

## **Combining Instrument and Performance Models for High-Quality Music Synthesis**

**Roger B. Dannenberg and Istvan Derenyi**  
School of Computer Science, Carnegie Mellon University

### **ABSTRACT**

Convincing synthesis of wind instruments requires more than the reproduction of individual tones. Since the player exerts continuous control over amplitude, frequency, and other parameters, it is not adequate to store simple templates for individual tones and string them together to make phrases. Transitions are important, and the details of a tone are affected by context. To address these problems, we present an approach to music synthesis that relies on a performance model to generate musical control signals and an instrument model to generate appropriate time-varying spectra. This approach is carefully designed to facilitate model construction from recorded examples of acoustic performances. We report on our experience developing a system to synthesize trumpet performances from a symbolic score input.

### **INTRODUCTION**

Our goal is the creation of high-quality synthesized musical performances of wind instruments. These instruments are especially interesting because they are driven continuously by a source of energy controlled by the player. Because energy is always being added to the instrument, the player exerts continuous control over the sound production. Proper instrument performance relies on

---

Correspondence: Roger B. Dannenberg, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA. Tel.: +1-3214-412-268-3827. E-mail: roger.dannenberg@cs.cmu.edu.

This work was inspired by the teachings of Arthur Benade, an extraordinary mentor. Marie-Helen Serra, Dean Rubine, Paul McAvinney, Chris Fraley, and Hank Pelerin all contributed to the research and development of Spectral Interpolation Synthesis. The SNDAN program provided by James Beachamp has been very useful.

Manuscript submitted: March 27, 1998.

this control, and any synthesis effort must address the wide range of sounds so enabled. We have studied the trumpet and achieved good results. We believe that our techniques will apply to wind instruments in general and perhaps even to strings, but this must be demonstrated before we can draw conclusions.

Our work is oriented toward a synthesis model in which the input is a symbolic score, such as common practice music notation, and the output is a digital audio performance. There are other interesting problems, but this one forces us to address the problem of control, which is so important to music.

Given the problem of rendering an audio performance from a symbolic score, continuous control plays a key role as an intermediate representation in the rendering process. We first generate control signals from the score and then synthesize sound from the control signals. (See Fig. 1.) This is a fairly simple idea, and in most ways it is consistent with nearly all synthesis techniques. However, we have selected our continuous control parameters and designed our synthesis methods especially to help us create realistic performances.

One common problem in synthesis is to produce control signals that result in the desired sounds. A classic example is the specification of parameters for FM synthesis in order to render a given spectrum or spectral evolution. If a synthesis algorithm can be inverted, then good control parameters can be derived or approximated from the analysis of acoustic performances. If the synthesis algorithm cannot be inverted, then the search for good control parameters can be difficult, requiring human perception and intelligence.

Another problem in synthesis is the production of control signals from the score. If control parameters are closely tied to musical concepts such as pitch and loudness, then rules can be derived by hand or through machine learning

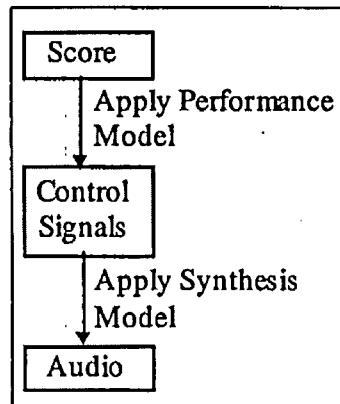


Fig. 1. The overall problem can be factored into two subproblems: the Performance Model and the Instrument Model.

to obtain control signals. On the other hand, if control parameters reflect peculiarities of the synthesis model (such as variations in lip tension in a physical model or the modulation index in FM synthesis), then control parameters are more difficult to obtain.

In our work the primary control signals are fundamental frequency and RMS amplitude. These signals are perceptually and musically relevant, which helps to derive these signals from a symbolic score. It is also possible to derive these signals from acoustic performances. This means that the synthesis model can be tested with "correct" information derived from a human performance. This in turn allows us to decompose the overall problem into the subproblems of control generation and sound synthesis.

### EXPERIMENTAL APPROACH

Before describing the techniques we have developed, we will explain the methodology that led to these techniques. The methodology is important because there is still much work to be done, and without a methodology, future progress would require new insight and breakthroughs. We do not expect all future work to be routine or automatic, but at least the methodology gives us hope that our techniques will apply to other instruments and musical styles.

Our first efforts were directed toward synthesis. We thought that we might be able to represent the time-varying spectrum as a function of amplitude and frequency envelopes. This led to a number of experiments to explore this hypothesis. In this work, acoustic performances are analyzed to derive amplitude and frequency envelopes. These are then fed back into the trial synthesis algorithm and the results are compared subjectively by listening. (See Fig. 2.)

The goal of this first stage is to develop an "instrument model" that can produce high-quality sound given the proper control signals. Proper control signals are insured because we use controls derived from the analysis of acoustic performances. Success at this stage is critical; if a good performance does not result from "real" control signals, there is little hope that artificially derived control signals will make the results any better.

The second stage is to derive control signals from a symbolic score. The task is to develop and refine a "performance model" that converts a score into control signals. There are many options available at this stage because of the way we have approached the problem. A good starting point is a set of examples of control signals extracted from performances of various articulations, intervals, pitches, and dynamic levels. These are used to (manually) build and refine rules for performance. To evaluate the resulting performance model, a human performer can be asked to play the symbolic score and the results can be ana-



mance synthesized using control signals from the artificial performance model. (See Fig. 4.) Listening to synthesized performances allows us to focus on the problems that are perceptible and ignore control signal discrepancies that do not matter. This may point to the need for more data collection, and the refinement process iterates.

The final test is to compare a completely synthetic performance to an acoustic, human performance. If these are indistinguishable, the synthesis is considered to be successful. In practice, however, there will always be unwanted imperfections and inconsistencies in the human performance that will not be duplicated in the synthetic one (even if the synthesis includes intentional human-like imperfections.) Therefore, there will always be noticeable differences between human and synthetic performances. Thus, we must evaluate the quality of synthesis in subjective terms, just as we would compare two human performers in an audition.

If the synthetic performance is not acceptable, the next step is to determine the source of the problem. If there is a subtle timbral difference, it may not be clear whether the performance model or the instrument model is at fault. To isolate the problem, we can analyze the acoustic performance and feed the control signals into the instrument model (as in Fig. 2). If this eliminates the problem, then the problem is with the performance model. We can compare

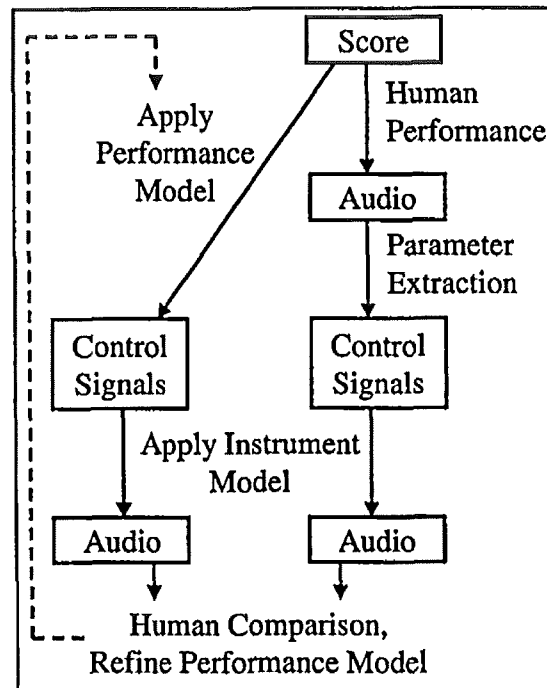


Fig. 4. Testing and refining the Performance Model by listening.

the synthesized controls to the analyzed controls to search for the problem. On the other hand, if the problem is with the instrument model, we now have a set of control signals to use in testing refinements to the model (as in Fig. 3).

To summarize, we believe that continuous controls are a key to high-quality synthesis. Synthesis algorithms should be designed to use controls that can be obtained automatically from acoustic examples. This allows an "instrument model" to be developed, tested, and refined by comparing acoustic to synthetic performances. Ideally, control signals should be musically relevant to simplify the task of deriving control signals from symbolic scores.

### OTHER APPROACHES

To clarify our approach and the main contributions of our research, we will describe some other approaches to sound synthesis. Most current commercial synthesis systems are based on MIDI. (Rothstein, 1992) MIDI was designed to represent keyboard performance information, and while not limited to keyboard information, MIDI certainly creates a mindset that is note-oriented. MIDI notes are generally started and stopped by discrete messages. This reinforces the abstraction that each note is independent, but with wind instruments, notes are *not* independent. The character of note attacks and decays is very important but is not portrayed by MIDI note-on and note-off messages. Of course, MIDI offers continuous control and system exclusive messages, which can convey more information, but this is not standard practice.

Sampling-based synthesizers illustrate the problems of note-oriented synthesis even further. With sampling, the initial portion of a note is recorded and saved with a "steady state" portion that is looped to extend the note's duration as needed. A problem with this approach is that the initial attack (in winds) is highly variable, so many samples are needed to represent different attacks. Furthermore, the shape of the amplitude envelope is also highly variable, and the spectrum of an acoustic instrument changes with amplitude. Sampling synthesis does not offer much control over the spectral content of the signal after the initial attack. Thus, while sampling can render a particular note quite well, it does not offer a range of control over spectrum, attacks, and envelopes required for high-quality wind synthesis.

Analysis/synthesis techniques and other synthesis studies share many of the problems of sampling. For example, classic studies of additive synthesis (Moorer, 1977) analyze single notes to obtain trajectories for a number of harmonics. These can be added together to reproduce the original note, but the envelope data is not so useful for synthesizing arbitrary notes. Scaling and stretching operations are possible, but they do not produce good sounding

results because scaling (for amplitude change) does not result in the proper spectral changes, and stretching (for duration change) does not result in the proper envelope shape.

In general, *the problem with sampling and additive synthesis is the focus on the note rather than either the production of sound or control mechanisms*. If we can produce sound and control it, we can synthesize notes and phrases of all kinds, but if we focus on producing only a single particular note, we have no control and no ability to produce phrases.

Other work has started by observing the spectral variation of single notes and searched for a vector basis for the range of spectra (Horner, 1997; Hourdin, Charbonneau, & Moussa, 1997; Kleczkowski, 1989; Laughlin, Truax, & Funt, 1990; Oates & Eaglestone, 1997). These approaches are also note-oriented and result in a certain set of wavetables that can be summed according to certain envelopes to approximate the original note. The result is essentially a specialization of additive synthesis, and all the limitations and problems described above apply.

Physical models (Roads, 1996) solve the problem of note-oriented synthesis, but leave open the problem of control. Learning to control an acoustic instrument is difficult and so are direct models of acoustic instruments. We focus on spectral models in order to simplify the problem of control.

## RELATED RESEARCH

Our instrument model can be viewed as a specialization of additive synthesis (De Poli, 1993) or a generalization of wavetable synthesis (Moorer, 1978). A good review of wavetable interpolation techniques is found in Horner and Beauchamp, 1996. Beauchamp (1995) demonstrated another method for obtaining an appropriate spectrum from a few control parameters for trumpet synthesis. His method relies on the observation that the spectral envelope of the trumpet can be predicted from the RMS amplitude. Given a spectral envelope and a pitch, the amplitudes and frequencies of harmonics can be computed.

The idea of timing variation as related to musical structure (Sundberg, 1991) and emotion (Canazza, De Poli, Roda, & Vidolin, 1997) are also relevant to our goal of deriving control information from scores. Chafe (1989) derived bowing information from a score and used it to control physical models in order to synthesize a string quartet, and Berndtsson (1996) used a rule-based system to synthesize the singing voice. Garton (1992) has used performance models to control and even compose for synthetic instruments. Arcos, Mantaras, and Serra (1997) use a combination of instrument models

and performance models to alter digitized instrumental performances. All of these are good examples of research that integrates performance and synthesis models.

### THE INSTRUMENT MODEL

The instrument model has a set of control functions as input and produces a digital audio sound as output. The output should be perceptually very close to the acoustic instrument being modeled. (Instrument designers should be able to model non-existent instruments as well, but this is beyond the scope of this article.) We first describe the basic synthesis technique, then we describe how data is analyzed to create instrument models, and finally, we describe experiments to validate the assumptions and simplifications implied by this approach.

Our instrument model is based on the idea that at every time-point the spectrum is nearly harmonic. This instantaneous harmonic spectrum is determined primarily by a small number of parameters, called modulation sources. These have meaning to the performer and can be automatically measured and extracted from real performances. As to the trumpet, the primary modulation sources are the current RMS amplitude and fundamental frequency. (A simple modification to provide inharmonic attack transients to this model will be presented later.) Synthesis is very simple and fast (see Fig. 5):

- (1) Create curves which describe the value of amplitude and pitch (frequency) as a function of time,
- (2) Use the values of amplitude and frequency to access (index) a database of stored representations of spectra, and
- (3) Output the spectrum.

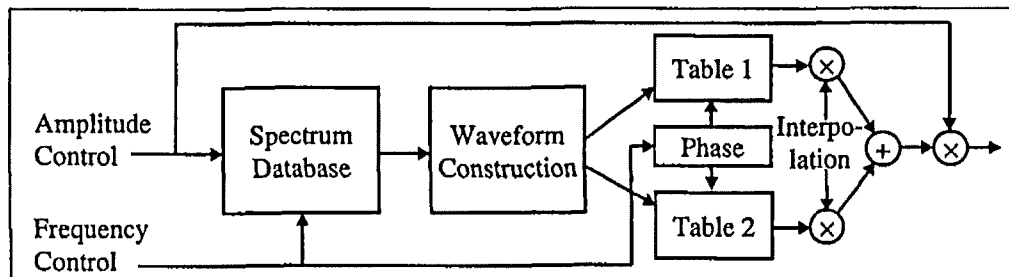


Fig. 5. Block diagram for Spectral Interpolation Synthesis. Amplitude and frequency control signals are used to select spectra from the Spectrum Database. Interpolation is used to make smooth transitions from each table to the next. Frequency also controls the phase increment of a table-lookup oscillator, and amplitude also scales the resulting waveform.



The spectrum is represented by an array of numbers describing the relative weights (amplitudes) of harmonics. We typically store a two-dimensional array of these sample spectra. When this two-dimensional database is accessed by the instantaneous amplitude along one dimension and the frequency along the other, we interpolate among the four nearest spectrum samples to yield an output spectrum.

To output the spectrum, we generate a wavetable (which represents one period of the periodic sound in the time domain) and interpolate audio samples from the table to produce the required frequency. In this way, we produce one period of the sound to output. Many options are available regarding exactly how and when the wavetables are computed. At one extreme, spectra are computed at the fundamental frequency. That is, each period of the synthesized sound can be computed from the stored spectra. To make the algorithm more efficient, we compute only 20 spectra per second and produce every sample by interpolation between two tables. This effects a smooth, continuous spectral change. The result is scaled by the instantaneous RMS amplitude to produce the proper amplitude fluctuations in the sound. After that step, our synthesized sound has controlled fluctuations in timbre, pitch, and amplitude. Because the resulting spectrum is the smooth interpolation of spectral samples, we call this Spectral Interpolation (SI) Synthesis.

Tables are created with matching phases to avoid any phase cancellation during the interpolation. We assume the phase information does not have an audible effect on the synthesized sound. We do not store phase information for the harmonics in the spectra, only their relative amplitudes. This basic model is further extended along a few dimensions as described in the following sections.

### **Spectral Data Collection**

To create an instrument model, we need an array of spectra indexed by pitch and amplitude. We create this array by analyzing actual performances. This process is described in detail in this section.

To measure spectra, we use the SNDAN utility package, implemented by Beauchamp (1993). We used SNDAN's phase-vocoder-based algorithm for extracting spectra, RMS amplitude, and frequency. This algorithm carries out pitch-synchronous Fourier Transforms over two windowed periods of the sound to measure the instantaneous spectrum. This method is intended for tones with nearly constant pitch. We also used SNDAN's other algorithm, based on MQ analysis (McAulay & Quatieri, 1986), to analyze phrases with several pitches.

A trumpet player played simple notes, increasing or decreasing the amplitude level at a moderately fast speed, covering as wide a dynamic range as

possible. We measured the maximum playable dynamic range to be under 30 dB.

During these measurements, we also discovered that it is easier for the player to produce a steady decrescendo over a large dynamic range than to produce a corresponding crescendo. Therefore, we extracted the spectra from notes with decreasing amplitudes to obtain spectra for different amplitude levels.

Collecting spectral data is quite simple: using SNDAN, we obtain a spectrum for each period. We scan through this data and retain only spectra at which the amplitude function crosses predetermined thresholds. We do the same measurement for several pre-defined pitches, and in this way build the database of spectra indexed by amplitude and frequency.

How many harmonics should spectra contain? To reduce computational power and storage requirements, the number should be as low as possible. At the same time, we do not want to sacrifice the quality of the synthesized sound in any degree. We did not find any audible difference between synthesized phrases using 30 harmonics or the maximum possible number (the Nyquist rate divided by the fundamental frequency), so we decided to use 30. In some cases, we limited the number of harmonics according to the frequency of the highest note in the phrase under study. This simplifies the implementation, but may not be generally applicable.

### **Spectra and Wavetables**

We can see that the step of computing wavetable data from spectral data has to be done approximately twenty times in a second, the rate at which we introduce new spectra for interpolation. Instead of storing the spectral data, we could store the corresponding time domain wavetable directly, saving those wavetable computations. This can be done if the phases of the harmonics do not need to be changed during synthesis, and as a matter of fact, we used this technique before we combined the pure SI synthesis with spliced attacks. However, if the phase distribution of the harmonics can be different from note to note (which is the case with spliced attacks, as we will see later), this technique can not be applied.

Therefore, we store amplitude spectra and compute wavetables as necessary. An interesting opportunity exists to simulate body resonances and frequency-dependent radiation patterns by inexpensive multiplies in the frequency domain, although we do not take advantage of this at present. There are several ways to compute time-domain data from the spectrum (IDFT, IFFT, etc.). At this point, we perform a simple addition of sinusoids. In the future, especially in a real-time environment, this issue should be addressed more carefully.

### Testing the Model

We have presented a synthesis technique and a corresponding analysis technique. It is now time to ask whether this approach really works. The synthesis technique is simple and makes a number of assumptions that need to be verified:

- (1) The frequency of the fluctuation of the timbre, that is, the spectral sample rate, is relatively low,
- (2) The phase information in the spectrum is perceptually irrelevant, and
- (3) The instantaneous spectrum is basically a function of the absolute instantaneous value of the RMS amplitude and pitch, and nothing else.

We performed a number of experiments to either verify these assumptions or, where the assumption does not exactly hold, to enhance the basic model. In the following sections, we describe the synthesis model in more detail through these experiments and enhancements.

### *Spectral Sample Rate*

In previous work Serra, Rubine, and Dannenberg (1990) analyzed tones from a variety of wind instruments to obtain spectra, and reconstructed the original tones by spectral interpolation. In that early form of the method, the only control function was time: The output spectra were sampled at certain times from an original digitized tone. Tones were reproduced from those sample spectra.

Listening tests determined that the results were good and that slow spectral sample rates (in the range of 5 to 20 spectra per second) were adequate and produced high quality representations of the original tone. This laid the groundwork for the present investigation.

### *Phase and Inharmonicity*

Listening tests also proved our second assumption, namely that the phases of the harmonics do not carry perceptually relevant information. The tests also determined that some instruments (notably trumpets, trombones, and saxophones) were not convincing when resynthesized. This was the result of inharmonicity within the attack portion of the sound. Many instrument tones have a very characteristic and inharmonic attack portion. As this portion has a significant audible effect, and as the basic Spectral Interpolation model is only capable of producing harmonic sounds, an extension of the model became necessary. Aside from attacks, the conventional sounds of wind instrument tones are essentially harmonic.

Fortunately, we have found that it is possible to use recorded attacks to give the impression of a natural attack. Carefully made splices are used for the transition from recorded attacks to synthesized tones. Spliced attacks can be

automated and incorporated easily into the basic Spectral Interpolation model. Note that attacks begin with a stopped airflow and silence, so there is no need to splice from synthesized tones to the beginning of the attack, only from the attack to the tone. The method of splicing is the following:

- (1) Start with a recorded attack. Shorter attacks are better for several reasons: The memory requirement is smaller with shorter attacks. Also, a short attack contains less envelope shape information, making it possible to attach the attack to different envelope shapes. On the other hand, the sampled attack should be long enough to capture the whole inharmonic part. Also, the end of the attack must settle into a harmonic structure because the amplitude and phase of each partial must be continuous at the splice point (that is, at the end of the sampled attack and the beginning of the Spectrum Interpolation).
- (2) Measure the phases and amplitudes of the harmonics as well as the overall RMS amplitude of the sound at the end of the attack. These attributes are stored with the attack.
- (3) Use these attributes to determine the initial phase and amplitude for the first spectrum used by Spectral Interpolation synthesis.
- (4) After that, to avoid phase cancellation, compute all wavetables using this same set of phases. Compute all subsequent spectra according to the frequency and RMS amplitude control signals.

This procedure requires that the phases of wavetables be adapted to the phases of the attacks. Therefore, we cannot precompute wavetables, but rather must compute them from amplitude spectra on the fly. Also notice that the final amplitude distribution in the attack may not exactly match the amplitude distribution predicted by the Spectral Interpolation model. To avoid any discontinuity, we simply adopt the final amplitude distribution of the attack as the "correct" spectrum, and interpolate to the spectrum generated by the Spectral Interpolation model during the first  $1/20^{\text{th}}$  second of synthesis.

The last parameter to be matched is the overall RMS amplitude. The amplitude control function can specify any value at the splice point. Meanwhile, the amplitude value at the end of the attack is arbitrary. The attack has to be scaled so that those two amplitude values match. At first glance, attack scaling is questionable, but in practice, we generally start with relatively loud attacks and scale them down for softer attacks. We found that this simple method produces good sounds.

As we mentioned, finding the ideal length of the sampled attacks could be automated by observing the relations of the frequencies of the partials, but for the time being we have not implemented this technique, and we set the length manually using listening tests to determine the right duration. We have had good results using attacks of about 30ms for the trumpet. This is long

enough for our analysis software to begin analyzing partials to determine their amplitude and phase.

As with the spectra, a collection of sampled attacks for different pitches and amplitude levels are necessary to produce realistic sounds, and one can even imagine that there might be more than two (amplitude and frequency) sources for variation in attack quality. For our experiments, we use a sample for each pitch we want to resynthesize with spliced attacks. We used only one attack per pitch and scale to achieve different amplitudes rather than use a set of attacks with different amplitude levels. In spite of this disregard for context, "transplanted" attacks sound authentic. This is encouraging because it indicates that a large library of specialized attack sounds is not necessary.

Working with attacks, we developed a strategy for their use. When there is a clear stoppage of air by the tongue, a spliced attack is appropriate at the next note onset. In the case of slurs and legato phrases, the spliced attack should be omitted. This simple rule can be used to automate the insertion of attacks, and the attack sample can be chosen purely on the basis of pitch.

#### *Sample Rates*

An interesting finding is that it is beneficial to decouple the amplitude control sample rate from the spectral interpolation rate. Originally, we stored the spectrum as the absolute amplitudes of the harmonics. Thus, the overall amplitude was encoded into the spectrum. We used 20 Hz as the combined spectral sample rate and amplitude control rate. We experienced that this was not fast enough to track rapid changes in the amplitude (e.g., between slurred notes) and introduced audible artifacts. To avoid this problem, we factored out the overall amplitude information from the stored spectra. At present, the stored spectra (which, as we already described, are accessed at a 20 Hz rate) are normalized, and the desired amplitude fluctuation is realized by multiplying the output of the core spectral interpolation (which has an essentially constant RMS amplitude) by the amplitude control function. This function is realized by a piece-wise linear curve, and resynthesis studies have shown a sample rate of 100 Hz is adequate to reproduce even fast amplitude fluctuations.

#### *Slurs*

Another possible source of "difficult" transient sounds is slurs, that is, continuous transitions from one pitch to another. We tried cross-fades from one pitch to the next, as suggested by Strawn (1985). We also tried treating the slur simply as a rapid pitch transition, using pitch and amplitude curves derived from acoustic performances. In this method, slurs are represented entirely in the frequency and amplitude control functions, so no special synthesis treatment

is required. Both approaches sound good in listening tests, so we use the latter method.

### Sources of Spectral Variation

After these first experiments, we knew that spectral interpolation could produce good sounds, but we still needed to know whether our main assumption is true: that the current spectrum can be determined solely from the current RMS amplitude and fundamental frequency. It is well known that upper partials grow faster than linearly with increases in amplitude, and this is why wind instruments sound brighter at higher amplitudes. To test the hypothesis that RMS amplitude determines the spectrum at a given pitch, we analyzed acoustic performances of a trumpet playing slow and fast increasing and decreasing amplitudes on a constant pitch. After analysis, the amplitude of a selected partial is plotted as a function of overall RMS amplitude. Note that the resulting graph is not a function of time, but an indication of how the amplitude of a selected partial relates to overall amplitude. If the spectrum is determined solely by RMS amplitude, then we expect curves plotted for different performances will be similar. If other factors, such as rate of amplitude change, are important, then we should see a separation in the plots. In other words, data analyzed from increasing amplitude changes and data analyzed from decreasing amplitude changes will form two clusters if direction of change is important.

In Figure 6, we show examples of the plotted data. Those figures compare the spectra measured at increasing and decreasing amplitudes. Each plot shows the amplitude of one particular harmonic, measured 5 times from notes with increasing amplitude, and 5 times from notes with decreasing amplitudes.

In the figures we see well-defined curves from the maximum amplitude (0 dB) down to a certain level, below which the curves become mixed with a substantial amount of noise. That "threshold" is well-below the softest sustainable tone at the first harmonic, but increases as we go higher in the harmonics, reaching around  $-10$ dB at the tenth harmonic. These curves indicate that RMS amplitude accounts for much of the spectral variation. Some separation can be discovered between the two sets of curves, but it seems to be insignificant. (Also, there seems to be no difference between the curves obtained from rapidly versus slowly changing amplitudes.)

We conducted listening tests using timbre databases obtained from both sets of measurements (increasing and decreasing) to compare their possible effect. We did not find significant (if any) audible differences. We conclude that the instantaneous amplitude (not the rate or direction of change) is enough to determine the timbre to a close approximation.

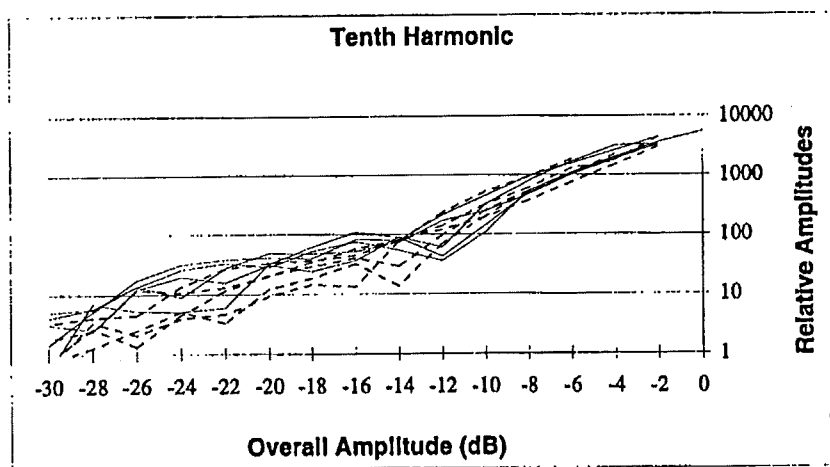
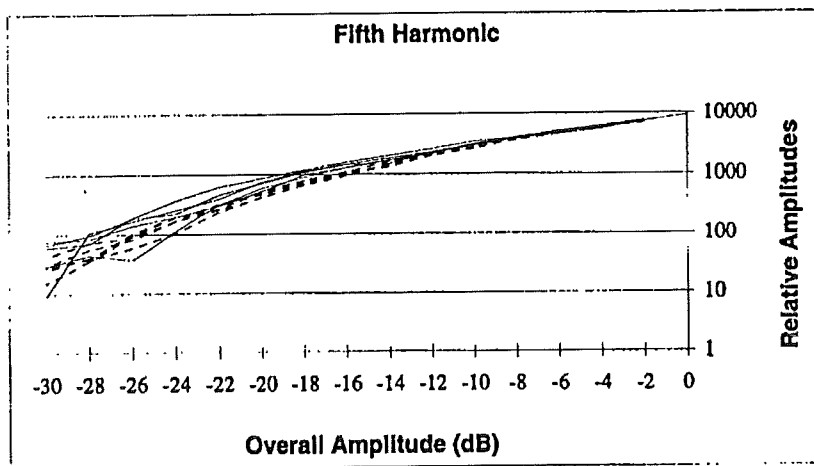
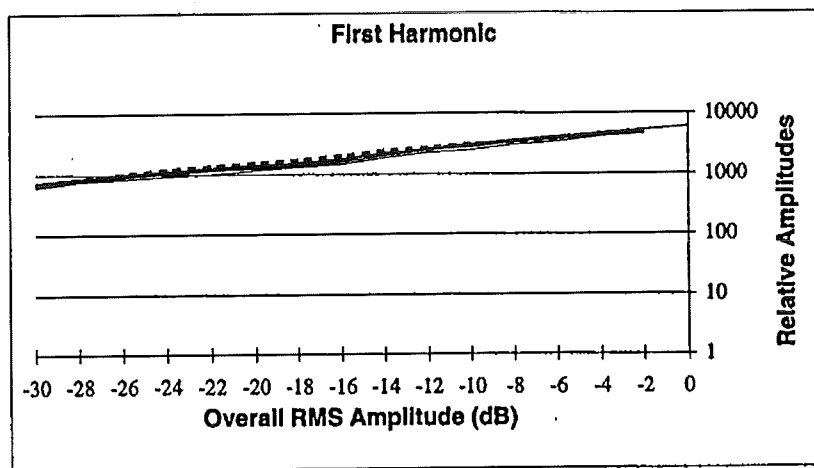


Fig. 6. Relative amplitudes of harmonics, measured from notes with increasing amplitude curves (solid lines) and decreasing amplitude curves (dashed lines), plotted logarithmically in the same amplitude range.

The noise in the curves indicates that there may be other factors at work, so either we need to identify other sources of variation or we need to show that this variation is not perceptually significant. A third possibility is that there is simply randomness in the spectrum. It might be interesting to model the randomness rather than ignore it, but we have not explored this possibility. Since there could be any number of sources of variation, we resort to resynthesis and subjective listening experiments to show that the observed variation is not significant. *Listening tests based on Figure 2 indicate that amplitude and frequency are adequate to produce very realistic tones.*

However, we found that if tables are built from one performance and then tones are synthesized using amplitude and pitch signals from another, there can be some very subtle differences between the synthesized and the original tones. The difference seemed to be related to brightness. Knowing that a trumpet player can modify the brightness slightly with changes in embouchure, we tried to use it as a third control function and attempted to build spectral tables based on different embouchure settings. We were able to cause some spectral variation based on different analyzed tones, but our results were inconsistent. This is an interesting area for future research. Meanwhile, although we sometimes observe slight variations in brightness between original and resynthesized tones and phrases, the synthesized results sound very realistic. If very slight timbral differences are critical, we expect that spectral brightness can be modified, for example, by translating the coordinates of the spectral lookup table.

Our trumpet model is capable of rendering very fine performances of classical music. We chose classical music because the playing style is more "pure" and free of effects than, say, jazz styles. (We believe that jazz performances can also be rendered, but this will require more research.) Given this instrument model, we can turn to the problem of the performance model.

### THE PERFORMANCE MODEL

The goal of the performance model is to automatically generate control information for the instrument model based solely on information in a symbolic score. (We assume that the score is available in a machine-readable form, so we are not concerned with interpreting images of printed pages.) So far, our performance model is simple and limited, but it has produced some very nice performances. Note that we have only studied the trumpet in detail. In this section, we will describe our progress to date.

Our work on the performance model has focussed on the fine details of control rather than the more coarse features of duration and pitch. For the most part, we will use duration and pitch as specified by the score, assuming that



we can apply the work of Sundberg (1991), Clynes (1987), and others to create more musical performances. This still leaves open the problem of appropriate amplitude and frequency envelopes, slurs, attacks, vibrato, and other fine details.

Clynes (1984) introduced the idea that amplitude envelopes should be modified according to context. In particular, if a passage is moving upward in pitch, the performer will tend to increase blowing pressure, and this increase will affect the note envelope, giving it relatively more amplitude toward the latter part of the note and relatively less at the beginning, as compared to a note in a descending passage. This idea can be extended to other situations. For example, the pitch contour associated with any three consecutive notes can be (up, up), (up, down), (down, up), or (down, down), and the magnitudes of "up" and "down" can vary from zero (unison) to a large leap.

Clynes (1987) originally studied envelope generation and developed rules by listening to synthesized tones. Sundberg, Askenfelt, and Fryden (1987) also developed performance models using analysis-by-synthesis techniques, but this work addressed discrete parameters of amplitude and timing rather than continuous control signals. In our work, we present a trumpet player with exercises designed to elicit different amplitude envelopes under controlled conditions, and we generalize from these examples.

### **Trumpets and Other Instruments**

Although most of our in-depth studies have used trumpet tones, we have used Spectral Interpolation Synthesis to realize clarinet, alto saxophone, bassoon, and trombone tones. These tests followed the analysis/synthesis paradigm, so control signals and spectra were derived from specific tones and then used to resynthesize those tones. We expect that Spectral Interpolation Synthesis models of most wind instruments will be successful for two reasons. First, the resynthesis experiments indicated that other winds can be synthesized given the proper control information. Second, none of the techniques for creating trumpet instrument models seem to depend on any specific feature of the trumpet. Of course, there could be specific problematic features of other instruments, such as the noise of a flute. Further work is needed to discover the applicability and limitations of Spectral Interpolation Synthesis.

### **Trumpet Envelope Features**

Earlier, we described the analysis of trumpet tones in order to build a mapping from instantaneous amplitude and frequency to spectra. For the performance model, our concern is to build mappings from features in the score to amplitude and frequency envelopes. Therefore, we took recordings of performances, segmented them into individual notes, and extracted amplitude envelopes for study.

We obtained a number of interesting results. The most striking (and in retrospect, perhaps the most obvious) feature we detected in the trumpet tone amplitude envelopes is a sudden decay near the end of the note (see Fig. 7). This decay has a simple explanation: these notes are articulated with the tongue, meaning that the tongue stops the flow of air until the moment of attack, at which point the tongue is lowered to release air to the lips. In order to articulate the next note, the tongue must, at some point, stop the air from the previous note. This causes the rapid decay. If there is even a slight rest or silence before the next note, the rapid decay will not be present. This observation immediately yields an important rule for trumpet synthesis: when there are consecutive tongued notes (no slur in the score), there should be a rapid decay followed by a tongued attack. Recall that tongued attacks are synthesized using sampled attacks in the instrument model.

Aside from rapidly tongued articulations, the amplitude envelope is quite smooth. This is to be expected because amplitude corresponds directly to driving pressure. As a simple experiment, try to blow air through a small opening in the lips while rapidly pulsing or modulating air pressure with the diaphragm. The first author, a trumpet player, can modulate pressure at about 8 Hz, but even this low rate feels very rapid, uncomfortable, and foreign to trumpet playing. Thus, we conclude that the trumpet player modulates driving pressure with muscle movements that are typically well below 8 Hz. We call this the "breath envelope". Superimposed upon this relatively slow fluctuation is the more rapid modulation of the tongue, which by stopping the air completely can effect rapid attacks and decays. We call these the "tongue attack envelope" and "tongue stop envelope," or collectively the "tongue envelopes".

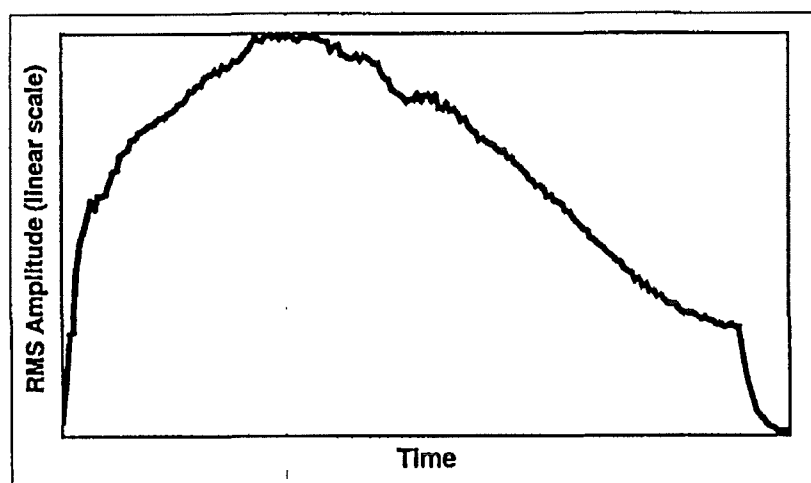


Fig. 7. A typical trumpet amplitude envelope (an Ab4, *mezzo forte*, from an ascending scale of tongued quarter notes). Note the rapid decay at the end.

Another feature of interest is a slight rise in the envelope before the tongue stop. Perhaps this is due to the tongue pushing out air as the passageway is blocked. Performances from other trumpet players and other instruments might shed more light on this phenomenon.

Although we often think of notes as having zero amplitude at the beginning and end, this is not always the case, even when the tongue completely closes the air passageway and the valves momentarily close off the bore of the instrument. Slurs have especially high amplitude from one note to the next (see Fig. 8).

In addition to this qualitative analysis of envelopes, we performed some statistical tests on the envelopes we extracted from acoustic performances. One test, performed by Hank Pelerin, measured the time position of the centroid (center of mass) in ascending versus descending pitch phrases. We found a significant difference between the centroids of notes played within different pitch contours. This confirms that Clynes' ideas on envelope shape are consistent with data measured from actual performances.

### Modeling Envelopes

Envelopes are typically described by a set of parameters. For example, the ADSR envelope (Adams 1986) of analog synthesizers is described by the attack time, decay time, sustain level, release time, and overall duration. An ADSR envelope is far too crude for realistic instrument synthesis, so a more

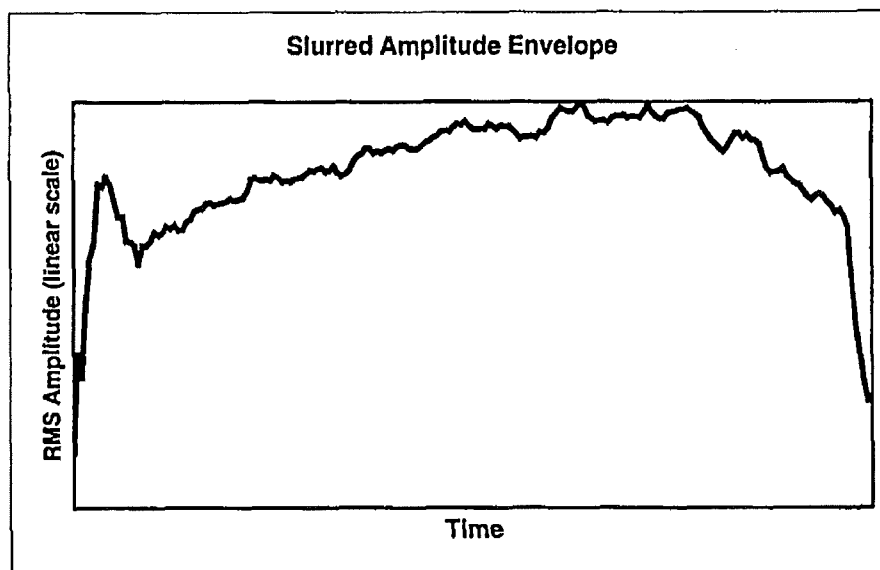


Fig. 8. A typical trumpet slurred amplitude envelope (a C5, *mezzo forte*, from an ascending scale of slurred quarter notes). Note the high initial and final amplitudes compared to Figure 7.

sophisticated model with more parameters is required. On the other hand, since parameters have to be computed, there is an advantage to having fewer parameters. The model should be just general enough to realize the necessary envelopes.

Our performance model is based on the idea of a smooth breath envelope, controlled by the player's chest and diaphragm muscles, modulated by rapid tongue envelopes. The tongue attack envelope is modeled simply using a smooth rise with a duration of around 32 to 34 ms. The tongue stop envelope is more complicated, starting with a "hump" of about 30 to 50 ms and ending with an exponential decay of about 50 to 60 ms. The "hump" is scaled so that when the tongue envelope is multiplied by the (decreasing) breath envelope, the peak of the hump has roughly the same amplitude as the beginning (see Figure 9).

Even with tongued articulation, the amplitude does not fall to zero unless there is a pause or rest between notes. The performance model is careful to match the envelope amplitude from the end of one note to the beginning of the next to avoid audible clicks, and the synthesis model insures signal and phase continuity by using just one oscillator.

For slurs between notes, we see a dip in the amplitude (again, see Figure 9) that is shallower and shorter than the tongued articulation. We use the same envelope model (so "tongue envelope" is misleading), but the parameters are changed. For example, if the pitch change is upward, and the articulation is a slur, then the envelope only decays to 20 percent of the maximum amplitude and the decay lasts only about 30 ms.

In addition to tongue envelopes, we need a breath envelope. Because the breath envelope has a simple shape, we decided to take the envelope from an

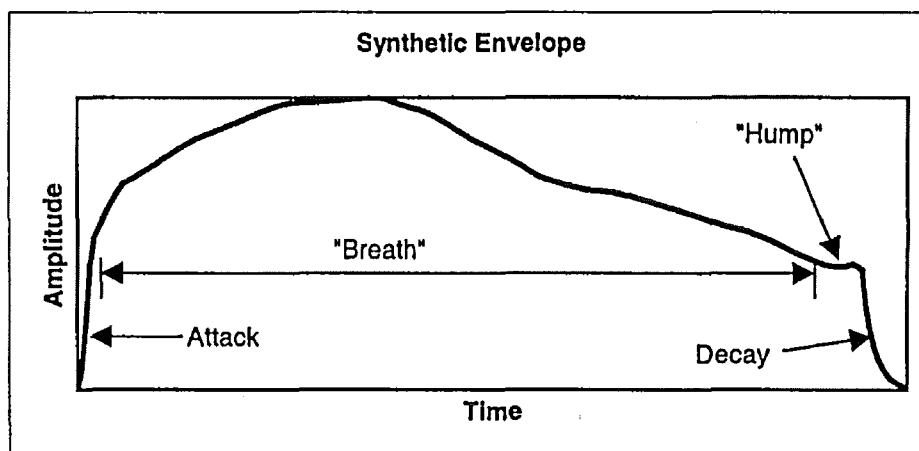


Fig. 9. A synthetic envelope showing the attack, breath envelope, Ahump@ and decay. Envelope features are exaggerated for clarity.

acoustic performance of a half note as a prototype. By extracting a region of this prototype and then stretching to the desired length and amplitude, a variety of breath envelopes can be constructed. These seem to satisfy our needs with just a few control parameters. Figure 10 illustrates the prototype envelope. To obtain a rising envelope characteristic of a note in an ascending phrase, an early portion of the envelope can be selected (labeled "A"). To obtain a diminishing envelope characteristic of a note in a descending phrase, a later portion of the envelope can be selected (labeled "B"). An envelope for a slurred note, where the amplitude is relatively constant, can be obtained by extracting the central part of the overall envelope (labeled "C"), ignoring most of the rise and decay.

Since the overall shape is simple, we suspect other formulations of the breath envelope are possible and would work equally well. For example, the beta functions suggested by Clynes (1984) generate similar shapes.

As observed by Sundberg (1987), amplitude generally increases with pitch. For our model, we took actual data from the performance of ascending and descending scales and used curve-fitting software to obtain a rule for amplitude change. We found that a linear relationship between fundamental frequency and pitch offers a reasonable fit to the data and sounds good, at least for mezzo-forte playing. More elaborate rules are undoubtedly necessary to deal with dynamic levels specified in the score, and in fact, we have already modified our rule to give a slight increase in dynamics to the final note in a phrase.

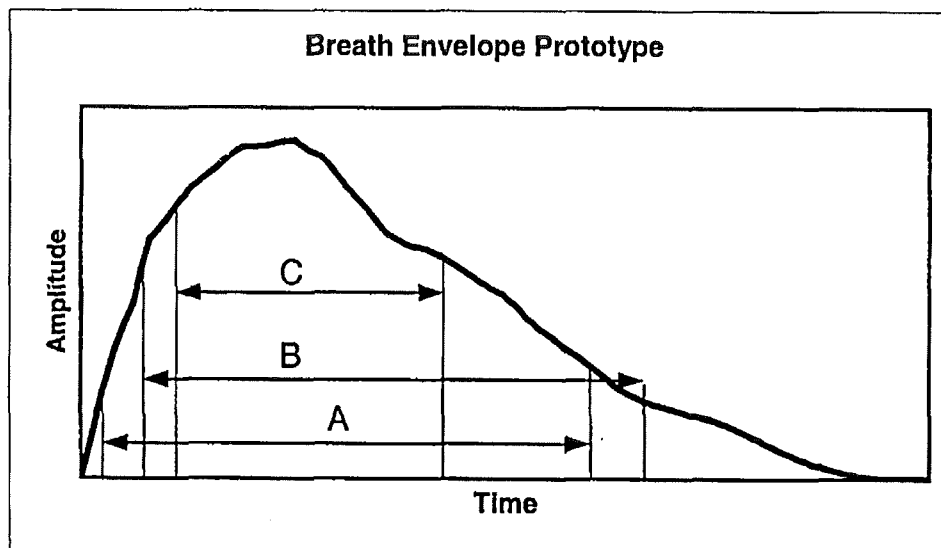


Fig. 10. Different regions (for example those labeled A, B, and C) of the prototype breath envelope are extracted and stretched to form a variety of breath envelopes. The breath envelope is then (usually) multiplied by tongue envelopes to form a complete envelope such as shown in Figure 11.

Frequency envelopes are also important, and a performance with a steady, unwavering frequency will stand out as artificial when compared to an acoustic recording. To date, we have used frequency envelopes extracted from acoustic performances for our model. In fact, we stretch and transpose the same function for every note without any problem. In the future, we need to incorporate a vibrato model and look more carefully for trends that we have overlooked.

The exact values for all envelope parameters are computed based on note durations, pitch, pitch contour, and articulation (tongued, slurred, or not contiguous). Note that articulation refers to transitions both from the previous note and to the subsequent note. Parameter values are based on an analysis of notes performed in different contexts. Figure 11 illustrates an acoustic envelope and a synthesized envelope with manually chosen parameters. For each parameter, we study example envelopes like this one to see what score attributes (e.g., duration, pitch, articulation) seem to affect the parameter. We then construct a simple linear or exponential function, essentially curve fitting, that predicts the parameter from the score attributes. When listening tests uncover a problem, we look for a relationship between the score and good parameter values that was overlooked. Then, we elaborate the model to take into account this new relationship.

So far, we have constructed the performance model manually, but as our techniques become routine, it becomes clear how we might use machine learning techniques to automate and perhaps improve on the model-building process. We assume that there is an algorithm or function for converting a set of

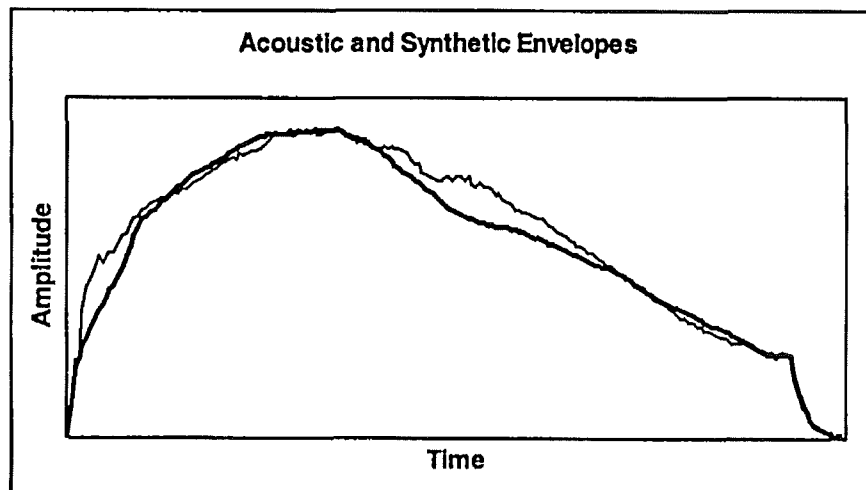


Fig. 11. An actual amplitude envelope (thin line) and a synthetic envelope (heavy line). The synthetic envelope parameters were adjusted by hand to achieve a close match to the original.

parameters to an envelope. We want to learn a method for selecting or computing parameters based on attributes of the score, for example a quarter note slurred to a half note a major third up in pitch. (In this example, the attributes are quarter, half, and a major third. In practice, there will be many more attributes.)

The first step is to obtain discrete envelope parameters from performed examples of various phrases. We have done this by hand, but it should be possible to use gradient-descent methods to pick parameters that minimize the least-squares error between the original and synthesized envelopes. (The current model has 7 parameters that are not directly measurable from the original.)

Now, assume we have a set of attributes from the score (inputs) and a set of envelope parameters (outputs). We want to learn a function from inputs to outputs. This is a supervised learning problem, and there are many machine-learning techniques that can be applied. Some relevant techniques are neural networks (McClelland & Rumelhart, 1988), function approximation (Boyan, Moore, & Sutton, 1995), and case-based reasoning (Leake, 1996).

## RESULTS

We have used our performance model to synthesize the beginning of the Haydn Trumpet Concerto. Various durations and articulations are required, so this piece is very difficult for most synthesis techniques. Our system is able to render a very realistic performance from the score with only a few added annotations on phrasing. The renderings are perhaps too perfect to sound human: attacks are clean, and the sound is very consistent throughout. It is very difficult to provide an objective evaluation. Sound examples are available at <http://www.cs.cmu.edu/~rbd/sis.html>.

## FUTURE WORK

Our success so far is very encouraging, but there is still much work to be done. Our goal is an orchestra of synthetic instruments and performers capable of realizing a notated score with realism. With the methodologies we have introduced, we believe this goal is possible. So far, most of our modeling has involved step-by-step manual tasks, but we envision a highly automated process of building instrument and performance models.

The process starts with the capture of acoustic performances. We need good methods to measure absolute amplitude as opposed to relative amplitude,

because absolute amplitude will be the primary control. Players perform various passages to produce examples of different pitches, amplitudes, articulations, and phrasing.

Next, spectral tables are constructed by locating source material with different combinations of pitch and amplitude. For instruments with inharmonic attacks, we also need to isolate a selection of attacks from the performed data. The instrument model construction can be automated almost entirely, including listening examples to help evaluate the model.

The performance model is more complex, but with experience, this too can be mostly automated. Assume that a general envelope model can be developed based on the ideas of breath and tongue envelopes. We then generate examples that provide good envelope parameters for a given set of score attributes, and we use machine learning to find a relationship between score attributes and envelope parameters. To obtain training examples, we first segment the acoustic performance into individual notes. Segmentation of phrases into individual notes currently requires hand editing, but since the score is known, segmentation can be automated by tracking frequency changes and looking for amplitude changes at note transitions. Then, an optimization algorithm derives sets of parameters that best fit the extracted envelopes. Next, machine learning uses these examples to derive functions from score attributes to envelope parameters.

In addition to low-level mappings from score attributes to performance data, a successful performance must incorporate a musical sense that operates at the level of phrases and sections. We believe that at these higher levels of abstraction, instrument-specific details tend to be less important, so one general set of performance rules should apply (perhaps with minor extensions and modifications to deal with different musical styles). Other researchers have made interesting progress in this area, and their results will provide most of the necessary high-level knowledge of music performance.

Thus, we believe that *instrument models and performance models can be constructed automatically with little more than a set of performances by a skilled musician*. The next step will be to develop the present trumpet model further and to try it out on more examples. We need to apply these techniques to other instruments and work toward automating the whole modeling process. While our specific findings apply to the trumpet, we hope that the more general ideas of relating envelopes to the breath, tongue, and musical structure will apply to all winds.

To many composers, the idea of recreating what humans already do well is boring at best. Although we enjoy the technological challenge and appreciate the advantages of a clearly defined research goal, we acknowledge that playing trumpets and other acoustic instruments may not be the best application of



computers. Future work can investigate the use of Spectral Interpolation Synthesis to create hybrid instruments by interpolating between different instrument models. Systematic alterations of instrument and performance models might also create a new performance practices. Finally, our techniques might be used to make new and interesting synthetic instruments more musically sophisticated and appealing.

### SUMMARY AND CONCLUSION

We have described research that takes a significant step towards high-quality synthesis of wind instrument performances. The strength of this work does not lie in any particular signal processing strategy or device, but rather in the overall perspective and approach in which synthesis is viewed as a combination of performance knowledge and instrument characterization. These two concepts are linked by the carefully chosen intermediate representation of time-varying frequency and amplitude control signals.

By design, our approach enables us to extract appropriate control signals from actual performances and then use these to refine both the performance model and the instrument model. This approach factors the overall problem into two parts, both of which now seem tractable.

The key to the instrument model is the realization that the time-varying spectrum is determined almost entirely by relatively simple modulation sources such as amplitude and frequency. Experiments with the trumpet have shown that this assumption holds even during rapid amplitude fluctuations. Because of this relationship, an instrument can be modeled as a function from modulation sources to spectrum. This mapping can be captured automatically from acoustic performances.

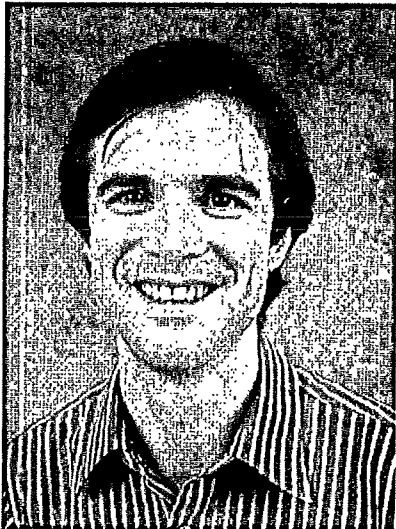
The remaining synthesis problem is to generate appropriate modulation. A careful analysis of trumpet envelopes has produced a wealth of information and guidance, and a performance model has been created to produce modulation control signals from score information. Many details of the control signals appear to be related directly to features of the score. For example, a tongued note has a characteristic attack and decay, and the amplitude centroid of a note occurs later in time if the note is in an ascending line. A trumpet performance model has been constructed, and it is clear how machine learning techniques can be applied.

We are just beginning to realize and evaluate musical phrases created by our models. In the future, we plan refine the models using a more extensive set of test cases, automate the modeling process, and apply these techniques to other instruments.

## REFERENCES

- Adams, R. (1986). *Electronic music composition for beginners*. Dubuque, IA: Wm. C. Brown Publishers.
- Arcos, J.L., de Mantaras, R.L. & Serra, X. (1997). SaxEx: a case-based reasoning system for generating expressive musical performances. In *Proceedings International Computer Music Conference 1997* (pp. 329–336). San Francisco: International Computer Music Association.
- 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. & Horner, A. (1995). Wavetable interpolation synthesis based on time-variant spectral analysis of musical sounds. *Audio Engineering Society Preprint*, No. 3960 (Paris Convention, February), pp. 1–17.
- Berndtsson, G. (1996). The KTH rule system for singing synthesis. *Computer Music Journal*, 20, 76–91.
- Boyan, Moore, and Sutton, (Eds.) (1995). *Proceedings of the Workshop on Value Function Approximation*. Tech. Report CMU-CS-95-206. Pittsburgh: Carnegie Mellon University School of Computer Science.
- Canazza, S., De Poli, G., Roda, A. & Vidolin, A. (1997). Analysis by synthesis of the expressive intentions in musical performance. In *Proceedings International Computer Music Conference 1997* (pp. 113–120). San Francisco: International Computer Music Association.
- Chafe, C. (1989). Simulating performance on a bowed instrument. In M. Mathews & J. Pierce (Eds.), *Current directions in computer music research* (pp. 185–198). Cambridge MA: M.I.T. Press.
- Clynes, M. (1984). Secrets of life in music: Musicality realised by computer. In *Proceedings of the 1984 International Computer Music Conference* (pp. 225–232), Computer Music Association.
- Clynes, M. (1987). What can a musician learn about music performance from newly discovered microstructure principles (PM and PAS)? In A. Gabrielsson (Ed.), *Action and perception in rhythm and music*, (pp. 201–233). Publications issued by the Royal Swedish Academy of Music No. 55.
- De Poli, G. (1993). Audio signal processing by computer. In G. Haus (Ed.), *Music processing* (pp. 73–105). Madison: A-R Editions.
- Garton, B. (1992). Virtual performance modeling. In *Proceedings of the 1992 International Computer Music Conference* (pp. 219–222). San Francisco: International Computer Music Association.
- Horner, A. & Beauchamp, J. (1996). Piecewise linear approximation of additive synthesis envelopes: A comparison of various methods. *Computer Music Journal* 20, 72–95.
- Horner, A. (1997). A Comparison of wavetable and FM parameter spaces. *Computer Music Journal* 21, 55–85.
- Hourdin, C., Charbonneau, G. & Moussa, T. (1997). A sound-synthesis technique based on multidimensional scaling of spectra. *Computer Music Journal*, 21, 56–68.
- Kleczkowski, P. (1989). Group additive synthesis. *Computer Music Journal* 13, 12–20.
- Laughlin, R., Truax, B., Funt, B. (1990). Synthesis of acoustic timbres using principal component analysis. In *ICMC Glasgow 1990* (pp. 95–99). San Francisco: International Computer Music Association.
- Leake, D., (Ed.) (1996). *Case-based reasoning: Experiences, lessons, and future directions*. Cambridge: MIT Press.
- McAulay, R. & Quatieri, T. (1986). Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 34, 744–754.

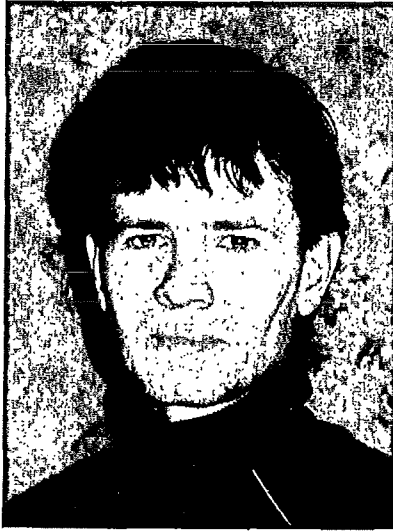
- McClelland, J. & Rumelhart, D. (1988). *Explorations in parallel distributed processing*. Cambridge: MIT Press.
- Moorer, J.A. (1977). Signal processing aspects of computer music — A survey. *Proceedings of the IEEE*. (July).
- Moorer, J.A. (1978). How does a computer make music? *Computer Music Journal*, 2, 32–37.
- Oates, S. & Eaglestone, B. (1997). Analytical methods for group additive synthesis. *Computer Music Journal*, 21 21–68.
- Roads, C. (1996). Physical modeling and formant synthesis. In *The Computer Music Tutorial* (pp. 263–316). Cambridge: MIT Press.
- Rothstein, J. (1992). MIDI: A comprehensive introduction. Madison, WI: A-R Editions.
- Serra, M.-H., Rubine, D. & Dannenberg, R.B. (1990). Analysis and synthesis of tones by spectral interpolation. *Journal of the Audio Engineering Society* 38, 111–128.
- Strawn, J. (1985). *Modeling musical transitions*. Ph.D. Dissertation, CCRMA/Department of Music, Stanford University.
- Sundberg, Askenfelt, & Fryden (1983). Musical performance: A synthesis-by-rule approach. *Computer Music Journal* 7, 37–43.
- Sundberg, J. (1991). Music performance research: An overview. In Sundberg, Nord, & Carlson (Eds.), *Music, language, speech and brain* (pp.173–193). Wenner-Gren International Symposium Series, Vol. 59. London: Macmillan.



Roger B. Dannenberg  
 School of computer Science  
 Carnegie Mellon University  
 Pittsburgh, PA 15213  
 USA  
 Tel.: +1-3214-412-268-3827  
 Fax: +1-3214-412-268-5576  
 E-mail: roger.dannenberg@cs.cmu.edu  
<http://www.cs.cmu.edu/~rbd/home.html>

Dr. Dannenberg is a Senior Research Computer Scientist on the faculty of the Carnegie Mellon University School of Computer Science, where he received a Ph.D. in 1982. His current work includes various projects in music understanding and the design and implementation of languages for music performance and synthesis. His work on automated accompaniment led to the Piano Tutor, a multi-media expert system for teaching beginners to play the piano, and Vivace, a practice aid used by thousands of students on a daily basis. He is co-editor of the book *Multimedia Interface Design*.

Dr. Dannenberg is also a trumpet player and composer. In addition to playing jazz with artists such as Roger Humphries and Eric Kloss, Dr. Dannenberg's most recent musical efforts involve interactive real-time computer graphics and computer music. His latest commission is a work for the Foro de Musica Nueva in 1998.



Istvan Derenyi  
Computer Science Department  
Carnegie Mellon University  
Pittsburgh, PA 15213  
USA  
E-mail: [derenyi@cs.cmu.edu](mailto:derenyi@cs.cmu.edu)

Istvan Derenyi was born in Budapest, Hungary, 1969. At present he participates in the Computer Music Project at the Computer Science Department of the Carnegie Mellon University and is working toward his Ph.D. degree in Computer Science. He received his first M.S. degree in Electrical Engineering in 1994. He also holds an M.S. in Technical Education and a B.S. in Economics.

Before coming to the CMU, he worked in the music industry for two years. As a musician, he has composed several short pieces.

His computer music related interest covers the different aspects of digital sound synthesis and signal processing, as well as the creation of composition supporting tools.