SYNTHESIZING TRUMPET PERFORMANCES

Istvan Derenyi and Roger B. Dannenberg

School of Computer Science, Carnegie Mellon University
Pittsburgh, PA 15213, USA
{derenyi, rbd}@cs.cmu.edu

Abstract: This paper presents our latest results in synthesizing high quality trumpet performances. Our approach identifies the continuous control over the sound as a fundamental element in the synthesis process. We developed our synthesis model as a cooperative system of two sub-models, continuous control parameters providing the interface between them. The two sub-parts are the instrument model, which takes sources of continuous control signals as input and produces an audio output, and the performance model, which takes some symbolic representation of music as an input, and produces control signals.

1. Introduction

Our goal is to develop a synthesis model capable of rendering highly realistic trumpet performances. Our impression is that current synthesis techniques fail to achieve that goal for a couple of reasons. The trumpet (among other instruments) can be characterized by the fact that the player exercises continuous control over the course of notes and phrases. This results in a continuous evolution of spectrum as a function of that control. It follows naturally that template-based synthesis is not able to synthesize highly realistic performances of this type of instrument. (By template based synthesis we mean storing and combining individual recorded notes. Most of the current commercially available synthesis systems are sample based and belong to this category.) A successful synthesis technique has to be able to render sound based on continuous control.

Also, the specific realization of such a control depends strongly on the musical context in which the actual note is embedded. As an example, we can show that the amplitude envelope shape of a single note is dependent upon the pitch contour of the containing phrase. (Dannenberg, Pellerin, and Derenyi 1998) From this, it follows that synthesis of single notes (which is the practice followed by most synthesis research) is not adequate for our purposes either. We believe that a more holistic integration of control and synthesis is necessary for realistic synthesis and to create appropriate control functions for the synthesis.

As we pointed out, the continuous control signals have to play a key role in the synthesis process. The idea of control signals is quite common and its use can be identified in most of the synthesis techniques. However, the problem of how to produce appropriate control signals remains. There are two “directions” in which we would like to derive the control signals. During testing, we would like to measure “reference” control signals from real performances and compare them to synthetic control signals. FM synthesis is a good example how problematic this issue can be. During synthesis, as an ultimate goal, we would like to derive our control signals from symbolic data. If those control signals are closely tied to musical concepts such as amplitude or pitch, then rules to produce those control signals can be derived by hand or by machine learning techniques. However, if the control signals represent peculiarities of the synthesis technique (such as with different physical modeling synthesis techniques, then control signals are more difficult to derive. We propose a new technique, which addresses these requirements.

The next section gives an overview of this new technique. Section 3 describes related work. We conducted experiments to test some of the assumptions of our technique, and these are described in Section 4. Sections 5 and 6 describe the instrument model and the performance model. Future work is described in Section 7, which is followed by a summary and conclusions.

2. The Combined SIS Model

Our synthesis model takes a symbolic score as input and produces a digital audio performance as output. As we described earlier, continuous control parameters play a key role as an intermediate representation in the synthesis process. The overall model is built upon the performance model, which generates control signals from the symbolic score, and the instrument model, which produces the audio output based on the control signals.

Derenyi, I. and R. B. Dannenberg. 1998. “Synthesizing Trumpet Performances.” In *Proceedings of the International Computer Music Conference*. San Francisco: International Computer Music Association.

2.1 The Performance Model

The performance model starts with a symbolic, machine-readable score and produces time-varying amplitude and frequency control functions for the Instrument Model, which is described below. Amplitude and frequency were chosen because they are easy to measure and musically salient. It turns out that these are sufficient to encode most of what is going on during a trumpet performance. However, we also need to know whether note transitions are tongued or slurred. Other control parameters certainly exist and could be added.

The performance model is currently constructed “by hand,” although machine learning techniques will be applied in the future. The first step is to examine control functions extracted from actual acoustic performances. Based on these, rules are developed to relate envelope shape to score parameters. The rules are tested and refined by comparing the shapes they generate to shapes measured from human players.

2.2. The Instrument Model

The instrument model, excited by the control functions, produces the final audio output. This digital sound should be perceptually very close to the modeled instrument to insure the success of the combined performance+instrument model. To assure this, the instrument model can be excited by control signals measured directly from real performances as opposed to control signals created by the performance model, and the audio output can be compared to that same real performance.

The instrument model used is an extended version of the early Spectrum Interpolation Synthesis (SIS), described by Serra, Rubine, and Dannenberg (1990). The basic underlying assumption of that technique is that the spectrum of a note is nearly harmonic at each time-point. The overall sound is only quasi-periodic though, as the timbre of the sound exhibits a continuously changing spectrum, pitch, and amplitude. Compared to audio frequencies however, the rate of this change is quite slow, and can generally be modeled using control signals with components below 20 Hz.

3. Related Work

The early SIS simply attempted to reproduce single notes by the following procedure: In the analysis step, a set of time points and corresponding spectra in the original tone was identified. In the synthesis step, the selected spectra were converted to a time-domain representation (basically wavetables with adjusted phases, representing one period of the sound). The original tone was reproduced by interpolating between those wave tables. The number of time-points and spectra was chosen to obtain a synthetic sound close to the original.

Using that technique, the authors successfully reproduced sounds of different wind instruments, using only 5 to 20 spectra per second (called the spectral sample rate). This showed that the basic technique of spectral interpolation is adequate to reproduces the sound of these instruments. These early experiments also revealed that, as one might expect, the basic assumption of harmonicity breaks down during the attack portion of some tones. Several instruments have attacks with noise and inharmonic spectra which cannot be reproduced by a pure spectral interpolation technique.

Fortunately, this inharmonicity is limited to only tens of milliseconds at the beginning of the sounds. Still, as inharmonicity has significant perceptual effects, an extension to the basic spectrum interpolation technique became necessary. The authors experimented with a couple of approaches, and applied the technique of splicing sampled attacks to solve this problem. We refined and applied the technique of splicing sampled attacks onto SI sounds. (See Section 2.1.1.)

In the previous work, absolute time plays the role of a single continuous control function. To meet our goals, time can be exchanged for a few more “meaningful” parameters. Our choices are the amplitude and pitch. The instrument model is based on the assumption that at every time-point except during the attack, the instantaneous harmonic spectrum of the trumpet is determined by the current RMS amplitude and fundamental frequency, and nothing else. (We will call these modulation sources later). These signals satisfy our requirements for control signals. They can be measured from real performances. Also, they are musically relevant, and have well-defined meanings to the performer, which helps in creating the performance model later.

The idea of spectral interpolation appears repeatedly in the literature in support of various schemes for spectral variation. One application uses a small set of interpolated wavetables for tone generation (Kleczkowski 1989). Our approach differs mainly in that we express spectrum as a function of control parameters rather than as a direct function of time. Closely related to our work, Beauchamp and Horner (1995) showed that for the trumpet, the spectral envelope depends only upon amplitude. They created a similar synthesis model in which spectral variation is controlled by amplitude envelopes.

4. Experiments

It is well known that the trumpet and other wind instruments sound brighter when played louder. As the instrument is played louder, the amplitudes of higher harmonics grow faster than the amplitudes of lower ones. It seems to be quite obvious that the timbre of the sound changes with different frequency and amplitude levels; however, we wanted to test that other factors than those do not contribute to significant timbre variations. These factors could be for example the speed (slow or fast) and the direction (crescendo or decrescendo) of the change of the amplitude level. To test this hypothesis, we recorded simple trumpet notes at different pitches, with slowly as well as rapidly increasing and decreasing amplitude levels. We plotted the amplitudes of selected harmonics against the overall RMS amplitude of several notes with the same pitch.

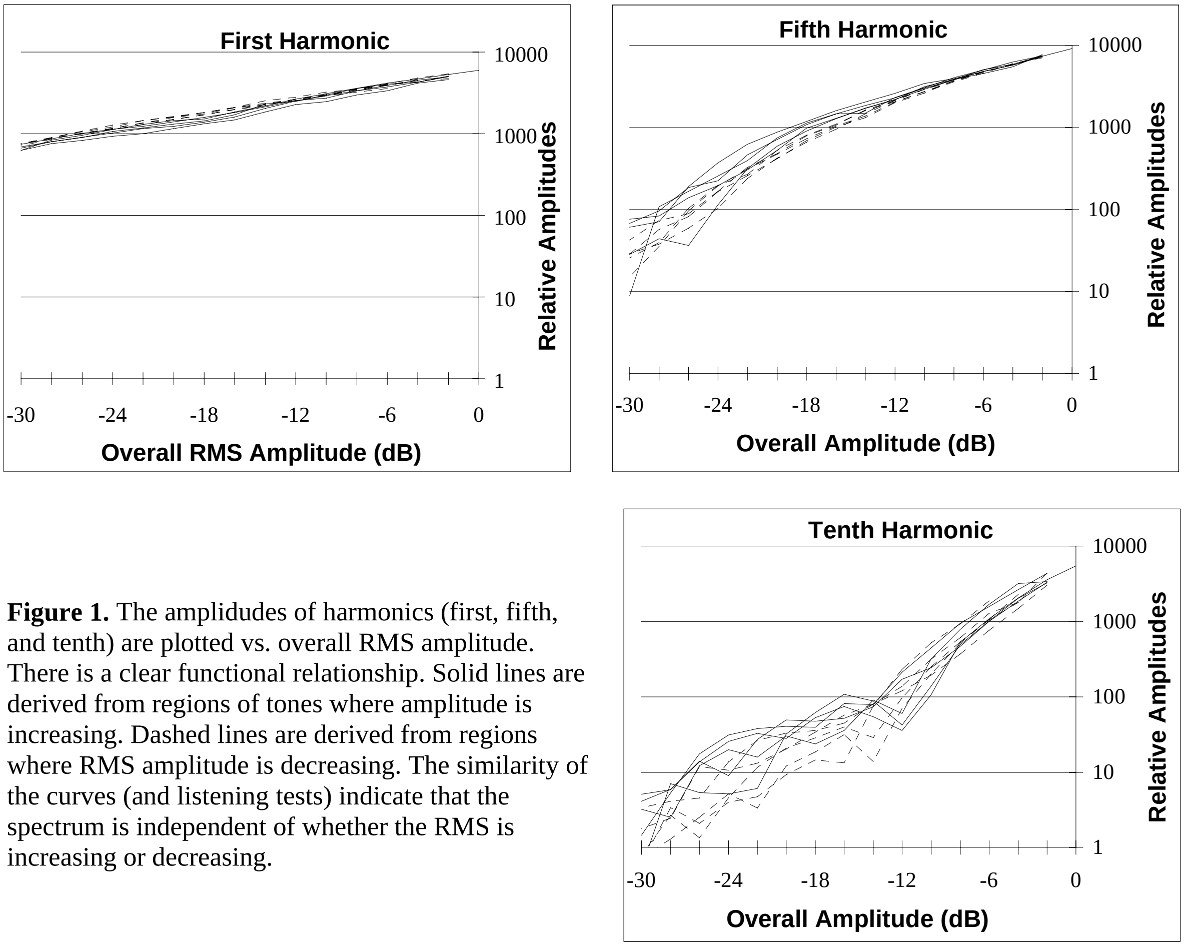

Figure 1. The amplidudes of harmonics (first, fifth, and tenth) are plotted vs. overall RMS amplitude. There is a clear functional relationship. Solid lines are derived from regions of tones where amplitude is increasing. Dashed lines are derived from regions where RMS amplitude is decreasing. The similarity of the curves (and listening tests) indicate that the spectrum is independent of whether the RMS is increasing or decreasing.

In Figure 1 we show examples of the resulting graphs. These show the first, fifth and tenth harmonics from ten measurements; five from crescendos and five from decrescendos. These graphs show that, at high amplitudes, there is indeed a direct relationship between RMS amplitude and the amplitude of a given harmonic. The relationship appears to be independent of whether amplitude is increasing or decreasing; otherwise, there would be a separation of the 10 curves into two clusters.

Listening tests were also created using two sets of spectra, one set derived from a crescendo and one set from a decrescendo. The differences, if any, were very small, indicating that the direction of change has a negligible effect on the spectrum. Similarly, we could find no support for the possibility that the rate of change (up to a point) makes a significant difference. Note, however, that the spectral content of attacks is different from crescendos which rise less rapidly.

Although the rate and direction of amplitude change seem to be of no consequence, there is quite a lot of variation among the curves in Figure 1, especially in the higher harmonics and lower amplitudes. There could be other factors at work systematically, or these could be random variations. Listening tests indicate that amplitude and frequency alone are sufficient to produce realistic trumpet tones.

5. Synthesis

The synthesis technique itself is quite simple, and can be summarized as follows:

- Take a set of two modulation sources, which describe the continuous value of amplitude and pitch (as a function of time);
- At well-defined time-points, take the values of those modulation sources, and use these values to index a two-dimensional database of spectra;
- Create time-domain representations of the spectra, which will be single periods of the sound at those time points;
- Create the desired sound by outputting these periods at the appropriate time-points, smoothly interpolating from one into the next between the time-points.

The spectrum itself is stored as an array of the relative amplitudes of the harmonics. We acquire the spectral content of the sounds of the trumpet for discrete values of the amplitude and fundamental frequency, and store them in a two dimensional Spectral Database. When this database is accessed by the instantaneous values of the modulation sources, we interpolate among the four neighboring spectra to get the required output spectrum.

We do not store the phase information for the harmonics in the spectra. In step 3 above, we create a series of wavetables with matching phases to avoid any phase cancellation when two wavetables are interpolated. The phases are determined by the spliced attacks, as described later. The synthesized phase may be different from the original one. This is possible because phase information does not have significant audible effects on the synthesized sound. We carried out listening tests to confirm this assumption.

Note that a frequency-domain representation of spectra creates an exciting opportunity to produce pitch-independent transfer functions using inexpensive multiplies. This could help simulate resonances, direction-dependent radiation losses, and other effects. We have not explored this possibility yet.

The time-domain signal resulting from the spectrum interpolation is frequency modulated by the pitch modulation source, and amplitude modulated by the amplitude modulation source. After that step, the synthesized sound has the expected fluctuation in timbre, amplitude and pitch. (See Figure 2.)

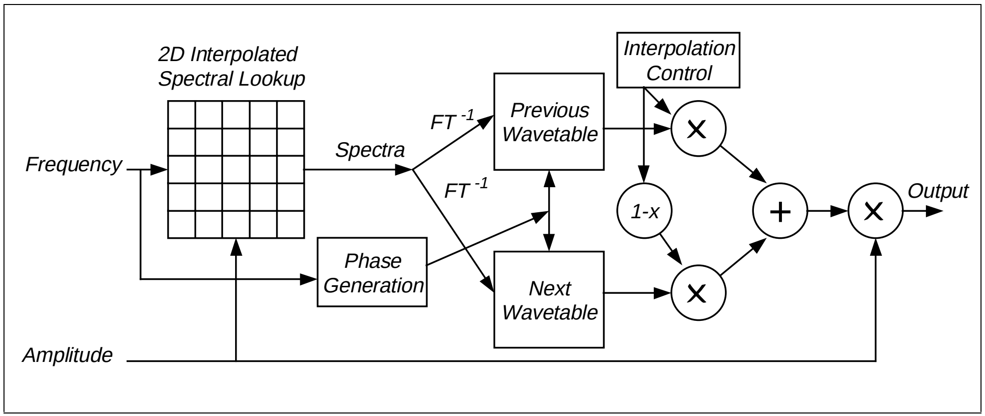

Figure 2. Spectral Interpolation with Frequency and Amplitude control. Attack splicing is not shown.

One could argue that the final amplitude modulation is not necessary. As the technique assumes a functional relationship between the amplitude and the spectrum, the amplitude variation could simply be represented in the stored spectrum itself. Indeed, originally we tried to follow that approach, which is computationally somewhat cheaper. However, that approach couples the rate of amplitude variation to the rate of spectral variation. Previous experiments determined that a spectral rate of 20 Hz is sufficient for our purposes. However, a sample rate of 20 Hz proved to be not fast enough to track rapid amplitude changes in the sound, especially between slurred notes.

To solve this problem, we store spectra with normalized amplitudes, and the amplitude fluctuation is realized by a multiplication with the modulation source. This signal is realized as a piece-wise linear curve, with 100 breakpoints per second, which proves to be fast enough for our purposes.

But to use the synthesis technique described above, we first have to build our Spectral Database. For this analysis step, we used the SNDAN utility package (Beauchamp 1993) to extract the instantaneous spectrum information, RMS amplitude, and pitch control signals from single notes. We define a number of amplitude and frequency levels at which we want to store the spectral data. At each pitch, a trumpet player plays single notes with decreasing amplitude, covering the playable dynamic range of the trumpet (approximately to 30dB). We found that it was easier for the player to produce a steady decrescendo than a crescendo. Software automatically extracts spectra corresponding to different amplitude levels from these recorded samples.

5.1. Attack Transients

Following this procedure, we can synthesize the trumpet with high quality except for one particular, but very important detail: the attack portion of the sound. It is well known that attacks carry significant perceptual cues for the listener. The heavily inharmonic attack cannot be reproduced by the spectral interpolation method, so an extension of the basic method became necessary.

We simply splice sampled attacks onto synthesized sounds. The process of splicing is as follows:

1. Choose a recorded attack from a database (more about this later). Several pieces of necessary information have to be stored together with the samples:
   - The phase distribution of the harmonics at the end of the recorded attack;
   - The amplitude distribution of the harmonics at the end of the recorded attack;
   - The overall RMS amplitude at the end of the recorded attack.
2. Use this phase distribution for computing all the wavetables in the subsequent spectrally interpolated sound. This ensures that the phases will match at the splice point and that no phase cancellation will occur in the spectrally interpolated sound.
3. The overall RMS amplitude modulation source will specify a certain amplitude distribution to generate at the splice point. Instead of using that, use the amplitude distribution measured at the end of the attack to generate the first wavetable in the spectral interpolation. All subsequent wavetables are derived using the normal spectral interpolation technique. This ensures that if there is a slight difference between amplitude distributions at the splice point, it will not cause audible clicks or other artifacts in the sound. During the first interpolation period, which is 1/20th second in our case, the amplitude distribution is smoothly interpolated from the one at the end of the spliced attack to the exact one which is described by the modulation sources.
4. Finally, using the original amplitude at the end of the recorded attack, match the RMS amplitudes of the attack and the amplitude modulation source at the splice point by linearly scaling the entire attack.

Note in step 4 that we do not use different recorded attacks for different amplitude levels; instead we scale one sample. Originally, we thought that we would have to store several sampled attacks at different pitch and amplitude levels, analogous to the Spectral Database. We imagined there might be some other dimensions as well, considering the complex nature of the attacks. However, we obtained satisfactory results using only one sampled attack for each pitch, recorded from a relatively loud trumpet attack. The attack is scaled downward to the required final RMS value.

Another issue is how to choose the length of the attacks. Attacks must be short enough so they convey only a negligible amount of amplitude shape information. We want shape to be determined by amplitude envelopes, not by the attack. We would like to avoid any cumbersome technique to “reshape” the attacks to follow the prescribed amplitude modulation source. This is all possible if the lengths of the attacks are short enough. Short attacks also minimize memory requirements. On the other hand, the sampled attack should be long enough to cover the whole inharmonic part of the sound, and at its end it should settle into a stable harmonic structure that can be analyzed accurately. This is necessary to produce a smooth, inaudible splice.

The point where the inharmonic portion of the sound ends can be measured automatically by observing the relationships among the partials. For the time being, we have not implemented this technique, and choose the length of the attacks manually, using listening tests. We found that 30 ms attacks work well with the trumpet. Note however, that during synthesis, the splicing is automatic and is incorporated into the basic spectral interpolation model.

Attacks are only used when there is a note with a tongued onset, which produces a stoppage of the airflow and a definite silence (see the description of the performance model). Due to the silence, we never need to splice to the beginning of an attack. In the case of slurs and legato transitions, attacks are simply omitted. Thus, sound is not synthesized note-by-note but rather in phrases of notes, starting with an attack and ending just before the next attack. Within the phrase, the harmonic phases are all dictated by the initial attack. Because the phases depend upon the attack used at the beginning of the phrase, we cannot precompute tables. Instead, we construct them as necessary.

Using this instrument model, we rendered performances of excerpts from the Haydn Trumpet Concerto, using amplitude and frequency control signals (modulation sources), measured from real performances of the same piece. This experiment shows that, given the proper modulation sources, we can synthesize realistic trumpet performances. The next section is concerned with the construction of these modulation sources.

6. The Performance Model

The goal of the performance model is to automatically create the control signals of amplitude and pitch for the instrument model, starting with symbolic music notation. We assume that the score is available in some machine-readable form. The general idea is that the musical context largely determines the shapes of the amplitude and frequency curves during a live performance. If we want to render a realistic synthesized performance, our model has to be able to create the appropriate controls.

To create the performance model, we performed a careful study of trumpet envelopes. A trumpet player played characteristic phrases, which were designed to elicit different typical envelopes under controlled conditions. We measured the resulting envelopes and generalized from them.

The performance model constructs amplitude envelopes depending upon the indicated articulation (e.g. attacked or slurred), direction and magnitude of pitch intervals, separation between notes if any, duration of notes, implied phrases, and pitch. A 10-parameter envelope model uses a combination of parameteric functions and actual envelope data to produce an appropriate envelope for synthesis.

Realism requires some frequency fluctuation, but we have not discovered any clear dependency between frequency deviation and articulation style, pitch, or other performance parameters. Elaborate models for frequency deviation based on performance parameters seem unnecessary, although vibrato would certainly require careful modeling. In this work, frequency modulation is based simply on stored envelopes derived from a performance.

More details on the features of trumpet envelopes and on the performance model are presented in a companion article (Dannenberg, Pellerin, and Derenyi, 1998). Our current model is simple and limited, and is based only on studies of the trumpet. However, we believe that similar performance models can be derived for other instruments as well, following the same process we developed. This work seems particularly applicable to wind instruments, and previous experience with various winds (trombone, alto saxophone, bassoon, clarinet) indicates that the SI synthesis will work well with these families of sounds.

7. Future Work

This work is only the beginning of a potentially long line of research. We have focussed on the trumpet, so a logical direction is to work with other instruments. The original SIS model worked well with other wind instruments. We need to discover whether the trumpet envelope models can be adapted to other instruments or whether entirely new models are needed. It would be interesting to explore the spectral changes brought about when players bend pitches. Can this be modeled by spectral multiplication? Is another control dimension necessary? There might also be extensions to model noise as in the Spectral Modeling approach (Serra 1994). Spectral interpolation lends itself to many interesting variations. Interpolation can take place across instruments, and the two dimensional spectral control space can be subjected to geometric rotations, reversals, and other transformations which might be of interest to composers and instrument designers.

Capturing spectral information is more-or-less automated, but building the performance model is an ad-hoc process. Having constructed a simple model of trumpet performance, we believe that machine learning could do the job better and faster. One approach is to develop a parameterized model for envelopes and then use machine learning to search for relationships between the score and the model parameters. This seems like a tractable problem. Another area for exploration is in real-time interfaces for spectral interpolation synthesis. This could in some ways bypass the problem of the performance model, allowing a human performer to produce control functions directly and hear the result in real time. It should also be possible to use acoustic instruments and the voice as controllers: a real-time analysis of pitch and amplitude can drive the SIS instrument model. There are many problems with this as a general approach, but the potential for cross-synthesis is interesting.

8. Summary and Conclusions

We have presented our latest research results in developing a complex synthesis model for high quality synthesis of the trumpet and other wind instruments. Our main contribution is the formulation of the synthesis task as a combination of performance knowledge and instrument characterization. Our synthesis model is divided correspondingly into a performance model and an instrument model. These two sub-models are linked together with the carefully chosen control signals, which play a key role in our approach. The chosen control signals, the instantaneous amplitude and frequency signals, can be extracted from real performances, and so can be used to refine both the performance and the instrument model.

The key element in our instrument model is the realization that the time-varying spectrum of the instrument is determined primarily by the instantaneous values of our chosen control signals. So, the instrument can be modeled as a mapping from those control signals onto the corresponding spectrum. The mapping function can be created automatically from real performances. We extended the basic model to reproduce realistic inharmonic attacks as well.

We carried out studies of trumpet envelopes and created a performance model to produce appropriate continuous control signals from symbolic music notation. Sound examples of classical trumpet performances, generated by our model, can be found at http://www.cs.cmu.edu/~rbd/music.

Overall, we believe this is a very promising technique. Our work with control functions in the context of phrases has convinced us that dealing with notes in isolation is a gross simplification that researchers must move beyond. Also, we believe that there is a great advantage to working with control functions that can be extracted easily from acoustic performances. Now we have a synthesis technique that can produce rich, natural sounds with tremendous control. The future challenges are to bring this technique into common practice and to explore the many possibilities for expressive control.

References

Beauchamp, J. 1993. “Unix Workstation Software for Analysis, Graphics, Modification, and Synthesis of Musical Sounds.” Audio Engineering Society Preprint, No. 3479 (Berlin Convention, March).

Beauchamp, J. and A. Horner. 1995. “Wavetable Interpolation Synthesis Based on Time-Variant Spectral Analysis of Musical Sounds,” *Audio Engineering Society Preprint,* No. 3960 (Paris Convention, Feb.), pp. 1-17.

Dannenberg, R. B., H. Pellerin, and I. Derenyi. 1998. “A Study of Trumpet Envelopes.” In *Proceedings of the International Computer Music Conference.* San Francisco: International Computer Music Association.

Kleczkowski, P. 1989. “Group Additive Synthesis.” *Computer Music Journal* 13(1), pp. 12-20.

Serra, M.-H., D. Rubine, and R. B. Dannenberg. 1990. “Analysis and Synthesis of Tones by Spectral Interpolation.” Journal of the Audio Engineering Society, 38(3) (March), pp. 111–128.

Serra, X. 1994. “Sound hybridization based on a deterministic plus stochastic decomposition model.” In *Proceedings of the International Computer Music Conference.* San Francisco: International Computer Music Association, pp. 348–351.