information, and finally extract region attributes.

2.2.2 Synthesis

From the output of the analysis techniques presented we can synthesize a new sound. The similarity with respect to the original sound will depend on how well the input sound fits the implicit model of the analysis technique and the settings of the different variables that the given technique has. In the context of this paper we are interested in transforming the analysis output in order to produce a specified effect in the synthesized sound.

All these transformations can be done in the frequency domain. Afterwards, the output sound can be synthesized using the techniques presented in this section. The sinusoidal component will be generated using some type of additive synthesis approach and the residual, if present, will be synthesized using some type of subtractive synthesis approach.

Thus, the transformation and synthesis of a sound is done in the frequency domain; generating sinusoids, noise, or arbitrary spectral components, and adding them all to a spectral frame. Then, we compute a single IFFT for each frame, which can yield efficient implementations.

Fig 4 shows a block diagram of the final part of the synthesis process. Previous to that, we have to transform and add all the High Level Features, if they have been extracted, and obtain the lower level data (sine and residual) for the frame to be synthesized. Since the stored data might have a different frame rate, or a variable one, we may also have to generate the appropriate frame by interpolating the stored ones.

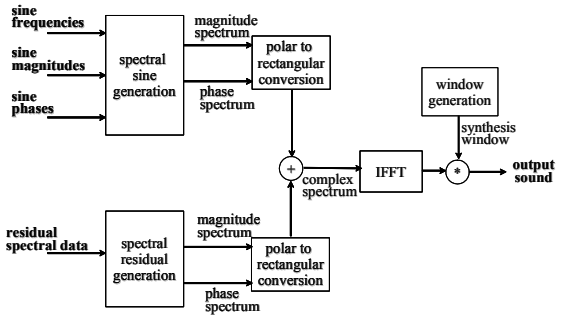

Fig.4. Diagram of the spectral synthesis.

3 FX and Transformations

In this section we intend to give a brief catalog of effects that can be implemented using the Sinusoidal plus Residual model and a transformation scheme like the one depicted in Figure 3.

3.1 Filtering with arbitrary resolution

Filters are probably the paradigm of a "classical" effect. Many different implementations are provided in the general DSP literature. Here we introduce a different approach that differs in many aspects from the classical one.

For our "filter" implementation, we take advantage of the sinusoidal plus residual model in order to modify the amplitude of any arbitrary partial present in the sinusoidal component.

For example, we can implement a band-pass filter defined by (x,y) points where x is the frequency value in Hertzs and y is the amplitude factor to apply. Thus, our filter does not need to be characterized by a traditional transfer function, and a more complex function can be defined by summing delta-functions. The transfer function is the defined by:

$$H(f) = \sum \delta(f_i) \cdot g_i \qquad (6)$$

where g_i is the gain applied to the i^{th} partial of frequency f_i.

Many applications derive from this sort of filtering scheme. For instance, we can filter out the even partials of a sound with a broadband spectrum, like a vocal sound, converting it to a clarinet-like sound.

3.2 Partial dependent frequency scaling

Another possible related effect that arises from the particularities of our model is the possibility to apply a frequency scaling to the sinusoidal components of our modeled sound, being able to process the residual or noisy component in a completely different way.

We can, for example, introduce a frequency shift factor to all the partials of our sound (see formula 7). Note, though, that if a constant is added to every partial of a harmonic spectrum, the resulting sound will be inharmonic.

$$f_i = f_i + k \qquad (7)$$

In the same way, we can scale all the partials multiplying them by a given scaling factor. Note that this effect will act as a pitch shifter without timbre preservation.

$$f_i = f_i \cdot k \qquad (8)$$

Another effect we can implement following this same idea is to add a stretching factor to the frequency of every partial. The relative shift of every partial can be computed, for exaple, depending on its original partial index, following the formula:

$$f_i = f_i \cdot fstretch^{(i-1)} \qquad (9)$$

with $i=1..N$ where N is the number of sinusoids. This kind of stretching can be observed in a real piano sound. Thus, if we ever intended to create a piano synthesizer we could make use of this transformation.

3.3 Pitch Transposition with Timbre Preservation

Pitch transposition is the scaling of all the partials of a sound by the same multiplying factor. Here we introduce the concept of timbre preservation by leaving the spectral shape unmodified. For that reason we scale the frequency of each partial applying the original spectral shape.

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

$$Sshape = \{(f_1, a_1)(f_2, a_2)...(f_N, a_N)\} \qquad (10)$$

And so, the resulting sinusoidal spectrum $X_{transp}(f)$ after a transposition of value k will be of the form:

$$X_{transp}(f) = \sum \delta(k \cdot f_i) \cdot Sshape(k \cdot f_i) \qquad (11)$$

formula that obviously implies that we must be able to find intermediate points of the spectral shape using some interpolation algorithm.

An overall more realistic effect is accomplished if we comb filter the original residual using the new pitch

before we merge it with the transposed sinusoidal part.

3.4 Pitch Discretization to Temperate Scale

An interesting effect can be accomplished by forcing the pitch to take the nearest frequency value of the temperate scale. It is indeed a very particular case of pitch transposition where the pitch is cuantified to one of the 12 semitones in which an octave is divided. This effect is widely used on vocal sounds for dance music and is many times referred to with the misleading name of *vocoder effect*.

For a perfectly harmonic sound we have:

$$f_i = f_0 \cdot i \qquad (12)$$

with $i=1..N$ where N is the number of sinusoids. The frequency of the i^{th} harmonic is just i times the frequency of the fundamental f_0. To find the new fundamental frequency, we apply the following formula:

$$f_0' = 55 \cdot \left(\left(2^{(1/12)} \right)^{\left(round\left(\frac{12 \cdot \log\left(f_0 / 55 \right)}{\log(2)} \right) \right)} \right) \qquad (13)$$

where 55 is the frequency in *Hz* that corresponds to an A0. From this new fundamental f_0', we can compute the transposition factor, defined as:

$$k = \frac{f_0'}{f_0} \qquad (14)$$

Finally, we only need to apply the pitch transposition algorithm defined in the previous section.

3.5 Vibrato and Tremolo

Vibrato and tremolo are common effects used in different kinds of acoustical instruments, including the human voice. Both are low frequency modulations: vibrato is applied to the frequency and tremolo to the amplitude of the partials.

In fact, vibrato usually implies a tremolo modulation. Both modulations share in this case the resonating frequency f_m. For example, to apply a vibrato with tremolo effect, we can modulate the fundamental frequency f_0 (as shown in formula 15):

$$f_0' = f_0 \cdot (1 + c \cdot \sin(2\pi \cdot f_m)) \qquad (15)$$

where c is the vibrato depth (usually around 75 cent). This would give us a transposition factor (formula 14) and we would apply the corresponding transposition with timbre preservation. For the tremolo we could apply the following modulation to each of the siunoids amplitude a_i:

$$a_i' = a_i + t(i) \cdot \frac{\sin(2\pi \cdot f_m) - 1}{2} \text{ (dB)} \qquad (16)$$

the modulation depth $t(i)$ would apply different depths at different frequencies emulating the spectral tilt variations suffered in a real tremolo sound. This curve could be a sampled version of the curve shown in Fig.5 where *SR* is the sampling rate.

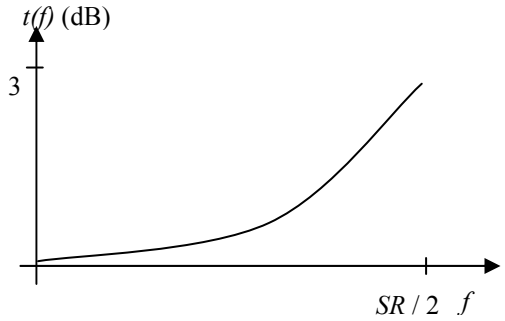

Fig. 5. *t(f)* curve

3.6 Spectral Shape Shift

Many interesting effects can be accomplished by shifting the spectral shape or spectral envelope of the sinusoidal component of a sound. This shift is performed in such a way that no new partials are generated, just the amplitude envelope of the spectrum is modified.

Such transformation results in a sinusoidal spectrum $X_{shifted}(f)$ such as:

$$X_{shifted}(f) = \sum \delta(f_i) \cdot Sshape(f_i - D) \qquad (17)$$

where D is the spectral shift in *Hz* (positive if we move the spectrum to the right and negative if we move it to the left) and *Sshape* is the original spectral shape and it is defined as in equation 10.

3.7 Gender Change

Using the results of 3.3 and 3.6 we can change the gender of a given vocal sound. Note how by combining different "basic" effects we are able to step higher in the level of abstraction and get closer to what a naive user could ask for in a sound

transformation environment, such as: imagine having a gender control on a vocal processor...

We apply two transformations in order to convert a male voice into a female one. The first one is a pitch transposition an octave higher. The other one is a shift in the spectral shape. The theoretical explanation to this effect is that women change their formant (resonant filters) frequencies depending on the pitch. That is, when a female singer rises up the pitch, the formants move along with the fundamental.

To convert a female into a male voice we also apply a pitch transposition and a shift in the spectral shape. This shifting has to be applied in a way the formants of the female voice remain stable along different pitches

3.7 Harmonizer

In order to create the effect of a harmonizing vocal chorus, we can add pitch-shifted versions of the original voice (with the same timbre) and force them to be in tune with the original melody.

So for a number *H* of harmonies we have that the resulting sinusoidal spectrum can be described as:

$$X'(f) = X(f) + \sum_{h=1}^{H} X_{transp}(f,h) \quad (18)$$

where $X_{transp}(f,h)$ is the original sinusoidal spectrum transposed (with timbre preservation) by a factor that depends on *h*.

3.8 Hoarseness

Although hoarseness is sometimes thought of as a symptom of some kind of vocal disorder, this effect has been widely used by singers in order to resemble the voice of famous performers (Louis Armstrong or Tom Waits, for example). We can accomplish a similar effect through a very basic transformation by just applying a gain to the residual component of our analysis.

3.9 Morphing

Morphing is a transformation with which, out of two or more elements, we can generate new ones with hybrid properties.

Most of the interpolation techniques are 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.

A first approach to such transformation could consist in interpolating two sinusoidal spectral shapes to obtain the hybrid $X'(f)$:

$$X'(f) = \alpha \cdot X_1(f) + (1-\alpha) \cdot X_2(f) \quad (19)$$

However, more musical meaningful results can be achieved in a more flexible way by moving the interpolation to a higher-level features plane, in which we could, for example interpolate between pitches:

$$f_0' = \alpha \cdot f_{01} + (1-\alpha) \cdot f_{02} \quad (20)$$

In both examples (equations 19 and 20) α would be the interpolation factor and would take values between 0 and 1.

4 Content dependent processing

The hierarchical data structure that includes a complete description of a given sound offers many possibilities for sound transformations. Modifying several attributes at the same time and at different abstraction levels achieve, as it has already been pointed out in the previous section, most musically or end-user meaningful transformations.

Higher-level transformations can refer to aspects like sound character, articulation or expressive phrasing. These ideas lead to the development of front ends such as graphical interfaces or knowledge-based systems [Arcos 98] that are able to deal with the complexity of this sound representation.

In this section we introduce two applications that have been developed with these ideas in mind: an automatic singing voice conversion application and a time scaling module.

4.1 Real-Time Singing Voice Conversion

Our automatic voice conversion application implements a very particular case of audio morphing, pursuing the possibility of morphing, in real-time, two singing voice signals in such a way we can control the resulting synthetic voice by mixing some characteristics of the two sources. Whenever this control is performed by means of modifying a reference voice signal matching its individuality parameters to another, we can refer to it as voice conversion.

In such a context, a karaoke-type application, in which the user can sing like his/her favorite singers, was developed [Cano et al 00]. The result is basically 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.

Fig 6 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 (see 2.2) and a Hidden Markov Model based Speech Recognizer (SR). The SMS implementation is responsible of providing a suitable parameterization of the singing voice in order to perform the morph in a flexible and musical-meaningful way. On the other hand, the SR is responsible of matching the singing voice of the user with the target.

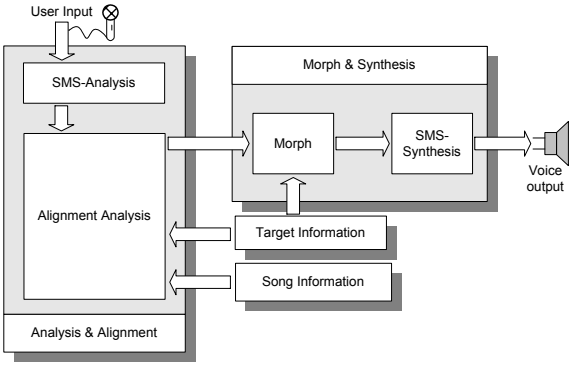

Fig.6. System block diagram

4.2 Time scaling

Time-scaling an audio signal means changing the length of the sound without affecting other perceptual features, such as pitch or timbre. Many different techniques, both in time and frequency domain, have been proposed to implement this effect. Some frequency domain techniques yield high-quality results and can work with large scaling factors. However, they are bound to present some artifacts, like phasiness, loss of attack sharpness and loss of stereo image. In this section we will present a frequency domain technique for near loss-less time-scale modification of a general musical stereo mix [Bonada 00].

The general block diagram of the system is represented in Fig 7. First, the input sound is windowed and goes through the FFT giving as a result the analysis frame (AFn), that is, the spectrum bins and the amplitude and phase envelopes. Then the time-scale module generates the synthesis frame (SFm) that is fed to the inverse FFT (IFFT). Finally, the Windowing&Overlap-Add block divides the sound segment by the analysis window and multiplies it by the overlap-add window, to reconstruct the output sound.

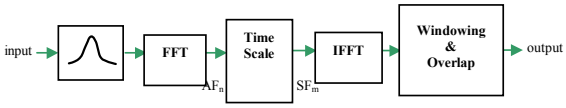

Fig. 7. General diagram

It is important to remark that the frame rate used in both the analysis and synthesis modules is the same, as opposed to the most broadly used time-scale techniques in which a change of frame rate in synthesis is used in order to achieve the effect. The window size and type must also be the same in both processes.

In some cases, an analysis frame is used twice (or more) while on other cases some frames are never used. This technique will not add any artifacts, provided the frame size we use is small enough and the sound does not present abrupt changes in that particular region. In the case of a percussive attack, though, a frame repetition or omission can be noticed regardless the analysis frame size. Therefore, some knowledge of the sound segment features is needed to decide where this technique can or cannot be applied.

In Fig 8, a more detailed block diagram of the time-scale module is depicted. The analysis frames (AFn), containing the spectrum amplitude and phase envelopes, are fed to the time-scaling module. This module performs a peak detection and a peak continuation algorithm on the current and previous (Z-1) amplitude envelopes. Then, only the peaks that belong to a sinusoidal track are used as inputs to the spectrum phase generation module. Note that the time-scale module only changes the phase, leaving the spectral amplitude envelope as it is.

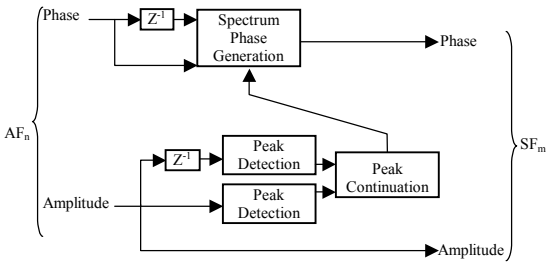

Fig. 8. The time-scale module

The phase of each peak is computed supposing that the frequency varies linearly between two

consecutives frames and that there is some phase deviation. The usage of the same frame rate in analysis and synthesis allows us to suppose that the phase variation between two consecutives frames is also the same.

5 Conclusions

Throughout this paper, we have shown how the use of higher-level spectral models can lead to new and interesting sound effects and transformations. We have also seen that it is not easy nor immediate to get a good spectral representation of a sound, so the usage of this kind of approach needs to be carefully considered bearing in mind the application and the type of sounds we want to process.

For example, most of the techniques here presented work well only on monophonic sounds and some rely on the pseudo-harmonicity of the input signal.

Nevertheless, the use of spectral models for musical processing has not been around too long and it has already proven useful for many applications, as the ones presented in this paper. Under many circumstances, higher-level spectral models, such as the Sinusoidal plus Residual, offer much more flexibility and processing capabilities than more immediate representations of the sound signal.

In general, higher-level sound representations will offer more flexibility at the cost of a more complex and time-consuming analysis process. It is important to remember that the model of the sound we choose will surely have great effect on the kind of transformations we will be able to achieve and on the complexity and efficiency of our implementation.

6 References

Arcos, J. L.; R. López de Mántaras; X. Serra. 1998. "Saxex: a Case-Based Reasoning System for Generating Expressive Musical Performances", Journal of New Music Research, Vol. 27, N. 3, Sept. 1998.

Bonada, J. 2000. “Automatic technique in frequency domain for near-lossless time-scale modification of audio.” 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.

Ding Y.; X. Qian. 1997. "Sinusoidal and Residual Decomposition and Residual Modeling of Musical Tones Using the QUASAR Signal Model". Proceedings of the 1997 International Computer Music Conference. San Francisco: Computer Music Association.

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.

McAulay, R. J. and T. F. Quatieri. 1986. “Speech Analysis/Synthesis based on a Sinusoidal Representation.” IEEE Transactions on Acoustics, Speech and Signal Processing 34(4):744--754.

Serra, X. 1987 A System for Sound Analysis/Transformation/Synthesis based on a Deterministic plus Stochastic Decomposition. Ph.D. Dissertation, Stanford University.

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. 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.

Smith, J.O. and X. Serra. 1987. “PARSHL: An Analysis/Synthesis Program for Non-Harmonic Sounds based on a Sinusoidal Representation.” Proceedings of the 1987 International Computer Music Conference. San Francisco: Computer Music Association.

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