Sound hybridization based on a deterministic plus stochastic decomposition model

Xavier Serra
Fundació Phonos - Institut Universitari de l'Audiovisual
Rambla 31
08002 Barcelona, Spain
xserra@upf.es

Abstract

Current sound hybridization techniques are based on the exciter-resonator model, where the spectral characteristics of a sound (resonator) are applied to another one (exciter). The traditional implementations use Linear Predictive Coding, the Short-time Fourier Transform (STFT) or the Phase Vocoder and the most common application is to hibridize speech (used as a resonator) with a spectrally rich sound (used as a exciter), resulting into the characteristic "talking sound" effect. The use of a more powerful sound representation model like Spectral Modeling Synthesis (SMS)[Serra and Smith, 1989] takes the concept of sound hybridization beyond the traditional exciter-resonator model. This paper starts by describing a hybridization system based on the STFT and then it presents the system based on SMS.

1. Hybridizing with the STFT

The Short-time Fourier Transform is a powerful technique with which the time-varying spectral shape of a sound (resonator) can be applied to another sound (exciter). We have implemented a version of this technique which permits a time-varying control of the hybridization process. We will refer to *sound1* as the exciter sound and to *sound2* as the resonator sound, but before describing how the process is done, let us define the parameters that the user can control.

- *frame-rate.* Frame rate used to calculate the STFT of *sound1.*
- *window-overlapping.* Amount of overlapping of the analysis window used for *sound1.* The actual length of the window in seconds is $1/(\textit{frame-rate} * \textit{window-overlapping})$.
- *n-spectral-coefficients.* Number of line segments used to approximate the spectrum of *sound2.* This parameter may vary in time using a user-defined envelope.
- *magnitude-balance.* Magnitude balance between the two sounds. With a value of 0 the hybridized sound follows the magnitude of *sound1.* With a value of 1 it follows the magnitude of *sound2.* Any number in between can also be given. This parameter may also vary in time by specifying an envelope.
- *time-stretch.* Time stretch value for *sound2.* Given that the two input sounds can have different durations, an implicit stretching or compression is applied to *sound2* in order to have the same number of frames in the two sounds. Therefore the analysis frame-rate used for *sound2* is different than the one used for *sound1.* However when the *time-stretch* parameter is set, it modifies the implicit stretching and the user can control the relative time-varying stretch of *sound2* by using and envelope, given the constrain that the duration of the output sound is the duration of *sound1.*
- *compression-envelope.* Before multiplying the magnitude spectra of the two sounds, they are normalized and compressed by applying this *compression-envelope* as a way to control the relative contribution of each one in the final output. This envelope can have any value between 0 and 2. A value of 0 compresses the corresponding spectrum magnitude point of *sound2* and does not modify the spectral value of *sound1*, therefore the only contributing spectral value is the one of *sound2.* A value of 2 produces the opposite, and a value of 1 leaves the two spectral magnitude values as they are. This envelope may also vary in time by giving another

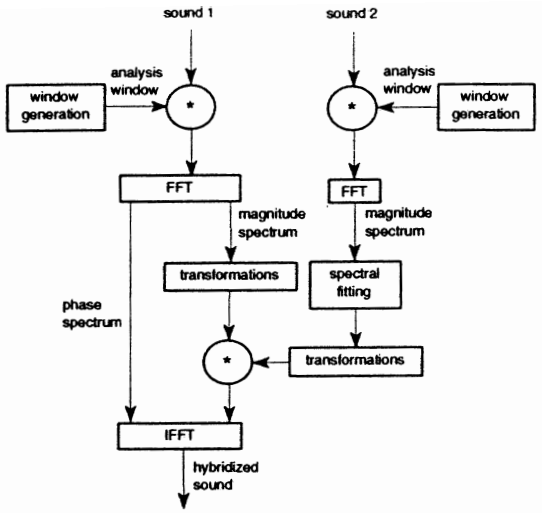

Figure 1: Diagram of the STFT-based hybridizing system.

compression-envelope for it and a function to interpolate between the two.

Figure 1 shows a block diagram of the system. Since it is frame-based, all the calculations are done one frame at a time and we refer to the current frame as the data that is being processed at the current time. The first step is to select the current frame by moving the data pointers to the current location of the two input sounds and multiplying each one with an analysis window. The length of the window used for *sound1* is defined by *frame-rate* * *window-overlapping* and the length of the window used for *sound2* can be set independently (e.g., as a percentage of the window-length used for *sound1*), and does not have the restrictions imposed by the STFT (the windows should overlap to a constant). Since the only relation between the two sounds is that we have to compute the same number of frames (*frame-rate* * sound1-duration), the center location of the current frame in *sound2* is determined by the relative duration of the two sounds and the current value of *time-stretch*. From these values the program calculates the appropiate time advance (or hop-size) from the center of the previous frame to the current one.

The next step is to compute the spectra of the two data frames with the Fast Fourier Transform (FFT). In the case of *sound1* we need both magnitude and phase values since we are going to do an inverse-FFT (IFFT) from them. However, of *sound2* we only need the magnitudes because only its spectral shape is used. The spectral shape of *sound2* is obtained by performing a line-segment approximation of the magnitude spectrum with a number of breakpoints given by the current value of *n-spectral-coefficients*.

To multiply the two spectra they have to be the same length, the one of the spectrum of *sound1*. Hence, the envelope of size *n-spectral-coefficients* is interpolated to this size. The two arrays are then normalized by dividing them by their respective average magnitudes. From these averages the hybridized average magnitude is calculated according to the *magnitude-balance* parameter and added to the hybridized spectrum. Before the spectra are multiplied, they are compressed according to the current *compression-envelope*, process that has the effect of controlling their relative contribution.

The final step is to use the IFFT to get the output frame from the hybridized magnitude spectrum and the phase spectrum of *sound1*. The output frames are combined with the overlap-add method.

This technique is most successful when *sound1* is spectrally rich and dynamically stable, so its filtering has a perceptual effect, and when *sound2* has a fast changing spectral shape, and/or rapidly changing dynamics, so it can impose clear attributes on *sound1*. For example, the sound of the sea is ideal as *sound1* and speech as *sound2*. Nevertheless, we have experimented with many other pairs of sounds with good success, especially because in our implementation we can control in time the degree of hybridization both in terms of timbre and dynamics.

2. Sound analysis with SMS

Spectral Modeling Synthesis (SMS) is a spectrum modeling technique that models the time-varying spectra of a sound as (1) a collection of sinusoids controlled through time by piecewise linear amplitude and frequency envelopes (the "deterministic" part), and (2) a time-varying filtered noise component (the "stochastic" part). This technique has proved to give general, high quality transformations for a wide variety of musical signals and a recent development on the SMS system has added the possibility to interpolate, using time-varying functions, the data of two or more analyzed sounds before the synthesis is done.

Figure 2 shows the block diagram for the analysis part of the system. The first two steps are the same ones used in the STFT-based system. In the SMS system, once the spectrum is obtained by the FFT, the prominent peaks in the magnitude spectrum are detected and organized into frequency trajectories by means of a peak continuation algorithm. The relevance of this algorithm is that it detects the magnitude, frequency and phase of the stable sinusoids present in the original sound

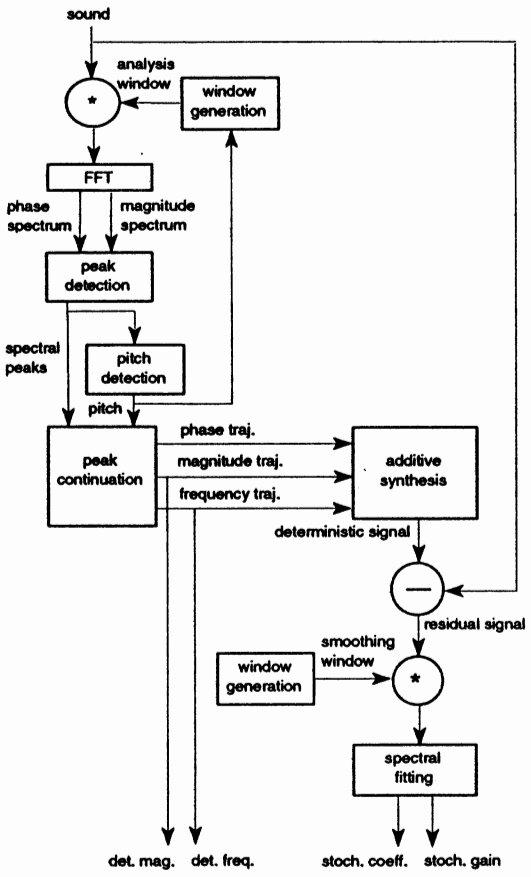

Figure 2: Diagram of the SMS analysis process.

(the deterministic component). When the sound is pseudo-harmonic, a pitch detection step can improve the analysis by using the pitch information in the peak continuation algorithm and in defining the analysis window for the next frame (pitch-synchronous analysis).

The stochastic component of the current frame is calculated by first generating the deterministic signal with additive synthesis, and then subtracting it from the original waveform in the time domain. This is possible because the phases of the original sound are matched and, as a result, the shape of the waveform is preserved. The stochastic representation is then obtained by performing a spectral fitting of the residual signal. This fitting can either be done by a line-segment approximation on the magnitude spectrum or by an LPC-type approximation.

From the analysis data, the synthesis of the deterministic signal (i.e., the sinusoidal component) results from the magnitude and frequency trajectories, or their transformation, by generating a sine wave for each trajectory (i.e., additive synthesis). The transformations are done directly on the frequency and magnitude trajectories or on more meaningful musical parameters that can be extracted from and added back to these trajectories. An additive synthesis implementation based on the IFFT has been proposed [Rodet and Depalle, 1992] which is more efficient than the traditional oscillator bank implementation and makes possible the integration of the deterministic and stochastic synthesis processes.

The synthesized stochastic signal, when its representation is based on line-segments, is the result of creating a complex spectrum (i.e., magnitude and phase spectra) for every spectral envelope of the residual, or its modification. Performing an IFFT using the overlap-add method to form the final output. The magnitude spectrum is the envelope itself, and the phase spectrum is generated by a random number generator. This process corresponds to the filtering of white noise by a filter with a frequency response equal to the spectral envelope (subtractive synthesis).

3. Hybridizing with SMS

Once a group of sounds has been "well analyzed" with SMS, we can transform and synthesize these sounds in many different ways. One of the possible transformations is the hybridization between different sounds by interpolating their representations. The possible interpolations are infinite and go from interpolating all the parameters progressively through the length of a sound, therefore obtaining the effect of going from one sound to another, to only interpolating a part of the two sounds, for example, the frequency or magnitude of a group of partials. Another possible interpolation is to use the complete deterministic magnitudes of one sound with the complete deterministic frequencies of the other, obtaining the traditional effect produced by the STFT-based system. In our implementation of this technique, shown in Figure 3, there are several parameters accessible to the user to control this interpolation process.

- *deterministic-frequency-factor.* Contributing factor of the frequency of *sound2* into the resulting sound. When the value is positive (between 0 and 1) the hybridization is done between all the partials of both sounds, and when it is negative (between 0 and -1), *sound1* only contributes with its fundamental. So, a value of -1 results into a sound that follows the fundamental of *sound1* and the frequency relations of *sound2*. This parameter may vary in time with a user-defined envelope.
- *deterministic-amplitude-factor.* Contributing factor of the deterministic magnitude of

sound2 into the resulting sound. When the value is positive (between 0 and 1) the hybridization is done between all the partials of both sounds, and when the value is negative (between 0 and −1) it is done using only of *sound1* its overall magnitude. Therefore, a value of −1 results into a sound with the overall magnitude evolution of *sound1* and the magnitude relations of *sound2*. This parameter can also vary in time by specifying an envelope.

- *stochastic-amplitude-factor.* Time-varying contributing factor of the stochastic magnitude of *sound2* into the output sound.
- *stochastic-coefficients-factor.* Time-varying contributing factor of the stochastic coefficients of *sound2* into the resulting sound.
- *time-stretch-factor.* Time-varying stretch applied to *sound2* independently of *sound1*.

Apart from these parameters there are others which can transform the data once it has been hybridized, therefore, giving the user more control over the hybridized sound. These are: *deterministic-amplitude-envelope*, *deterministic--amplitude-partials-factor*, *stochastic--amplitude-envelope*, *stochastic-coefficients-factor*, *deterministic-frequency-envelope*, *deterministic--frequency-partials-factor*, *time-stretch-envelope*, *frequency-stretch-envelope*.

The use of SMS as a hybridization technique is most succesful when the sounds involved are monophonic and pseudo-harmonic, so they have well ordered partials and the correlations between the data sets can be properly established. In this situation the user control is very powerful and the concept of moving around in the timbre space formed by a set of sounds, becomes a reality. Compared with the STFT-based system, we lose the possibility of hybridizing very complex sounds, but we gain flexibility in the transformations. Thus, the two systems complement each other offering different musical possibilities. The current implementations of both systems run under NeXTStep and use the note-list input language that is part of the MusicKit [Jaffe, 1989], giving a very high degree of control to them.

SMS can be thought as an analysis/synthesis environment where the user controls perceptually meaningful parameters. We could even go further in that direction by controlling sound attributes such as vibrato, overall spectral shape, attack time, etc.. This can be done with SMS by extracting such attributes from the analysis data before the transformations are applied, then making the transformations on these new parameters, and finally adding them back to the SMS representation before the synthesis.

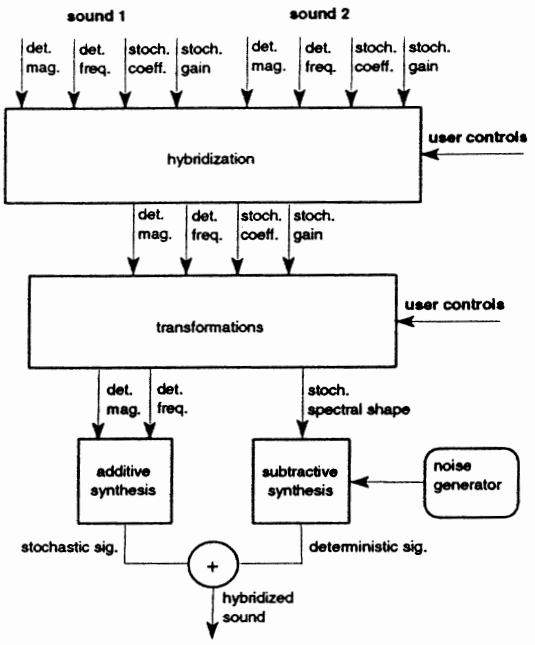

Figure 3: Diagram of the SMS-based system for hybridizing sounds.

4. Conclusion

The STFT and SMS are two analysis/synthesis techniques with a lot of musical potential. With the STFT the spectral shape of a sound can be applied to another sound and with SMS we can interpolate a large number of perceptual characteristics of two sounds. With SMS the hybridization, or cross-synthesis, concept used in Computer Music is taken beyond the traditional meaning.

All the software that implements these techniques is publicly available on the ftp server: ccrma-ftp.stanford.edu, under the name sms.tar.Z.

5. References

[Jaffe, 1989] D. Jaffe. "Overview of the NeXT Music Kit." *Proceedings of the 1990 ICMC.* San Francisco: Computer Music Association.

[Rodet and Depalle, 1992] X. Rodet and P. Depalle. "Spectral Envelopes and Inverse FFT Synthesis." 93rd Convention of the AES.

[Serra, 1989] X. Serra. *A System for Sound Analysis/Transformation/Synthesis based on a Deterministic plus Stochastic Decomposition.* Ph.D. Dissertation, Stanford University.

[Serra and Smith, 1989] X. Serra and J. Smith. "Spectral Modeling Synthesis." *Proceedings of the 1989 ICMC.* San Francisco: Computer Music Association.