Mark Phillips, Jeff Barish, and Rob Maher
3Com Corporation/U.S. Robotics Corporation
Boulder, CO 80301, USA

Presented at
the 109th Convention
2000 September 22-25
Los Angeles, California, USA

AN AUDIO ENGINEERING SOCIETY PREPRINT

THE MODELING AND SYNTHESIS OF MUSICAL SIGNALS WITH PRISM

Mark Phillips, Jeff Barish, Rob Maher

3Com Corporation/U.S. Robotics Corporation
4840 Pearl East Circle, Suite 201E
Boulder, CO, 80301 USA
PH: 720-406-2214 FAX: 720-406-2235
Mark.Phillips@Ieee.org, Rob_Maher@3Com.com, JeffBarish@Earthlink.net

Abstract - Multidimensional signal modeling of musical notes can produce expressive and realistic synthesis for musical applications. The PRISM synthesis technology employs time-varying signal models using envelopes for pitch, power, and timbre evolution. PRISM also models the statistical variations of parameters over time in order to produce indefinitely long, non-looped sustains while maintaining the character of the original note's sustain. The models from multiple signals are combined into a multidimensional parameter space. During synthesis, the desired location in the parameter space is mapped from input control values. Based on the desired location, the neighboring data sets are interpolated to form an intermediate, smoothly varying data set for synthesis. Using signal models allows for intuitive and powerful modifications of note properties both offline and real-time.

I. INTRODUCTION

In this paper, the fundamental concepts that the PRISM technology uses to model and synthesize sounds are presented [1]. First, the strengths and weaknesses of two of the most prevalent synthesis techniques are presented in order to show the need for a synthesis method with expanded capabilities. Then a description of the PRISM technology applied to the most basic scenario of modeling and synthesizing a single, static sound is presented. Next, the procedure is generalized to model and synthesize single, time-varying sounds. Then the procedure is generalized further to model and synthesize sounds from a group of time-varying sounds. Finally, applications and embellishments of the technology are discussed.

II. OVERVIEW OF SYNTHESIS TECHNIQUES

Two of the most prevalent synthesis techniques currently used in electronic musical devices (such as keyboards and sound cards) are wavetable synthesis and physical modeling.

Wavetable Synthesis

Wavetable, or sample playback, systems rely on sample rate conversion of prerecorded sounds [2]. Often, various tricks are played in order to enhance the apparent expressiveness of the notes. These tricks include velocity sensitive layering, velocity sensitive switching, velocity-sensitive filter parameters, and a host of other such techniques. These approaches do add variability to the "loud" and "soft" versions of a particular note, but this variability can sound like either two notes playing at once, or a sudden change from a "quiet" timbre to a "loud" timbre. Another limitation of wavetable synthesis is that sustains may sound predictable and repetitive due to "looping" a waveform segment.

Since not every note in an instrument's scale is typically stored due to the large memory requirements, pitch shifting is used to play a particular wavetable at multiple pitches. Unfortunately, pitch shifting introduces unnatural changes in timbre in a recorded tone. In particular, when a tone is pitch shifted up it often sounds tinny or "munchkinized". Analogous distortions occur when a recorded tone is pitch shifted down too much. These distortions limit the

amount that a recorded tone can be pitch shifted. Therefore, to cover the pitch range of the instrument, multiple tones must be recorded so that each individual tone is only pitch shifted over a limited range. One-fourth to one-half octave pitch shift ranges are typical.

Despite attempts to cover up the defects of looping with interesting amplitude envelopes, there are still often undesirable artifacts using this approach to simulate sustains. If loops are long -- on the order of half a second -- then there are usually audible discontinuities at the loop edge. To smooth these discontinuities crossfading is used across the loop splice. This results in a kind of "wah-wah" chorusing artifact. Short loops, for example the length of a single pitch period, don't suffer from these problems, but are nonetheless problematic since they have a completely static timbre, sounding like an electronic oscillator. For certain instruments, especially those such as the piano which have non-harmonic overtones, it is often difficult to find a single period loop which does not have discontinuities at boundaries.

Traditional wavetable synthesizers also often suffer from a general lack of expressiveness and naturalness. This is due to the fact that every time a note is played the same recording is used. No real variation occurs except that introduced by randomization of the amplitude envelope, between one realization of a tone and the next.

Physical Modeling

Physical modeling [3] is a technique that offers realistic, expressive sounds, but has several drawbacks. The first is the inherent difficulty of designing a mathematical model for the instrument. This requires a thorough knowledge of the physics of the instrument and the ability to express the wave equations within the instrument. If suitable equations can be found, they are usually detailed and computationally expensive and must be simplified to enable implementation. Of course, this simplification typically reduces the realism of the output.

Another significant drawback of physical modeling is that many desired sounds are not produced by physical acoustics, making the modeling of wave physics impossible. Finally, not all physical models are compatible. The physical model for a stringed instrument typically varies in form from the model for a brass instrument. This prevents a straightforward creation of a new sound with characteristics from the two physical models.

Introduction to PRISM

PRISM stands for "Parametric Resynthesis by Interpolated Signal Models." For a wide palette of tones, PRISM offers additional flexibility and expression over other synthesis methods. This flexibility stems from the ability to acquire a working model for a given signal, which can be manipulated and/or interpolated with other models in real-time. Instead of modeling an instrument by its physics, the PRISM technology models sounds using the recorded *sounds* themselves. PRISM can model a particular sax played by a particular performer on a particular day, given some choice recordings of that performance. Multiple dimensions of expression can be recorded, which PRISM can capture and recreate with a MIDI controller.

III. PRISM FUNDAMENTALS

Preparation

In order to provide greater realism and enhanced control, the pitched and noise portions of the sound are modeled and synthesized separately. The goal in the separation is to align the separated signals with the two types of modeling involved in PRISM: pitched signal modeling and noise signal modeling. The pitched signal modeling assumes harmonic content. The noise signal modeling models colored random noise. For this reason, the separation of the original signal into a "pitched" signal and a "noise" signal separates the signal into harmonic content and random noise. Inharmonic content can be put in the pitched signal or the noise signal. The modeling works best when the signal fits the "harmonics plus noise" description, but inharmonic content can be modeled to a certain degree by harmonics or noise.

As will be described, there are links between the pitched and noise models to make sure that the pitched and noise portions of the synthesized sound are correlated correctly. The resulting outputs are summed to form the desired note.

Modeling a Single Static Note

To be able to model a group of time-varying sounds (or instrument), the analysis and synthesis must first be able to model a single static sound.

Any static (non-time-varying) sound can be described by three characteristics: pitch, power, and timbre. For a harmonic sound, pitch and power are easily definable and intuitive, whereas timbre is trickier to specify. For our purposes timbre will be defined as all information required to specify a sound, except power and pitch. For truly harmonic sounds, this is the relative amplitude of each harmonic (since the frequencies are known to be multiples of a fundamental frequency). For a noise signal, this would be the spectral magnitude of the noise.

Pitch

Pitch is defined as the fundamental frequency in this paper. Detection of the pitch is a huge field by itself and will not be discussed in detail here. It will be assumed that an estimated pitch, *f0*, has been derived from the pitched signal to be modeled.

Power

Power can be measured according to root mean square (RMS) or average absolute value (AAV), or a more perceptually correct measure of volume. The challenge in measuring the power is ensuring that the section of the signal measured is sufficiently long to measure the lowest frequencies present in the signal.

For the pitched signal, one efficient method entails determining the power over one period of the signal. Assuming an accurate measure of *f0* exists, the power can be accurately measured over this period (1/*f0*). If the period were unknown, a much larger duration would have to be measured in order to minimize the accuracy-limiting effects of measuring the power over a span with a non-integer number of periods.

For the noise signal, which has no period, a larger segment must be measured in the power analysis procedure. The more low frequency content that the noise is allowed to have, the larger the duration of the signal that must be measured for reliable power measurements. By limiting the low frequency content of the noise, the power measurements can be accomplished with shorter segments.

Timbre

Signal modeling is another large field that will not be discussed in detail here. The typical premise of signal modeling is to assume that the observed signal is the output of a filter that has white noise as the input [4] (see Figure 1). The result is a set of filter coefficients, which when used to filter white noise give the best (in the least squares sense) approximation of the spectrum of the signal. The white noise spectrum is flat and the filter colors the spectrum to make it look like the signal to be modeled. This approach works well for the noise signal. An autoregressive (AR) model has been found sufficient for this application.

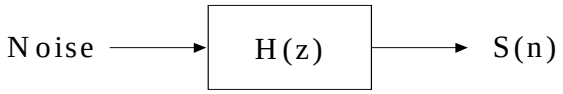

Figure 1: Noise feeding a filter, H(z), with the signal, S(n), as the output.

One way to model the timbre of a *harmonic* signal is to model the signal as if it had no harmonic structure. First, the spectrum of the signal is calculated and the phase information is discarded. The harmonic structure is smoothed out to leave the overall shape of the spectrum (spectral envelope). The resulting signal is then modeled with AR signal

modeling as described above, which yields model filter coefficients. Figure 2 shows an example signal spectrum, a smoothed version of the spectrum, and the resulting AR model's spectrum.

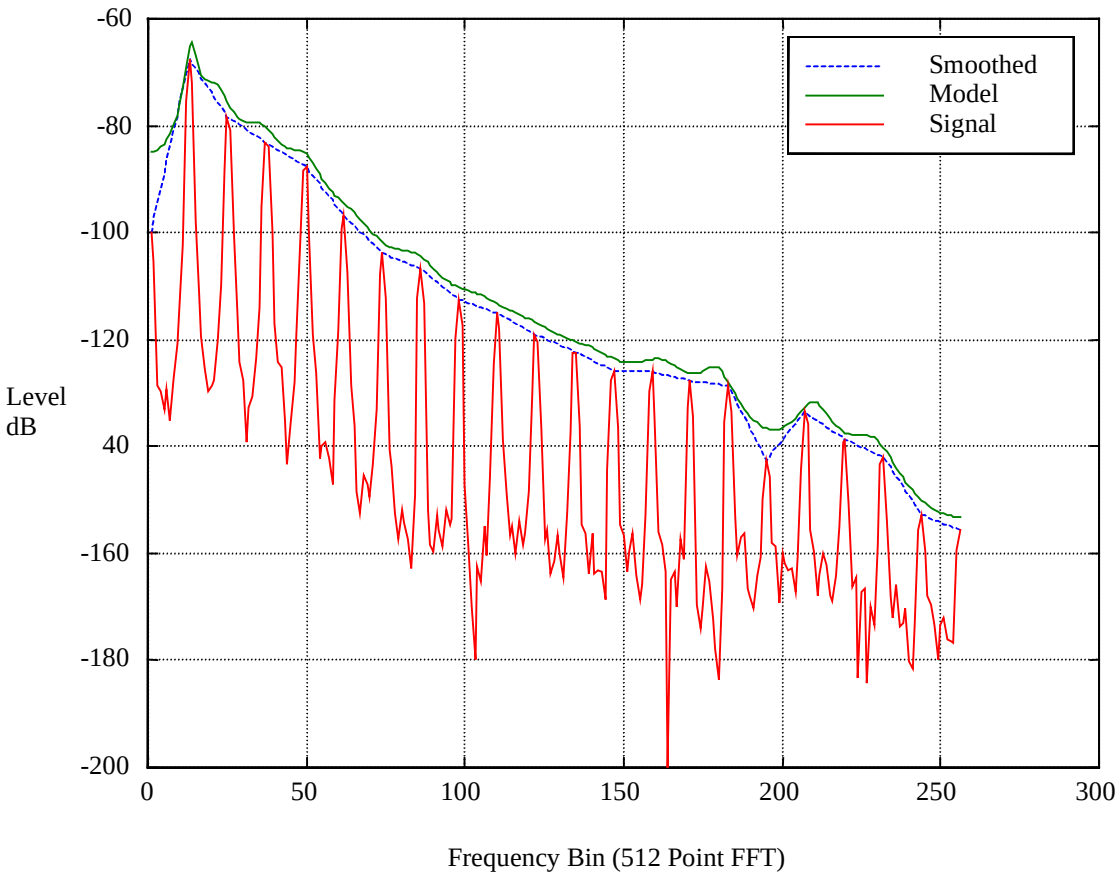

Figure 2: Example of a signal's spectrum, smoothed signal's spectrum, and model's spectrum.

Typical model orders for good results vary from 4 to 32, with 16 being common. Using too low of a model order will fail in reproducing the spectrum accurately. Sometimes using too high of a model order will produce unwanted variation in the filter coefficients in the time-varying case in addition to being more expensive.

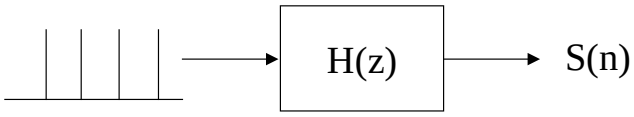

Figure 3: Using an impulse train for a model of a harmonic signal, S(n).

For harmonic signals, an impulse train can be used as the input instead of white noise (see Figure 3). The spectrum of an impulse train is an impulse train in frequency where each harmonically located peak is the same height (harmonically white). Running this through the model filter colors the peaks to look like the harmonic sound to be modeled. This models only the spectral magnitude of the signal.

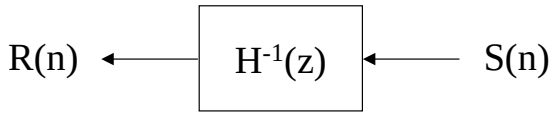

Figure 4: Running the signal, S(n), through the inverse filter model, $H^{-1}(z)$, to produce the residual signal, R(n).

In order to model the phase of the signal, an alternative signal must be found to be the input to the model filter. Running the signal backwards through the filter yields the "residual" signal (see Figure 4). This signal has the necessary phase information to produce the correct output. If the modeling were perfect, the residual signal would be harmonically white. Since any finite-order model has error in general, the residual will not be truly white, despite the assumption of the modeling, but it will be sufficiently white for the desired application. For a harmonic signal, the residual will be periodic since the system is linear and time-invariant. If the residual is run forward through the filter, the original signal will be produced exactly.

For harmonic modeling, a loop of the periodic residual can be used to excite the filter and produce better results than a looped impulse. The loop from the residual is called an "excitation." In order to work in the goal application, the excitation table is normalized, both in power and in length.

The filter imparts the general spectral qualities to the sound while the excitation contains the fine spectral qualities and the phase information. It is advantageous to model as much of the timbre as possible with the filter coefficients, as will be explained below.

Synthesizing a Single Static Note

Given the power, pitch, and timbre (contained in a number of filter coefficients and an excitation stored in a table), the model of the original sound can be synthesized.

The excitation table is looped at a rate that produces the fundamental frequency, *f0*, derived from the pitch detection. This produces an excitation signal that is fed into a filter with the coefficients derived from the signal modeling. The power of the output is measured and the output signal is amplified to have the detected power. This is the basic core of the PRISM technology (see Figure 5).

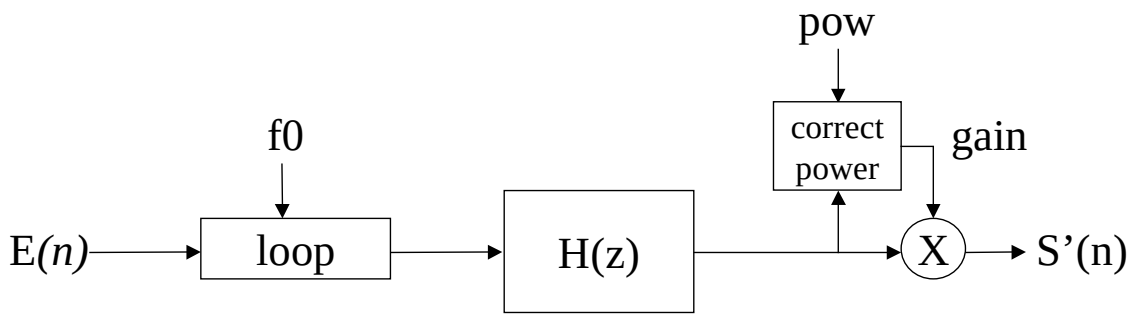

Figure 5: Excitation table, E(n), is looped according the to the fundamental frequency, f0, filtered by H(z) and amplified to have the correct power, pow.

The power correction must be performed after the timbre filter because the filter modifies the power of the signal. It is true that a static filter modifies a static excitation in a fixed way. This could enable the power correction factor to be calculated offline and not take up time in the synthesis. However, in the goal application unknown coefficients will be generated. When the coefficients are unknown, the power cannot be predicted, so the power correction must occur during synthesis.

Modeling a Single Time-Varying Note

In order to model a signal with time-varying characteristics, several additional considerations must be made.

Windowing Scheme

Modeling a time-varying signal requires measuring characteristics at varying times. This can be accomplished with the common short-time Fourier transform (STFT) approach. An overlap-add (OLA) framework separates the time-varying waveform into windowed segments, or frames, where it is assumed the waveform is effectively static. Since the perceptual integration time of the human hearing system is roughly 20 milliseconds [5], segments on the order of 20 ms long are used. Longer segments may be used for sounds with low frequency content. Once the segments are isolated, the segment of sound is modeled in the same way as a static sound. This gives a measure of pitch (in the case of the pitched signal), power, and timbre for sequential segments of the time-varying sound. Effectively, the time-varying sound is represented by a pitch envelope, a power envelope, and filter coefficient envelopes where each point in each envelope is the value needed for a successive frame.

Excitation for Time-Varying Signals

An excitation is still needed for the pitched signal. Though a separate excitation could be used for each segment of the original signal, the storage and computation requirements would be immense. For a given sound, it is assumed that there is some continuity of phase and that one excitation will sufficiently model the fine spectral structure for the entire waveform.

For pitched signals with noticeable dynamic phase changes, additional excitations may be necessary to sufficiently recreate the original timbre of the sound. In this case, two or more excitations can be derived and interpolated over time during synthesis. Excitation interpolation will be discussed in more detail below.

Statistical Modeling

The attack and release portions of a note often exhibit transient behavior. The decay portion of the note is usually labeled as the section after the transients of the attack where the note “settles down” into the sustaining portion of the note. Most notes can have a sustained section where the interior portion of the note has relatively consistent behavior. In contrast to the other regions of a note, the sustain can continue indefinitely.

Traditionally, sustains are implemented with loops. Looped sustains sound static and are predictable because they are the same with each repetition. Alternatively, a long portion of the sound can be recorded and played back. Though this can increase realism, it requires more memory and still requires a loop or static section in order to sustain indefinitely.

It is preferable to record, process and store the data for a note that is sustained just long enough to exhibit its characteristic sustained behavior. Then behavior of the sustain’s parameters can be modeled and reproduced for an indefinite time in a realistic and non-deterministic manner. As an additional benefit, significant reductions can be made in the amount of memory necessary to recreate a long note.

The statistical modeling of the parameters in the sustain region of the note takes the form of a moving mean multiplied by near-periodic variations and random variations. Near-periodic variations include note features such as vibrato and tremolo, whereas random variations include the natural unsteadiness and randomness of real sounds. For this reason, the data must be separated into these three categories. This modeling approach applies to the pitch, power, and timbre filter coefficient envelopes.

The very low frequency movements (less than about 2 Hz) and progressions in the data are defined as the trend. A trend includes the slow variations of a mean over time. Trends would include note features such as the pitch decreasing or the amplitude increasing over the course of the sustain. The data in the desired region is filtered by a low pass filter both forward and backward to preserve phase. The low pass filter’s cutoff frequency can be adjusted in order to fine-tune what is considered part of the trend. Once the trend is established, it can be modeled as a highly decimated envelope represented by long linear segments.

The detected trend is divided out of the data to leave the deviations. Using a multiplicative model has the added benefit of scaling the deviations by the trend value. This is especially appropriate for the power envelopes where the variations often decrease in scale as the trend decreases. This effectively normalizes the variations for better modeling. During synthesis, the variations are scaled appropriately by the trend values.

The remaining variations are further separated into near-periodic variations and random variations.

One method for this separation is to assume that the near-periodic variations are related to tremolo and vibrato and are usually larger in amplitude than the random noise variations. The frequency spectrum of the variations can be examined. The frequency of the peak of the frequency spectrum gives an estimate of the mean frequency of the near-periodic variations. Using this frequency as a guide, there can be an additional separation of the lower frequencies (assumed to be near-periodic variations) from the higher frequencies (assumed to be random variations). Vibrato frequencies are typically 2 to 10 Hz.

If a more accurate separation of near-periodic and random variations is needed, adaptive filtering can perform the separation and allow the frequency ranges of the near-periodic and random variations to overlap.

Once the near-periodic variations are isolated, each period is isolated and its amplitude and length are recorded. The single period loop is stretched to a set length (e.g. 32 samples) and all loops are averaged together. This forms a characteristic shape of the near-periodic variations that is relatively immune to unrelated random variations and noise in the signal.

The amplitudes for each period are averaged to form a measure of the mean amplitude. The standard deviation of the amplitudes around the mean is used to calculate a measure of the randomness in the amplitude. All of the period lengths are converted into frequencies. The frequencies for each period are averaged to form a measure of the mean frequency. The standard deviation of the frequencies around the mean is used to calculate a measure of the randomness in the frequency.

For increased realism, the variations in the amplitude and the frequency of the near-periodic variations can be model with low-order ARMA models. However, the single shape with scaled random variations in amplitude around the mean amplitude and scaled random variations in frequency around the mean frequency sufficiently models most signals with a much lower computational cost.

The near-periodic variation model can be used in at least three applications: for capturing the vibrato in the pitch envelope, the tremolo in the power envelope of the pitched signal, and the tremolo in the power envelope of the noise signal. The difference in phase between the vibrato and each of the tremolos is also modeled and stored. Though this process starts with a looped shape, the vibrato is significantly improved from common sine wave and triangle wave implementations. The loops have variations in speed and amplitude, they have specific shapes based on the original note, and the relationship between the vibrato and tremolo is maintained accurately. Since all this information is extracted from the original recording automatically, it simplifies the sound designing process significantly and improves the realism of the result.

Once the trend and near-periodic variations of the parameter envelope are modeled, the random variations are isolated and can be modeled.

In a basic implementation, the building block is a filter with 2 zeros and 2 poles. Each random devation is modeled with general linear modeling with at least one input. A noise source is always used and additional inputs can be added. The models are determined offline, using procedures such as least-squares modeling to come up with values for the general linear model:

$$A(q)\, y(t) = [B(q)/F(q)]\, u(t-nk) + [C(q)/D(q)]\, e(t)$$

where $y(t)$ is the desired output, $e(t)$ is a noise source, $u(t-nk)$ specifies the inputs, and $A(q)$, $B(q)$, $F(q)$, $C(q)$, and $D(q)$ all describe filters. $A(q)$ can be combined with $F(q)$ and $D(q)$ so the desired output can be formulated as the sum of filtered signals.

Though there are several possible formations, the most used model is formulated as follows:

- The random variations in pitch are modeled as filtered noise.
- The variations in the power of the pitched signal are modeled as the sum of filtered noise and filtered pitch variations.

- The variations in the power of the noise signal are modeled as the sum of filtered noise and filtered pitch variations.
- The timbre filter coefficient deviations are reduced to a set of eigenvectors, typically 4. This is done by doing an eigenvalue decomposition of the filter coefficient deviations. The eigenvectors corresponding to the 4 largest eigenvalues are taken to be the most essential in rebuilding the full filter coefficient set. This gives 4 eigenvectors to be modeled and an eigenmatrix with which to multiply the 4 eigenvectors into a full coefficient set.
- The pitched signal eigenvector deviations are each modeled as the sum of filtered noise, filtered pitch deviations, and filtered power deviations. There are individual models for each eigenvector.
- For the eigenvectors of the noise signal's timbre, the eigenvectors are each modeled as the sum of filtered noise, filtered pitch deviations, and filtered noise power deviations. There are again individual models for each eigenvector.

Modeling the variations in this fashion ensures that the power deviations synthesized from the model will have a similar relationship to the frequency deviations as in the analyzed data. Similarly, the coefficients that determine the tone depend (in part) on the pitch and power deviations. The noise and pitched signal's power envelope deviations both use pitch deviations as an input in order to create a correlation between the pitched signal power deviations and the noise signal variations. This model increases the realism of the result because most signals exhibit such relationships.

Synthesizing a Single Time-Varying Note

Synthesizing a time-varying note is similar to synthesizing a static timbre. There are at least two frameworks that accomplish this task. Both rely on pulling the necessary parameters from envelopes stored in the analysis.

Overlap-Add

One approach is to formulate the synthesis according to the inverse STFT. In the same fashion as the analysis, windowed segments of output can be generated, overlapped in time, and summed to form the output signal. Each of these segments is synthesized with static parameters from the corresponding analysis window. Successive blocks have evolving characteristics. The window shape insures that the sound changes smoothly.

For window i, the excitation table is looped at a rate that produces the fundamental frequency from the i^{th} analysis window. This produces an excitation signal that is fed into a filter with the coefficients derived from the i^{th} analysis window. The power of the output is measured and the output signal is amplified to have the detected power, accounting for the imparted power of the windowing function.

Overlap-add has the benefit of filter stability and smooth changes. The disadvantages are computational cost, limited flexibility, and sound quality issues. For every block of final output two blocks of the excitation are generated, filtered and power corrected (in the case of 50% overlap). The fixed window length limits the flexibility of the synthesis. Low frequency notes require longer windows for successful power measurements whereas shorter window lengths make the note more responsive and can vary faster in time. If both short and long window lengths are used in the same implementation, transition windows have to be designed and stored in memory.

Sound quality can also be an issue. An audible distortion linked to the frequency of blocks per second can often be heard on lower notes. Slight phase discrepancies can hinder the sound quality in the overlap regions.

Real-time Coefficient Interpolation

The second method used for generating time-varying notes is to vary the parameters of the time-varying synthesis constantly. There is a continual excitation and continual filtering in response to a continual change of parameters.

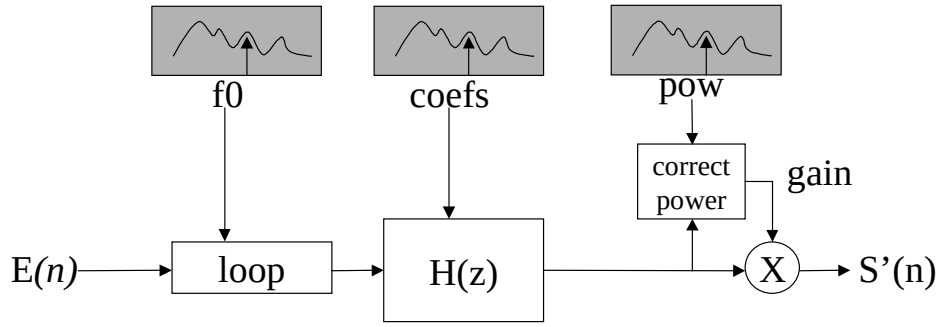

Figure 6: The pitch, f0, timbre coefficients, coefs, and the power, pow, are read from envelopes.

This method (see Figure 6) revolves around the notion of synthesized blocks of output where the beginning of the block is associated with certain parameter values and the end of the block is associated with new parameter values. The next block starts with the previous values and evolves to a new set of values. This can be accomplished by linearly varying many of the parameters (including the filter coefficients) over the course of the block. This implementation will be referred to as real-time coefficient interpolation (RTCI). RTCI requires special considerations to ensure a stable filter and a predictable output.

Going into more detail, the excitation generator must be able to synthesize an excitation with time-varying pitch (as specified by the pitch envelope). The excitation table is read from at an increment determined by the desired pitch. Lowpass filtering and bandlimited interpolation may be needed for certain desired pitches in order to prevent aliasing and unwanted distortion. This will be discussed in detail below.

There are several considerations involved in implementing a time-varying filter. Simply changing the filter coefficients at the beginning of every block produces unacceptable artifacts. Good results can be achieved by linearly interpolating the filter coefficients from the beginning of a block to the end assuming a suitable filter form is used. As each filter tap is calculated at each time step, the filter coefficient is incremented. The filter coefficient increments must be calculated at the block rate.

Interpolating the coefficients for many filter forms can yield unrelated, even unstable, intermediate filters. Guaranteed stability with interpolated coefficients is mandatory for a working solution. Preferably, the interpolated coefficients would have intermediate and predictable spectral qualities as well. There are at least two filter forms that are friendly to coefficient interpolation.

One usable filter form is a direct form filter composed of second order sections. Interpolation of the coefficients of 2 different second-order (2 zeros and 2 poles) models is guaranteed to be stable. The pole movements corresponding to the interpolation of the coefficients is fairly predictable and intuitive. In order to have models with orders greater than 2, multiple sections are cascaded together.

Another choice is the all-pole lattice form. It is typically somewhat more expensive than the cascade direct form filter, but it has a couple of benefits that are extremely useful. As long as the coefficients of a lattice filter are less than 1 in magnitude, the filter is stable. This means interpolation between any two stable coefficient sets is guaranteed to produce a new filter set that is stable. In addition, the interpolation of reflection coefficients generally produces intermediate filters that sound perceptually "in between" the timbres of the filter coefficient sets being interpolated.

Another benefit of the lattice form is that coefficients can be interpolated even if the order of the involved models differ. Reflection coefficients are calculated recursively (using Levinson's method, for example). The first coefficient represents the best first-order model. When the optimal second-order model is calculated, the first coefficient does not change; only the second coefficient is added. This is true for all of the reflection coefficients. Because of this, any set of reflection coefficients can be zero padded to a longer length to make the coefficient set "compatible" with another longer length coefficient set.

Since the coefficients of the timbre filter are continually varying, the power correction stage must apply a continually varying envelope to the output. For a given synthesis block, the beginning of the block will have a previous power measurement and a previous desired power in order to form the previous correction factor. The

filter gain associated with the new coefficients must be measured at the end of the synthesis block, where the filter coefficients have evolved to the new coefficient set. The power measurement should use as few points to measure the power as possible. First, if fewer points are used in the power measurement, the measurement will be less computationally expensive. Secondly, since the output block is the result of a filter whose coefficients are changing, the more localized the power measurement is with regard to the new coefficients, the more accurate the measurement becomes.

For the pitched signal, the power can be measured over one period of the signal. Since the synthesizer generates the excitation, the exact pitch of the signal is known. For the noise signal, a larger area must be measured in the power correction procedure. The more low frequency content that the noise is allowed to have, the larger the duration of the signal that must be measured for reliable power measurements.

Once the power is measured at the end of the block, a new correction factor can be calculated. A previous correction factor exists from the end of the previous block. During the block each output sample can be multiplied by a linear interpolation of the correction factors for the beginning and end of the block. Effectively, a linear amplitude envelope is applied to the output of the time-varying filter that simultaneously removes the gain from the time-varying filter and imparts the desired power.

The linear amplitude envelope assumes that the power of the time-varying filter's output varies linearly from the beginning of the block until the end. This is not the case, but it is a very good approximation when the filter coefficients are not changing drastically in the time dimension.

Statistical Generation of Parameter Deviations

Given the models for the random variations, a random number generator feeds the random deviation models (see Figure 7). These statistical models only need to produce one pitch deviation, one power deviation, and one set of coefficient deviations per block. This keeps the cost down.

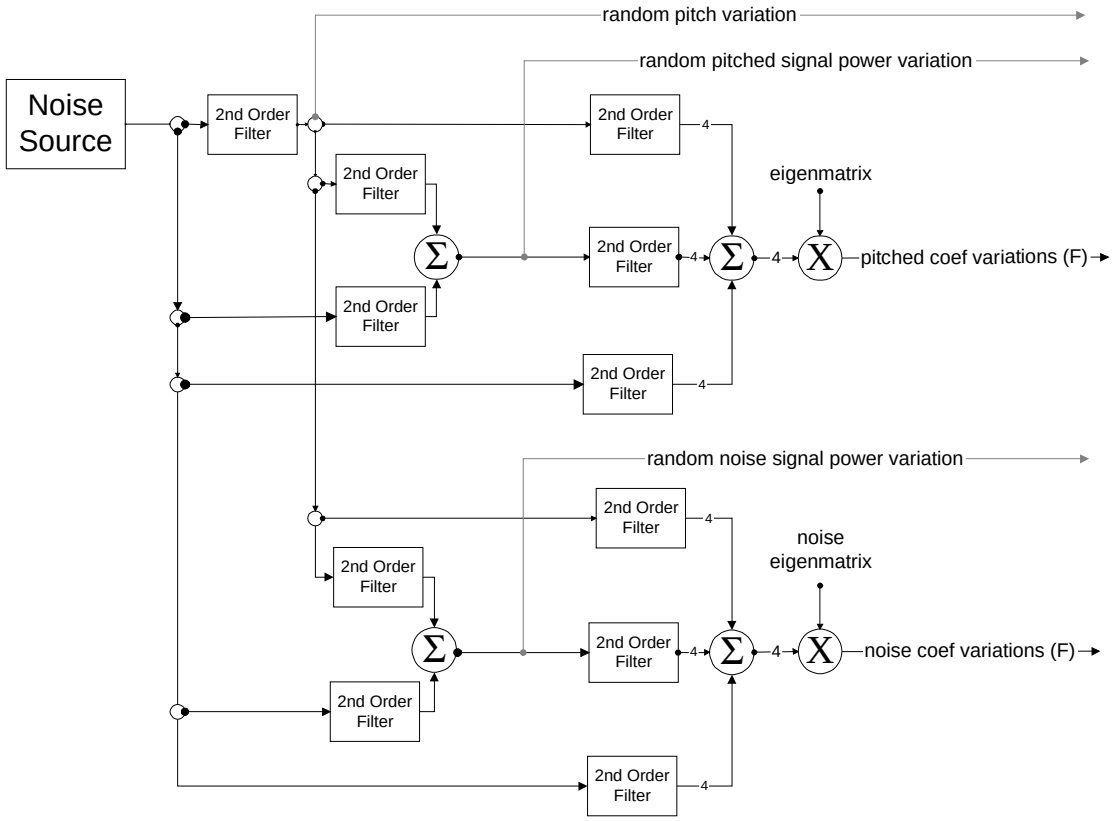

Figure 7: Topology for producing random deviations of parameters. There are F different coefficient variations.

The near-periodic deviations are produced according to the stored model information as well (see Figure 8). Once the random and near-periodic deviation values are produced, they are each added to 1 and multiplied with the trend values to produce the final values used in the synthesis engine.

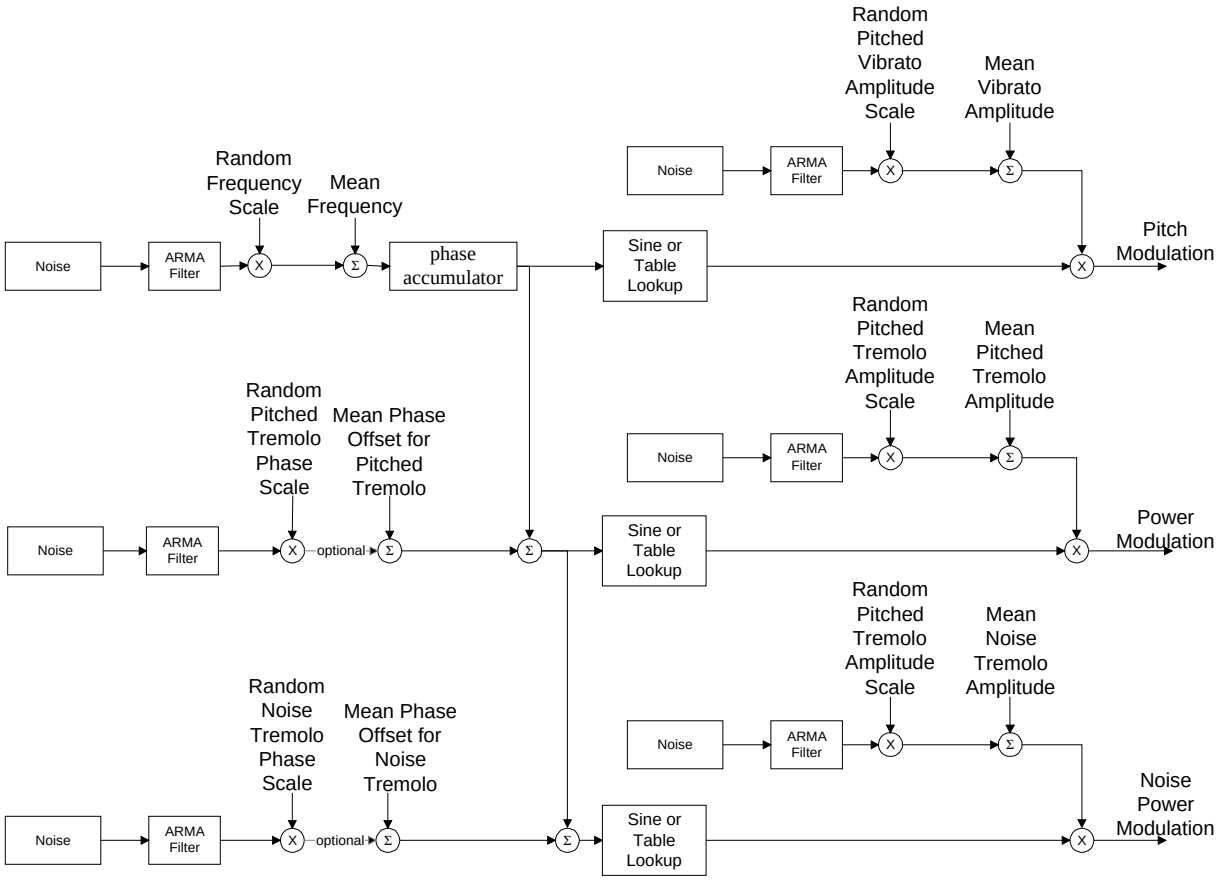

Figure 8: Topology for producing near-periodic deviations of parameters.

Modeling Instruments: Multiple Notes with Time-Varying Characteristics

Once the synthesis engine is capable of modeling and synthesizing time-varying notes, the process can be further expanded to model collections of notes and form an instrument. The notes included in an instrument can come from a collection of recordings of various notes of a real instrument, or the notes can come from various real instruments and even completely synthetic recordings designed on a computer.

For an instrument, pitch, power, and timbre can vary in other dimensions than just time (e.g. which musical note of the instrument is played). The low note on a saxophone is not only lower than the other saxophone notes (meaning the pitch will be different), but the power and timbre will be different. The attack may be slower and harsher, for example. "Intensity" is another valid dimension. As a performer plays any given note at a higher intensity, the timbre can change (it may become rougher) and so can the pitch envelope (maybe the pitch bend in the bite of the note will be more pronounced). Any dimension of control that a performer of an instrument can use can be captured. All that is needed are individual recorded notes that exhibit the desired behavior.

Once models are derived for each of the recorded notes, the dimensions of exhibited behavior need to be mapped to dimensions of MIDI control, a "MIDI-space" of sorts. The first dimension is generally the dimension of pitch. Notes are mapped to the appropriate MIDI note number according to their mean pitch. From this point on, the

mapping is flexible and is decided by the sound designer. Perhaps louder and quieter notes would be mapped to a dimension of MIDI-space that would be controlled by velocity. One dimension of MIDI-space could correspond to morphing from a clarinet to a Moog classic analog synthesizer mode. The modulation wheel could navigate this dimension. There are limitless examples of implementation, the specifics of which can be decided by the sound designer.

Multidimensional Representation

Data sets for synthesis are stored in a multidimensional parameter space called an Instrument Parameter Space (IPS). The IPS is a flexible and powerful representation of the data used for synthesis. The number of dimensions in the IPS may vary from instrument to instrument. It may have dimensions associated with pitch, intensity, and other distinctions in timbre. Since some of the data stored in an IPS has envelopes that progress with time, an IPS can be seen as having the dimension of time for those data types.

Each data set included in the IPS is associated with a specific *location*. A location is defined by specifying coordinates in each dimension, not including the time. Each dimension of the IPS (except time) is associated with a specific control variable. The control variable is mapped such that a given control variable value maps one-to-one to a single coordinate in the associated dimension. The dimension of time may be controlled by an input control value in special cases.

A desired location in the IPS is specified by the set of control variable values. Effectively, there is a mapping from a location in MIDI-space to a location in the IPS. There may or may not be a data set corresponding to this exact location. In general, the specified location has one or two neighboring data sets in each dimension. The location will have only one neighboring data set in a given dimension when the specified coordinate in that dimension is beyond the maximum or minimum coordinate of all the data sets. Typically, the coordinate in a given dimension will lie between the coordinates of two data sets. Considering an n-dimensional space, a given location may have as many as 2^n different neighbors among the stored data sets.

Requesting data for a certain location in the IPS requires that the neighboring data sets be identified. The Instrument Parameter Space must be realized with a particular structure in order to enable finding the neighboring data sets and interpolating them. There are many data structures that exhibit the necessary attributes. One convenient choice is a recursive structure that will be referred to as an N-Space.

Multidimensional Implementation

An N-Space begins with an N-Space Node, which is a vector that can have a variable number of N-Space items. Each item has a coordinate and the items are sorted in the vector according to ascending coordinates. Each item is either an N-Space Node or an N-Space Leaf. An N-Space Leaf holds the model data. Given a fixed dimensionality, a specific N-Space Node will hold a collection of either N-Space Nodes or N-Space Leaves, not a mixture of both.

The first N-Space Node represents the first dimension. If the IPS is one-dimensional, this first N-Space Node is the only N-Space Node and it contains N-Space Leaves. Each item in the N-Space Node has a coordinate. Since the N-Space is one-dimensional, the one coordinate associated with each N-Space Leaf completely describes the location of the Leaf's data. In this very simple example, the interpolation is straightforward. Given a desired location (which would have a single coordinate), the N-Space Node is searched until a Leaf coordinate is found which is less than or equal to the desired coordinate. If the desired coordinate equals the coordinate of a Leaf, then no interpolation is necessary and the desired data corresponds to the data in that Leaf. In the more general case, the desired coordinate lies in between the coordinates of two Leaves. At this point, interpolation between the data in the two Leaves is performed. Linear interpolation is the preferred embodiment, though higher orders of interpolation could be performed at higher computational cost. If linear interpolation is used, the surrounding points are linearly combined using weights inversely proportional to the distance of the desired point from each surrounding point.

In case that the desired coordinate is less than the minimum coordinate or greater than the maximum coordinate, either the data associated with the closest coordinate can be used, or new data can be extrapolated from the closest observed coordinates.

An N-Space with more than one dimension is similar to the one dimensional N-Space. Where the one-dimensional N-Space had Leaves in the first Node, higher dimensional N-Spaces will have Nodes as the items in the first Node. These "sub-Nodes" represent the second dimension and have an associated coordinate that marks their location in

the first dimension (the "super-Node"). These "second dimension" Nodes have Leaves if the space is two dimensional, or they have more Nodes if the space has three or more dimensions. Nodes will always have sub-Nodes until the last dimension is reached. The last dimension will always contain Leaves and not Nodes.

The recursive structure of the N-Space allows any number of dimensions and yields an efficient means for interpolation. The interpolation calculations are somewhat more involved for the cases of higher dimensions, but the same process is followed as in the one-dimensional case. The calculation of interpolation weights is done successively in each dimension. The desired location will have a desired coordinate for each dimension. The search begins in the first dimension. The one or two neighboring Nodes in that dimension are given the first dimensional weighting. This weighting is just like the one-dimensional example except that the resulting weights apply to the Nodes within the first dimension and everything that they contain.

Within the neighboring Nodes, the second dimension of the desired coordinate is compared to the second dimensional coordinates in those Nodes. In each of these Nodes, the neighboring coordinates are found and a new set of interpolation weights is calculated based on the relevant second dimensional coordinates.

This continues through all the relevant sub-Nodes until the last dimension is reached. In this dimension, the last desired coordinate is compared to the coordinates on the Leaves in all the neighboring Nodes, and another pair of interpolation weights is calculated for each neighboring data set based on the last dimension. All of the weights associated with a particular Leaf and its super-Nodes are first multiplied together and then the data set is multiplied by the resulting single interpolation weight. All of the resulting single interpolation weights for each of the neighboring data sets will sum to one due to the nature of the recursive interpolation. After all the relevant data sets are multiplied by their respective interpolation weights, the results are summed to form the desired data set.

Labeling Regions and Time

After all the interpolation weights are calculated based on the location of the desired note, the values within the envelopes themselves are interpolated based on the desired time, which will in general fall between the timestamps of 2 envelope points. Since time can be viewed as an additional dimension in the IPS, this interpolation is done recursively within the interpolation structure explained above.

For consistency and usefulness, the time axis is labeled in a normalized form. Every note is broken into a specific number of regions. The classic number of regions in a synthesizer is four regions, but this procedure could easily accommodate any number of regions. The four regions typically used include the attack, the decay, the sustain, and the release. Each of these regions is labeled in the data sets. Time is specified in a normalized "x.y" format. The integer specifies the region (1: Attack, 2: Decay, 3: Sustain, 4: Release). The fraction specifies the proportional time within a region. For example, if the normalized time were 2.5, it would specify the midpoint of the decay.

The actual time to progress through a particular region in a note during synthesis depends on the read-rate for that region. The read-rate describes how far to step in normalized time for a block of synthesis. The read-rate is derived from the absolute time of a region during the analysis. During interpolation, the read-rates are interpolated within the IPS such that the resulting region will have an intermediate length in absolute time.

This formulation for the time organization of the data allows for better representation of the intermediate notes. Regions within notes can be labeled consistently based on note features, which may or may not occur at the same absolute time during a note. For this reason, similar regions in various notes may have various lengths. As long as the regions are labeled consistently, intermediate notes will have the same consistent features evolving in intermediate times. If the notes in a given instrument have more than four regions of interest, more regions may be used. This time representation allows the shape *and* the time of the region to be interpolated for very smoothly varying and realistic intermediate tones.

A well-designed IPS enables very realistic reproductions of instruments, including acoustic and electric instruments. Instead of relying of "split" points, the PRISM technology has *modeled* points, and all places in between have intermediate (but smoothly varying) characteristics. Often, a PRISM instrument will require fewer analyzed notes than the number of sounds used in a corresponding wavetable instrument. It would also be possible to use the PRISM technology to extrapolate behavior not recorded or observed. Whereas wavetable can do crossfades that usually sound like two notes playing at the same time, the PRISM technology generates a true, single note that has the characteristics of both sounds.

Synthesizing Instruments

In order to synthesize whole instruments, the multidimensional interpolation must be added to the synthesis engine. Based on incoming control values, the current location in MIDI-space and normalized time value are calculated for each block of output data. Based on these values, the IPS is searched and interpolated to produce the values necessary to feed the time-varying synthesis (see Figure 9).

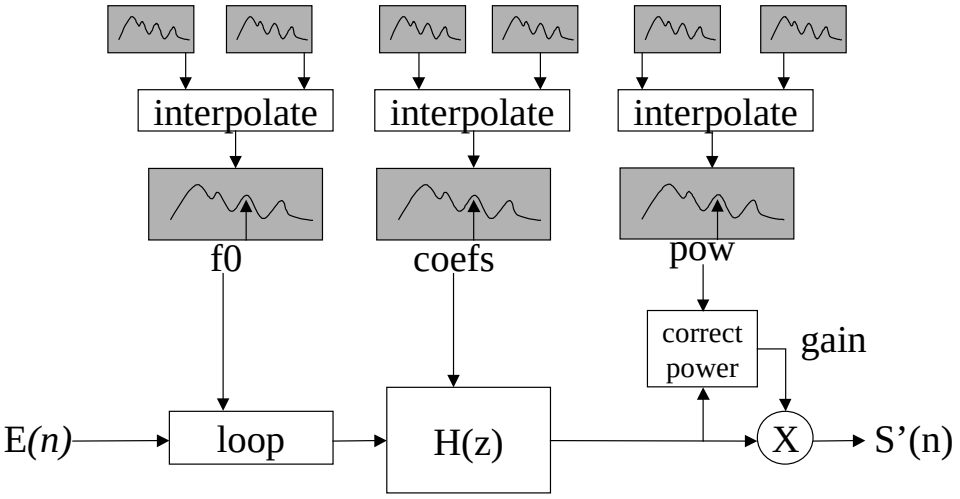

Figure 9: Simplified view of a PRISM engine with interpolation of parameter envelopes.

Most parameters are linearly interpolated in a straightforward manner, but the interpolation of the excitation table requires special attention. The excitation table can be interpolated several ways.

Since the tables are all stored at the same length, they can simply be interpolated and the result can be stored. However, this can produce unwanted phase cancellations. It is often sufficient to align all of the tables in analysis using an autocorrelation to prevent the most serious phase cancellations. However, subtle phase problems may remain. It is also possible to impart the same phase functions to all the excitation tables of a given instrument. This removes the possibility of cancellations, but can remove important phase information.

After the tables are interpolated, the bandlimiting must be based on the lowest frequency table included in the interpolation. This is to make sure that none of the component tables contributes to aliasing.

Alternatively, the tables can be stored as spectral magnitude and phase functions. The magnitude and phase can be interpolated separately and then an inverse FFT can generate the table to be used by the excitation generator. During this process, any harmonics that would lead to aliasing can be removed as well.

Overall, linear interpolation of the parameters produces sufficiently smooth and perceptually correct results. For the dimension of pitch, the interpolation is linear in the logarithmic domain.

Since the overall spectral shape is modeled by the timbre filter, the shape is maintained independent of the fundamental frequency of the excitation. One of the benefits of this approach is that many real instruments exhibit a relatively fixed resonance. As the PRISM synthesis produces notes beyond the highest or lowest notes in the IPS, the resonance is fixed, similar to the behavior of many real instruments. For the many instruments that exhibit different resonances for different notes, the resonances will shift in between the notes appropriately and then hold their shape as the fundamental frequency of the excitation goes beyond the defined extremes of the IPS.

IV. PRISM SYNTHESIZER

The PRISM technology provides a powerful and flexible synthesis core. The parameterized data representation facilitates note manipulation for realism, expression, and entertainment.

Slur Capabilities

Since many real instruments (like the saxophone, flute, and vintage synthesizers for example) are monophonic, a monophonic mode of operation can offer a more realistic representation of the modeled instrument. One essential feature of a monophonic instrument is a slurred note. A slur occurs when two successive notes are not separated by silence triggering the end of the first note to slur into the next note. The monophonic mode is particularly conducive to slurring, but slurring can be triggered in a polyphonic mode as well.

Sliding the pitch during a slur is often called portamento. The PRISM technology offers the capability to slur from the lowest note to the highest note while maintaining the character of the intermediate note during the transition. All controlling parameters (including the timbre coefficients, excitation, and power envelope) are continually interpolated during the slur to reflect the current pitch. This produces a much more realistic slur than ignoring the intermediate note properties and simply keeping the properties of the bottom or top note of the slur.

The note can slur to a specified normalized time in the new note. The target time defaults to the same timestamp of the previously played note for continuity, but it can be changed so that the slurs always end up in the attack of the new note, or halfway through the decay, etc. This can emulate the transitional effects of particular slurs.

For the most realistic slurs, the slurs of real instruments can be modeled. A real instrument, such as a tenor saxophone, behaves in a certain way when slurring between certain notes. This behavior can be recorded, modeled, and included in the IPS. When a certain slur type occurs, the relevant data can be interpolated and used for synthesis.

Time Manipulations

The normalized time within a note is a variable that can be modified based on input controls or instrument configuration settings. The progression of time is controlled primarily by the read-rate.

The rate at which the normalized note time progresses can be modified based on input control variables. Modifying the read-rate can heighten the realism of an instrument. If a given instrument is constructed from recordings of only one intensity level, it is missing an important dimension of expression. In order to simulate the missing expression, the note can be modified such that if the input control variables specify a higher desired intensity, the volume can be scaled louder and the read-rate for the attack regions can be increased. This simulates the faster attacks that more intense notes usually have.

Each section of the note (attack, decay, sustain, and release) can be given a random component to its length. For example, the attack and decay segments could be given 20% randomness in their length to simulate the irregularities in the attacks of notes as played by human performers. This also helps to prevent the note from sounding exactly the same each time it is sounded.

The progression of time within the note can be manipulated for more novel purposes as well. Time could be linked directly to an input control variable such that a performer could directly control the progression of the note. It could be played forward at any speed and backward as well.

It is important to reiterate that modifying the read-rate changes not only the progression of the power envelope, but also the progression of all other relevant envelopes.

Expression

Since the PRISM technology determines a model for each recorded note, interesting performance characteristics can be recreated. The subtle nuances from real players on real instruments can be captured and optionally manipulated by the synthesis. Manipulation of the features of a given note model are accomplished easily by manipulating the parameterized data. Characteristics can be heightened or eliminated altogether.

Since a note has been broken down into pitch, power, and timbre, each property of the synthesized notes can be controlled individually. If the analyzed note had no vibrato, vibrato can be added in any form. If the tremolo of a modeled flute is too strong, it can be toned down. If the timbre of the modeled note is too jittery, running the filter coefficients through a low pass filter during the analysis stage can smooth it out. The sustain statistics, as well, can be augmented to the sound designer's tastes.

Deliberately unusual results can be achieved. A sound designer could take the power envelope from a plucked guitar note and use it with the timbre of a clarinet to create a "plucked clarinet." A note could be played backwards, where the pitch, power, and timbre progresses from the release of the note to the attack. This would sound different than playing the waveform backwards. Notes can be played slower or faster without any change in pitch in order to produce believable variations in attack rate based on the velocity of a note.

Morphing between unrelated sounds can occur based on any MIDI input. It can even happen automatically over time. A bank of timbres could be stored. As a note is held, the timbre could morph between random selections within the bank of timbres.

MIDI information like velocity, aftertouch, controllers and others can be mapped to control or modulate almost any of the parameters of the notes in real-time. This allows for extremely expressive control of the synthesis.

Realism

Since the PRISM technology has a consistent model structure for each analyzed note, intermediate points in the MIDI space will have intermediate characteristics derived from the multidimensional interpolation of neighboring note models. Furthermore, the characteristics of each note played will vary smoothly as the MIDI note (or velocity, or controller value) is changed and as it evolves in time. Spectral characteristics (or formants) shift appropriately in between modeled notes.

By capturing multiple dimensions of an instrument's behavior, this technology allows a fuller, more realistic reproduction. For example, wavetable will often implement a release for a note with an amplitude envelope applied to the wavetable loop, the PRISM technology and reproduces the timbre information in addition to the power and pitch envelopes during the release. The PRISM technology automatically models performance characteristics whereas many synthesis techniques will emulate expression and realism with tricks.

Realism is further increased by the use of statistical models. These models allow a note to sustain indefinitely, have the same character as the original note without relying on predicable looping.

V. CONCLUSION

In this paper, the core technology of PRISM has been presented. The full parameterization of sound makes PRISM both powerful and flexible. Currently, the PRISM technology works best with sounds composed of harmonic content and noise content. Though inharmonic content is not directly modeled, the PRISM technology would model some characteristics of inharmonic content, such as the beating in the amplitude of low piano notes with stretched high harmonics. Modeling inharmonic content is currently being investigated further.

Implementation of the PRISM synthesis technique in C++ has shown the computational expense to be greater than a wavetable voice, but on par with physical modeling synthesis.

Several examples of useful data manipulation have been presented and many more exist. All of these features combine to form a synthesis engine capable of flexibility, expression, realism, and creative potential that exceeds what is offered by wavetable or physical modeling.

REFERENCES

1. Lindemann et al., “Parametric Signal Modeling Musical Synthesizer,” U.S. Patent #5,744,742, issued Apr. 28, 1998.
2. Massie, D.C., “Wavetable Sampling Synthesis”, in Kahrs, M., and Brandenburg, K., Applications of Digital Signal Processing to Audio and Acoustics, Kluwer Academic Publishers, 1998, pp. 311-342.
3. Smith, J.O., “Principles of Digital Waveguide Models of Musical Instruments,” in Kahrs, M., and Brandenburg, K., Applications of Digital Signal Processing to Audio and Acoustics, Kluwer Academic Publishers, 1998, pp. 417-466.
4. Hayes, Monson H., 1996. Statistical Digital Signal Processing and Modeling, John Wiley & Sons, Inc., New York, New York, pp. 129-195.
5. Patterson, J. H., and Green, D. M. 1970. “Discrimination of Transient Signals Having Identical Energy Spectra.” *Journal of the Acoustical Society of America* 48: 121-131.